IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 1, Ser. 3 (Jan. – Feb. 2024), PP 12-27
www.iosrjournals.org

Application Of Darts Model For Comparing Machine Learning Algorithms To Predict Wind Speed And Direction Using Solar Data

Tosin Harold Akingbemisilu[1]
[1](Department Of Architecture, Computing And Engineering, University Of East London, London)

Abstract:
Wind energy prediction plays a crucial role in renewable energy planning and policymaking. This study leveraged the Darts time-series framework to compare 10 different machine learning algorithms to investigate the use of solar radiation data for accurate wind energy prediction and compare with previous study results. Evaluation was conducted using two performance metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to inform on the prediction's accuracy.The best-performing model for wind speed prediction was the CatBoost model with a 12-hour lag, achieving an RMSE score of 0.877, while for wind direction prediction, CatBoost also achieved the best performance with an RMSE value of 93.8, but with a 2-hour lag, both with the inclusion of covariates. In comparing the performance of the top model trained using Darts in this study with previous research conducted without Darts but utilizing similar models available in Darts, the use of Darts yielded comparable or even superior results with better computational efficiency. Classical models, particularly CatBoost, outperform neural network models in terms of accuracy and computational time, providing valuable insights for investment decision-making. Also, the inclusion of covariates significantly enhances the performance of wind energy prediction models. Covariates such as gust, global horizontal irradiance (GHI), and relative humidity demonstrate a strong influence on wind speed and wind direction predictions.With efficient and accurate prediction of wind speed and wind direction using the Darts Model framework, optimal utilization of wind resources can be achieved, ensuring a sustainable and reliable energy supply.
Keywords*:Darts; Random Forest; Neural Network; Covariates; Wind Energy; Forecasting; Nigeria*

Date of Submission: 25-01-2024 Date of Acceptance: 05-02-2024

I. Introduction

Wind energy is a rapidly growing source of renewable energy, and it has the potential to contribute significantly to any country's energy mix [1]. Wind energy prediction plays a crucial role in various applications within the renewable energy sector, including renewable energy planning, grid integration, and energy storage which is vital for the sustainability of the sector [2]. Renewable energy planning greatly benefits from wind energy prediction by enabling optimal site selection for wind energy projects [3]. By analyzing wind energy data from multiple locations, planners can identify areas with high wind energy potential, resulting in more efficient and cost-effective project implementation [4].

Machine learning (ML), a subset of artificial intelligence (AI), utilizes algorithms and statistical models to enable computers to learn from data, recognize patterns, and make predictions or decisions autonomously [5]. In the case of wind energy time series forecasting, ML algorithms can analyze historical wind data to identify patterns and forecast future wind speed and direction. These algorithms are trained on extensive datasets of past wind measurements and can generate models for predicting wind power generation over different time spans, e.g., hours or days [6]. Another significant advantage of ML in wind energy time series forecasting lies in its capacity to handle complex nonlinear relationships among variables like wind speed, direction, and power output, even when conventional statistical methods may falter [7].

Accurate and reliable prediction of wind speed and direction is essential for the effective management and planning of wind energy projects. Some researchers have highlighted the importance of accurate forecasts of wind direction and speed in various sectors, including aviation, energy production, and public safety [8]. Accurate forecasts of these variables can help optimize decision-making and risk management in these sectors, which could directly translate to a higher utilization of wind energy, aiding the country's shift to a low-carbon economy [9].

In recent studies, the application of various models for predicting wind speed and direction, as well as incorporating solar radiation data, has been explored, although without the use of Darts model. Notably,

[10]leveraged advanced deep learning techniques for medium-term forecasting of solar irradiation and wind speed, finding deep learning models to outperform traditional methods, with the encoder-decoder LSTM model showing superior accuracy. [11] focused on short-term wind speed prediction using hybrid techniques, identifying the Least Squares Support Vector Machine (LSSVM) as the most accurate and computationally efficient among evaluated non-parametric techniques. [12]introduced a feedforward backpropagation neural network for predicting wind speed and direction, demonstrating its superior accuracy over other models. Meanwhile, [13] compared various artificial learning-based algorithms for short-time wind speed forecasting, highlighting the ConvLSTM model's efficiency and accuracy. These studies contribute valuable insights into the predictive capabilities of different models, underscoring the importance of incorporating various data inputs and evaluating against a range of computational and accuracy metrics, yet also point out a gap in the literature regarding long-term forecasting and the consideration of additional influential factors such as temperature and humidity. With the efficiency of Darts model in easy comparison of different models and the use of covariates for time series analysis, introduction of lags and more, most of the limitations found in previous studies is aimed to be addressed in this study.

