KDD Cup 2022

Spatial Dynamic Wind Power Forecasting Challenge

LAKHANI Sunil Harsh
20910249

hslakhani@connect.ust.hk

The Hong Kong University of
Science and Technology

ABSTRACT

In this report, I present a solution for the KDD Cup 2022, focusing on the Spatial Dynamic Wind Power Forecasting (SDWPF) dataset. With the growing importance of clean and renewable energy, wind turbines have become a significant contributor to this energy sector. This challenge necessitates the development of a model capable of accurately forecasting the active power generated by 134 distinct turbines within a single wind farm, providing valuable insights and identifying limitations of the current Supervisory Control and Data Acquisition (SCADA) system. The models' performance is evaluated using the RMSE (root mean squared error) and MAE (mean absolute error) scores. I explore various preprocessing methods, detailed in subsequent sections, and experiment with different model architectures and training settings to achieve the most feasible result.

1 Introduction

The global emphasis on clean and renewable energy has elevated wind energy, generated through wind farms, to unprecedented significance. To enhance operational efficiency, and to enable more informed construction and analysis of wind farms, it is crucial to estimate wind power generation accurately. Such accurate estimations also serve as reliable projections for businesses, thereby attracting investments. However, the task of Wind Power Forecasting presents its own set of complexities. The power generated by wind turbines is notably variable, making the forecasting process challenging.

Another crucial aspect to be considered is the total energy output of a wind farm. Turbines within a wind farm are typically arranged in a grid-like structure. The spatial distribution of these turbines may have varying degrees of influence on the cumulative energy produced by the wind farm and this must be taken into consideration when estimating the energy produced.

For this challenge, we are provided with the SDWPF dataset. The dataset is comprehensive, containing information about the spatial distribution of wind turbines along with a range of other potentially significant factors. Dynamic context features such as temperature and wind speed are included, along with inherent turbine features like internal temperature, nacelle direction, and the pitch angle of each blade. The dataset and its details are discussed further in the later sections of this report, as well as in the original paper [1].

The objective of our challenge is to forecast the active power of an entire wind farm for the forthcoming two days. The specific methods of evaluation are provided and will be discussed further in the subsequent sections of this report.

2 Literature Review

The practice of power forecasting has a rich history of research and applications across multiple domains, particularly within the realm of renewable energy. This literature review will focus on the origin of the dense model architecture used in the solution and its applications in related fields.

The dense model, also known as the multi-layer perceptron (MLP), originates from the field of artificial neural networks. It is arguably one of the most basic and widely-used types of neural networks today. The MLP is characterized by fully connected layers wherein each node in a layer connects to every node in the following layer [2].

Dense layers have found widespread application in the forecasting domain, particularly in weather and energy forecasting tasks. In one such study, MLPs were used to forecast solar and wind energy, which demonstrated the capacity of these models to handle time-series data and make accurate predictions [3]. Another research leveraged MLPs for electricity price forecasting, indicating the adaptability of this architecture to a variety of energy-related tasks [4].

However, the success of MLPs in such tasks does not discount the challenges faced. Overfitting, a common issue in machine learning, is often encountered when using dense layers due to the large number of parameters [5]. Various techniques like dropout, regularization, and early stopping have been proposed to alleviate this problem.

It's worth noting that while our final model architecture is a simpler MLP, we did experiment with more complex architectures, such as Long Short-Term Memory (LSTM) with bi-directional layers and Gated Recurrent Units (GRUs) with convolutional layers. These models are grounded in deep learning

research and have shown promising results in other forecasting tasks [6], [7]. However, due to the risk of overfitting and computational resource constraints, they were not utilized in the final model.

3 Data

3.1 Dataset

As previously indicated, the SDWPF dataset, offers a blend of spatial, internal, and dynamic features and therefore forms the backbone of the analysis. The dataset collates data from the SCADA system and provides a comprehensive profile of 134 wind turbines situated within a single wind farm, forming a grid-like structure. Each turbine contributes data at ten-minute intervals, offering a snapshot of its operations and surrounding conditions over 245 days. This leads to a dataset of approximately 4.7 million rows of data.

The dataset is characterized by 13 distinct features, 10 of which constitute a time series amalgamating internal status and external features influencing each wind turbine's performance. They provide vital information such as wind speed, wind direction, temperature, and nacelle direction. These data points give a detailed and closer look into the operational status and environmental factors each turbine encounters over time.