The Darts model is a new versatile Python library designed for time series forecasting and anomaly detection. It encompasses a wide range of models, from classic statistical models like ARIMA to advanced deep learning models. One of its key features is its ability to allow models to be used in a consistent manner with simple 'fit and predict' functions. Darts supports multivariate time series and can handle multiple series training with machine learning-based models. It also offers probabilistic forecasting capabilities, allowing for confidence interval estimation, and includes tools for anomaly detection, data processing, and performance metrics evaluation. The library additionally provides a range of evaluation and comparison tools, including performance metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). It supports both univariate and multivariate time series and models, and some of the models offer probabilistic forecasting and the ability to explain models using Shap values This makes it a comprehensive tool for time series analysis [14].

This study is driven by the need to improve the accuracy and efficiency of wind direction and speed forecasts and to see how Darts model framework can enhance this. It is expected that the comparison of various algorithms for predicting wind direction and speed using solar radiation measurement data will deliver valuable insights for improving wind direction and speed forecasts, optimizing decision-making, and risk management, as well as contributing to scientific knowledge on machine learning applications in meteorology and inform future studies. This study is also expected to fill the gap in knowledge, based on lack of studies on the use of Darts framework and few studies that have been conducted on the use of machine learning algorithms to predict wind direction and speed feasibly and accurately using solar measurement data. The study hypotheses include:

- Machine learning algorithms can accurately predict wind direction and speed using solar radiation measurement data and at least one machine learning algorithm will perform significantly better than the others.

Scope and limitations of the study

- Data availability: In this study, the availability of the solar radiation measurement data used is limited to a year period of per second data collected in Bauchi by World Bank in a solar development project meant to collect two years of ground measurement data for the planned utility scale photovoltaic (PV) power plant at the described location. The project commenced in 2021 and so far, the data repository only contains data from September 2021 – September 2022 [15].
- Sample size: The sample size of the data used in this study may be a limitation, since the data is collected over a short period of time and from only one State. A larger sample size would likely result in more accurate and reliable predictions but may also require more resources and time to collect and analyse. Using this data as a baseline, however, can still give a clue on the potential of collecting more data to improve the quality of the Darts models in future work.
- Generalizability: Other locations or time periods may not be directly generalizable based on the results of this study. The accuracy of machine learning algorithms in predicting wind direction and speed in other states or countries with different climatic conditions may vary from their performance in the selected Nigerian state. However, one can apply the process to create a new model for any state in the presence of data.
- External factors: Machine learning algorithm predictions could potentially be impacted by external factors mentioned in a study conducted. Wind direction and speed may be altered by both topographic features like mountains and human activities like building or changing land usage, which affects prediction accuracy, and are not accounted for in the solar radiation measurement data.

The remaining part of this study is structured into three main sections. Section 2 explores the materials and methodologies employed in this research. Section 3 presents the results and discussions of findings from the data and the models. Finally, Section 4 encapsulates the conclusions and recommendations from this study.

II. Material And Methods

The study adopts a comparative analysis approach to evaluate and compare the various machine learning algorithms from the Darts framework. The statistical analyses in this study utilized the Pandas library for computations and analysis. Descriptive statistics, such as mean, standard deviation, minimum, maximum, and percentiles, were calculated to understand data distribution and variability. Data visualization was performed using matplotlib and seaborn libraries. A pair-plot was generated to visualize pairwise relationships between features, revealing patterns and correlations. A correlation matrix was also used to assess the strength of correlations between columns, including target variables and other features. Heatmaps were created to examine patterns of predictive labels over time, including variables such as hour of the day, day of the month, week of the month, and month of the year. These heatmaps revealed time-related trends and patterns, aiding in the understanding of influential factors and potential feature engineering for future predictions.