Along with these time series variables, the dataset provides three auxiliary features that add further depth to the dataset. These features include the day and time, both important elements in understanding the temporal patterns of power generation, and the wind turbine ID, which serves as a unique identifier for each turbine.

A unique aspect of the SDWPF dataset is its spatial component. It includes information on the relative location of each wind turbine within the wind farm. This spatial data allows us to explore the impact of turbine placement and proximity on power generation, offering the potential to extract insights that transcend individual turbine performance.

Detailed information about each feature can be found in Table 1.

Table 1: Feature names and their specifications

Column	Feature Name	Specification
1	TurbID	Wind turbine ID
2	Day	Day of the record
3	Tmstamp	Created time of the record
4	Wspd (m/s)	The wind speed recorded by the anemometer
5	Wdir (°)	The angle between the wind direction and the position of turbine nacelle
6	Etmp (°C)	Temperature of the surrounding environment
7	Itmp (°C)	Temperature inside the turbine nacelle
8	Ndir (°)	Nacelle direction, i.e., the yaw angle of the nacelle
9	Pab1 (°)	Pitch angle of blade 1
10	Pab2 (°)	Pitch angle of blade 2
11	Pab3 (°)	Pitch angle of blade 3
12	Prtv (kW)	Reactive power
13	Patv (kW)	Active power (target variable)

3.2 Exploratory Data Analysis

Along with the dataset, we are provided with some caveats of the data as mentioned down below. This section is the same as in the official dataset introduction report [1].

Zero values. There are some active power and reactive power which are smaller than zeros. We simply treat all the values which are smaller than 0 as 0, i.e. if $Patv < 0$, then $Patv = 0$.

Missing values. Note that due to some reasons, some values at some time are not collected from the SCADA system. These missing values will not be used for evaluating the model. In other word, if p_{t_0+j} is a missing value, we set $|Patv_{t_0+j} - \overline{Patv}_{t_0+j}| = 0$ regardless of the actual predicted value of $\overline{Patv}_{t_0+j}$.

Unknown values. In some time, the wind turbines are stopped to generate power by external reasons such as wind turbine renovation and/or actively scheduling the powering to avoid overloading the grid. In these cases, the actual generated power of the wind turbine is unknown. These unknown values will also not be used for evaluating the model. Similarly with the missing values, if $Patv_{t_0+j}$ is a unknown value, we always set $|Patv_{t_0+j} - \overline{Patv}_{t_0+j}| = 0$. Here we introduce two conditions to determine whether the target variable is unknown:

• If at time t, $Patv \leq 0$ and $Wspd > 2.5$, then the actual active power $Patv$ of this wind turbine at time t is unknown;

• If at time t, $Pab1 > 89°$ or $Pab2 > 89°$ or $Pab3 > 89°$, then the actual active power $Patv$ of this wind turbine at time t is unknown.

Abnormal values. There are some abnormal values from the SCADA system. If a data record has any abnormal value of any column, these values also will not be used for evaluating the model. If $\overline{Patv}_{t_0+j}$ is an abnormal value, we always set $|Patv_{t_0+j} - \overline{Patv}_{t_0+j}| = 0$. Here we define two rules to identify the abnormal values:

• The reasonable range for Ndir is [-720°, 720°], as the turbine system allows the nacelle to turn at most two rounds in one direction and would force the nacelle to return to the original position otherwise. Therefore, records beyond the range can be seen as outliers caused by the recording system. Thus, if at time t

there are Nidir > 720° or Nidir < -720°, then the actual active power $Patv$ of this wind turbine at time t is abnormal.

• The reasonable range for Wdir is [-180°, 180°]. Records beyond this range can be seen as outliers caused by the recording system. If at time t there are Widr > 180° or Widr < -180°, then the actual active power $Patv$ of this wind turbine at time t is abnormal.

Drawing from these guidelines, it is clear that the model performance will not be evaluated on missing, abnormal, or unknown values. Thus, it is not logical to train the model to recognize patterns within these values. However, dismissing these values entirely is also not a viable solution. I discuss how to manage these values in the Data Preprocessing section of this report.