Data Collection and Preparation

In this study, the data was obtained from solar and meteorological ground measurements repository, collected from a network of weather stations located in Bauchi State, Nigeria, as part of the West African Power Pool project [15]. The data contains measurements of different parameters taken per second, with a minimum period of one year from September 2021 to September 2022. Bauchi State was chosen for this study due to its availability of data. Bauchi is a northern state in Nigeria with large land area and diverse topography, including the northern end of the Jos Plateau, the Yankari Game Reserve, and the Gongola River.

The World Bank and West African Power Pool (WAPP) implemented a solar resource measurement campaign project in Bauchi, Nigeria to measure solar irradiance and other relevant parameters for solar energy power plant projects. An automatic weather station (AWS) was installed at the TCN substation in Bauchi with the objective of collecting two years of ground measurement data for a planned utility-scale photovoltaic (PV) power plant. The measurement data is regularly transmitted to Concentrating Solar Power Services (CSPS) for data quality monitoring and control, and it is also accessible on a protected web server for real-time data monitoring and download [15]. Table 1 below shows the measurement parameter descriptions in the data which was equally used in this study.

Table 1: Measurement Parameter Descriptions of the Solar Measurement Data [15]

Measurement	Description	Unit
Timestamp	Date and time according to ISO8601	yyyy-mm-dd hh:mm
GHI	Global horizontal irradiance	W/m²
DNI	Direct normal irradiance	W/m²
DHI	Diffuse horizontal irradiance	W/m²
Mod_A	PV soiling measurement system (reference module)	W/m²
Mod_B	PV soiling measurement system (measurement module)	W/m²
T_{amb}	Ambient temperature	°C
RH	Relative humidity	%
WS	Wind speed	m/s
WS_{gust}	Maximal wind speed (3 second average)	m/s
WS_{stdev}	Standard deviation of wind speed	m/s
WD	Wind direction	°N (to east)
WD_{stdev}	Standard deviation of wind direction	°
BP	Barometric air pressure	hPa
Cleaning	Cleaning of sensors: A cleaning event is marked with a "1"	1 or 0
Precipitation	Precipitation (rain)	mm/min
T_{ModA}	Backside module temperature (Mod_A)	°C
T_{ModB}	Backside module temperature (Mod_B)	°C

Data Pre-processing Techniques, Feature Extraction and Selection

Various pre-processing techniques were applied to prepare the dataset for model training using Darts. Initially, the relevant columns from the Bauchi data were selected, including Timestamp, GHI, DNI, DHI, ModA, ModB, Tamb, RH, WSgust, BP, Precipitation, TModA, TModB, WS, and WD. Less relevant columns such as standard deviation of wind speed and wind direction, as well as sensor cleaning indication, were dropped to prevent data leakage since these variables were derived from the labels to be predicted.

The data was loaded and cleaned using the Pandas library which is the most common and powerful library for data manipulation in python. Since the first row contained the variable name, and the second row contain the unit of measurement, we dropped the second row that contains the unit of measurement when loading, as we do not need it by using the skiprows parameter set to "1" to skip the second row that contains the unit of measurement. Also, while loading, the data had an encoding, so we used a 'unicode_escape()' encoding method and got the data information using the 'info()' function in python, giving us 525,600 records with 14 columns. We parsed the timestamp column as dates and used it as our index column. Other data types were checked and fixed accordingly to ensure each numeric values are float or integer type. We also checked for missing values using the 'isnull()' method and found no missing values, therefore there was no need to address missing values through interpolation with nearest numbers, since we are dealing with time series.

Since we have our data in per second, which was too granular, we resampled our data using the 'resample()' method, which is a powerful resampling tool in pandas. For all the variables, including wind speed we aggregated the mean for each hour. However, for wind direction which is a circular data (360^o), we used the circular mean method or the mean resultant vector. To achieve the resampling method, we used the Numpy library to efficiently define a function that first converts the input wind direction data x from degrees to radians using 'np.radians(x)'. It then takes the mean of the cosine and sine of the input data using 'np.mean(np.cos(x))' and 'np.mean(np.sin(x))', respectively. The mean wind direction is then calculated using the arctan2() function, which returns the angle between the positive x-axis and a point given in polar coordinates (i.e., the sine and cosine of the angle).