3.3 Data Preprocessing

This section outlines the major steps taken in preprocessing the data, adhering to the official paper's guidelines and logical methodologies for handling the dataset's different types of values.

Zero Values. Values less than zero are treated as zero, that is, if $Patv < 0$, then $Patv = 0$.

Missing Values. There are 2 logical methods proposed to deal with Missing Values:

1. Leverage the spatial data provided in the dataset to substitute missing values with those from the neighboring turbine for a given day and time.
2. Use linear interpolation of the same turbine's values for that day to fill missing values.

The rationale for using the nearest turbine's values is that, assuming the nearest neighbor is oriented in the same direction as the original turbine, the recorded values would likely be similar. At the same time, the linear interpolation method is also a viable option and a good estimator of the missing values. Additionally, linear interpolation for a given day preserves the natural pattern of that day, making it a superior estimator compared to traditional methods.

To handle missing values, I make use of the linear interpolation method. The reason why the nearest neighbour method is not used is due to its computational expense and time consumption when run on the entire dataset. However, the code is included in the supplementary notebook for future reference and experimentation. Moreover, when tested on a smaller sample dataset, the linear interpolation method, although logically the second-best estimator, yielded better prediction performance than the nearest turbine method. This, however, may not be the case when applied to the entire dataset.

Abnormal/Unknown values. I categorize and treat abnormal and unknown values the same. The average of the immediate previous and next row's data is used to replace the categorized abnormal values. More specifically, for a given cell of data, considering it is categorized abnormal, it is replaced with the average of the immediate normal cell before and after it within the same column. This method maintains the natural pattern of the data and reduces overall abnormalities in the dataset, thereby training the model to make more accurate predictions. Moreover, since the abnormal values will not be used to evaluate the model, it is unnecessary to train our model to predict the target variable when considering these values.

3.4 Feature Engineering

The process of feature engineering played a crucial role in the analysis. Utilizing a heatmap shown in Figure 1 for visual representation, I was able to discern correlations between different features and make informed decisions about the approach taken.

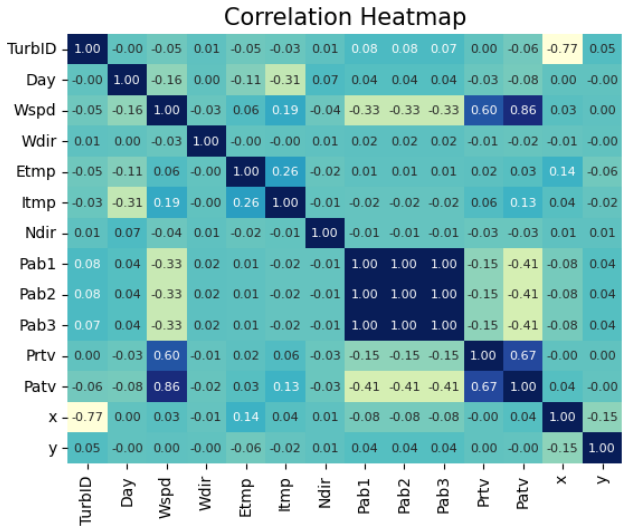

Figure 1: Heatmap

One significant observation was the perfect correlation among Pab1, Pab2, and Pab3. These features represent the pitch angles of a turbine's blades at a given time. Given their perfect correlation, it was decided to amalgamate them into a single feature, Pab_max, which essentially stores the maximum value of the three pitch angles.

Further analysis revealed minimal correlation between the target variable (Patv) and several other features: 'Wdir', 'Etmp', 'Itmp', and 'Ndir'. Given their lack of substantial contribution towards predicting the target variable, it was deemed efficient to remove these features from the dataset.

In addition to correlation analysis, I explored potential trends within the data. For some turbines on certain days, a unique pattern was observed wherein the wind speed (Wspd) increased during the first half of the day, then decreased, and subsequently increased again towards the end of the day, the code and figure supporting this can be viewed in the supplementary notebook. This trend, coupled with Wspd's strong correlation with Patv, led me to focus on Wspd as a potentially important feature.

One of the more intuitive steps in the feature engineering process involved the 'Tmstamp' feature. Given its potential to uncover hidden patterns, I decided to transform 'Tmstamp' into sine and cosine values. This approach, commonly adopted in time series forecasting tasks, allowed me to capture the cyclical nature of

time. As a result, I added the following columns: 'hour_sin', 'hour_cos', 'minute_sin', 'minute_cos'.

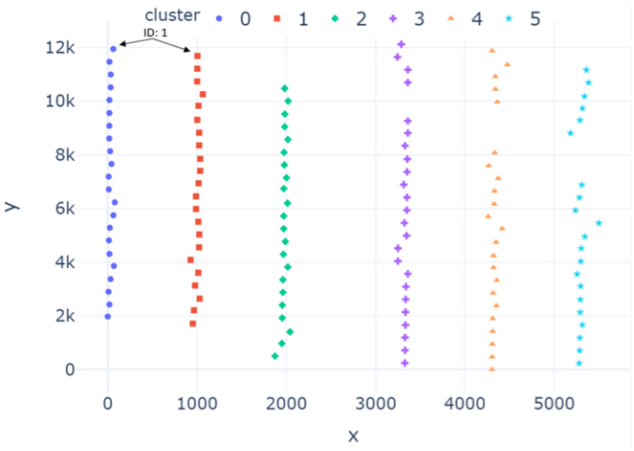

Figure 2: Cluster and ID assignments

'TurbID', a categorical variable with 134 unique values, presented a unique challenge. While one-hot-encoding could remove any ranking bias, it would exponentially increase the dimensions, making this a unfeasible option. Therefore, I opted to split 'TurbID' into two features, 'Cluster' and 'ID'. 'Cluster' represents the turbine's location with respect to its X-coordinate, and 'ID' represents its location with respect to its Y-coordinate. This can be seen in Figure 2. This approach significantly reduced the ranking bias, making it a practical solution to this categorical feature.

In summary, the data preprocessing and feature engineering process was instrumental in refining the dataset, providing valuable insights and a more manageable, efficient set of features for the model.

4 Methodology

This section elucidates the experiments conducted to train the model, providing a comprehensive analysis of the resulting outcomes.

4.1 Experiments

These are the 2 experiments conducted:

1. Train a model for all turbines (entire wind farm) together.
2. Train a model for each turbine separately.

To ensure fairness, these experiments were conducted using the same dense model architecture.

Rationale for Two Experiments. The choice of experimenting with these two settings (training a model for each turbine separately and training a model for the entire wind farm) stems from the possibility that the dataset may contain information about correlations between the turbines. Behaviors of one turbine may influence another, establishing a pattern that the model may recognize. This could be a double-edged sword since these influences could be random and thus confuse the model, negatively affecting its performance. However, these correlations and patterns could also be meaningful, providing vital information that improves the model's performance. Therefore, the only way to truly find out is by experimenting with these two methods.

4.2 Evaluation Metrics

According to the official dataset technical paper, the evaluation method is as follows: The challenge requires to address the Spatial Dynamic Wind Power Forecasting ahead of 48 hours. For example, given at 6:00 A.M. today, it is required to effectively forecast the wind power generation beginning from 6:00 A.M. on this day to 5:50 AM on the day after tomorrow, given a series of historical records of the wind farm and the related wind turbines. It is required to output the predicted values every 10 minutes. To be specific, at one time point, it is required to predict a future length-288 wind power supply time-series. The average of RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) is used as the main evaluation score.

More specifically, at a time step t_0, it is required to predict a time series of wind power of the wind farm $P = \{p_{t_0+1}, p_{t_0+2}, \cdots, p_{t_0+288}\}$.

However, due to the missing and unknown values for each wind turbine, in this challenge, we evaluate the prediction results for each wind turbine, and then sum the prediction scores as the final score of the model. The evaluation score si for wind turbine i at the time step t_0 is defined as:

$$s_{t_0}^i = \frac{1}{2}\left(\sqrt{\frac{\sum_{j=1}^{288}\left(Patv_{t_0+j}^i - \overline{Patv}_{t_0+j}^i\right)}{288}} + \frac{\sum_{j=1}^{288}\left|Patv_{t_0+j}^i - \overline{Patv}_{t_0+j}^i\right|}{288}\right)$$

where $Patv_{t_0+j}^i$ is the actual power of wind turbine i and $\overline{Patv}_{t_0+j}^i$ is the predicted power of the wind turbine i at time step $t_0 + j$. Note that each time step of j is 10 minutes. The overall score of the prediction model S_{t_0} at time t_0 is the sum of the prediction score on all wind turbine, i.e.:

$$S_{t_0} = \sum_{i=1}^{134} s_{t_0}^i$$

Following the evaluation guidelines, I calculate and report the RMSE and MAE scores between the actual and predicted value for each turbine separately. After this, I proceed to add up the RMSE and MAE score of each turbine to get the overall RMSE and MAE score for the entire wind farm. The scores can be seen in the results section. The overall score for the entire wind farm is calculated as the average of the two scores.

4.3 Data Generation

The final feature set comprises of:

{'hour_sin', 'hour_cos', 'minute_sin', 'minute_cos', 'Day', 'Cluster', 'ID', 'Wspd', 'Pab_max', 'Prtv', and 'Patv'}.

A custom function was employed to generate the data required for training the model. Specifically, the function is a sliding window that selects one weeks' worth of data (1008 rows) from the feature set (including the target variable 'Patv') and trains the model to predict the next two days' worth of data (288 rows). More

information on the custom data generation function can be found in the supplementary code.

4.4 Model Architecture and Training

To ensure fair comparison, the same model architecture was used across the experiments. This section discusses the model architecture used in detail.

Rationale for Model selection. After numerous experiments, he final model architecture selected for this task is a deep feed-forward neural network, also known as a multi-layer perceptron (MLP), with dropout and L2 regularization. This decision was influenced by several considerations, particularly the nature of the problem and the goal of the task.

First, it's worth noting that the architecture incorporates several layers of dense neurons, which allows the model to learn more complex patterns in the input data. The number of neurons in each layer (128, 64, 32, and finally 256) forms a funnel-like structure that helps condense the high-dimensional input data into a lower-dimensional output while retaining the most salient information. The activation function for these layers, 'relu', or Rectified Linear Unit, helps the model learn non-linear relationships in the data, crucial for tasks such as power forecasting which can often involve complex, non-linear dynamics.

The inclusion of dropout layers is a strategic move to prevent overfitting, a common issue when training deep neural networks. By randomly setting a proportion (70% in this case) of input units to 0 at each update during training time, dropout helps to create a more robust model by ensuring that it doesn't rely too heavily on any single feature or combination of features. This encourages the network to learn a more generalizable representation of the data, thereby improving its performance on unseen data.

The architecture also employs L2 regularization, another measure aimed at preventing overfitting. By adding a penalty equivalent to the square of the magnitude of the weights to the loss function (scaled by a factor of reg_lambda), L2 regularization discourages the model from assigning too much importance to any single feature. This again helps the model generalize better to unseen data.

The final layer of the model is a dense layer with 288 neurons, corresponding to the number of predictions the model needs to make (i.e., power output for the next 288 time steps). The model does not employ an activation function in this layer as the task is a regression problem, and I want the outputs to be unbounded.

The learning rate (lr) of 0.0005 was chosen based on empirical tuning, and it determines the step size during gradient descent. A smaller learning rate can help the model converge to a global minimum by taking smaller, more precise steps.

It's important to note that while other more complex architectures such as Long Short-Term Memory (LSTM) with bi-directional layers and Gated Recurrent Units (GRUs) with convolutional layers were experimented with, these models were found to be overly complex for the task at hand. Despite their theoretical capabilities, these models resulted in overfitting and did not yield better results. This underscores the principle that a more complex model is not always better, and it's essential to choose a model architecture that's appropriate for the complexity of the data and the task.

4.5 Results

This section presents a comprehensive analysis of the results obtained from the experiments conducted.

Table 2 displays the scores for two experiments conducted.

Table 2: Train and Validation Scores

Training Method	Type of Score	Scores
Each turbine separately	Train	RMSE: 512.70 MAE: 362.92
	Validation	RMSE: 456.89 MAE: 272.51
All turbines together	Train	RMSE: 351.76 MAE: 255.95
	Validation	RMSE: 390.84 MAE: 281.30

It is important to note here that the scores calculated are in Kilo Watts as opposed to Mega Watts. However, the scores calculated on the test set is in Mega Watts to draw a comparison against the baseline model. After obtaining the results for both of the experiments conducted, it is observed that training a model for all turbines performs considerably better than the other, therefore, I select that as the best model and evaluate it on the test sets provided. The training and validation loss curves for this model can be seen in Figure 3.

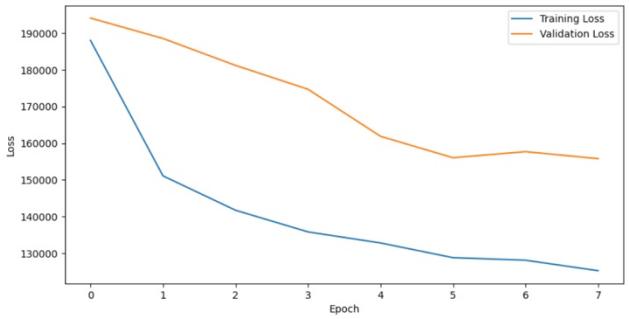

Figure 3: Training and Validation Loss Curves

As seen in Table 3 that displays the scores for test set, the final model was able to attain an MAE score of 42.11 and a RMSE score of 63.93. It is important to note here that these scores are calculated in Mega Watts. The overall score, calculated as the average of the two scores, equates to 53.03. We observe that this score is not as low as the baseline model score which is 42.32. This is attributed to several reasons including the preprocessing methods as well as the model architecture and settings, more notably, there was a heavy influence of the limited computational resources available on the decisions made when training the model. These scores do give us an estimate of how the model performs, but we cannot directly compare it to the performance of the baseline since the training and validation settings differ.

Table 3: Test Scores

Training Method	Type of Score	Scores
Each turbine separately	Test	RMSE: 63.93 MAE: 42.11

5 Conclusion

This report presented an approach to tackle the complex problem of Spatial Dynamic Wind Power Forecasting, a crucial task in the renewable energy sector. Wind energy, due to its inherent variability, poses significant challenges in terms of accurately predicting power output. However, precise forecasting models are integral for improving operational efficiency, enabling informed decision-making, and attracting potential investments in the wind energy sector.

The solution discussed in this report was applied to the SDWPF dataset provided for the KDD Cup 2022. This dataset included a wide range of features, from spatial and temporal context variables to specific turbine characteristics. Throughout this study, various preprocessing methods were employed to appropriately prepare the data for subsequent modeling.

Experimentation with different model architectures and training settings was a significant part of this project. Although complex models like LSTM with bi-directional layers and GRUs with convolutional layers were considered, the final model was a more straightforward multi-layer perceptron with dropout and L2 regularization. This decision was primarily guided by the need to avoid overfitting and the constraints imposed by the available computational resources.

Despite these constraints, the final model achieved a promising overall score of 53.03 on the test set. This outcome underscores the utility of the chosen model architecture, as well as the effectiveness of the preprocessing methods employed. However, this result also highlights the room for improvement in future work. Further refinements in the model architecture, more advanced preprocessing techniques, or even the incorporation of additional relevant features could potentially enhance the model's forecasting performance.

In conclusion, this study demonstrates that while the task of wind power forecasting is complex, it is tractable with the appropriate preprocessing techniques, careful model selection, and thoughtful training strategies. Despite the challenges faced, the accomplishments in this study contribute to the broader effort of making renewable energy sources, like wind energy, more predictable and thus more reliable.

6 Future Work

Potential directions for future work include:

1. Experiment with using the nearest turbine’s values to replace missing values for entire dataset.
2. Experiment with different methods of dealing with abnormal, unknown and outlier values.
3. Experiment with more complex models such as XTGN, MDLinear, AGCRN and MTGNN, where some of these are also capable of taking into consideration spatial data that may help to improve the model’s performance.
4. Collect data for the entire year as opposed to only 245 days to capture possible trends and patterns within the whole year.

REFERENCES

[1] Zhou, J., Lu, X., Xiao, Y., Su, J., Lyu, J., Ma, Y., & Dou, D. (2022). SDWPF: A Dataset for Spatial Dynamic Wind Power Forecasting Challenge at KDD Cup 2022.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[3] Mellit, A., & Kalogirou, S. A. (2008). Artificial intelligence techniques for photovoltaic applications: A review. Progress in Energy and Combustion Science, 34(5), 574-632.

[4] Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting, 30(4), 1030-1081.

[5] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

[6] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[7] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.