Finally, we converted the result back to degrees using 'np.degrees()' and applied modulo 360 to ensure that the output falls within the range of 0 to 360 degrees. This approach is known as the circular mean or directional statistics, and it is a well-established technique for calculating the mean direction of circular data such as wind direction. While we took the mean of other variables, precipitation variable was summed, because this shows the amount of rainfall measured over a certain period, usually over an hour. Therefore, by summing precipitation over the hour, we get the total amount of rainfall over that period, which is a useful measure of weather conditions and averaging precipitation would not give us an accurate representation of the amount of rainfall during that hour.

To accommodate the training and testing of models with and without covariates, the wind speed and wind direction covariates were converted into Darts time series objects, and like the target variables, these covariates were transformed and split into training and validation sets. The transformation process involved the use of a scaler to normalize the data and a missing values filler to handle any missing values. Covariates in this context refer to additional variables in the data that are used to inform the model training process for both wind speed and direction. By including covariates, the models can consider other relevant factors that may influence the prediction accuracy, thereby improving the overall performance.

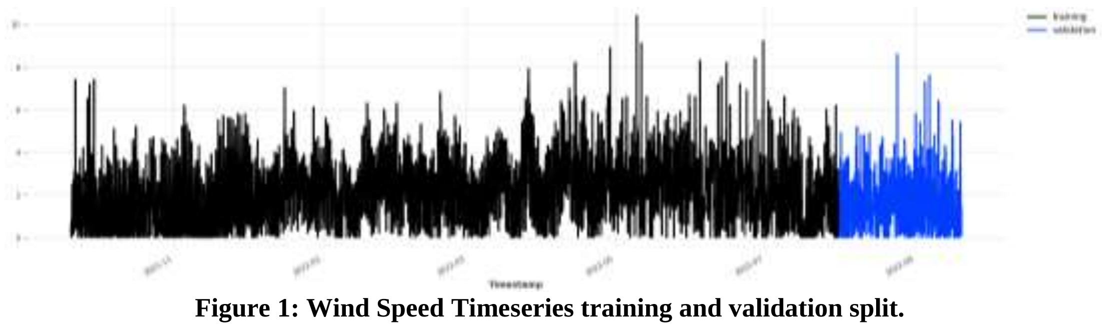

Figure 1: Wind Speed Timeseries training and validation split.

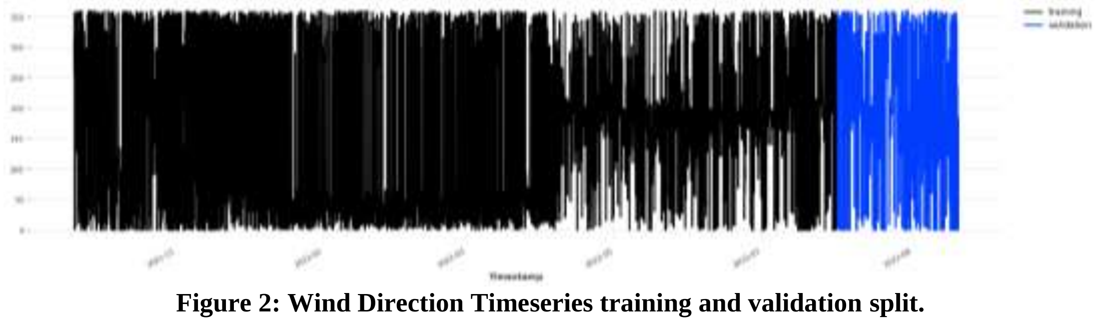

Figure 2: Wind Direction Timeseries training and validation split.

Darts Machine Learning Algorithms Selection and Configuration

10 machine learning algorithms were selected to provide an overview of how different types of algorithms performed in effectively predicting wind speed and direction across Darts. To help understand how each model would perform on univariate and multivariate data, considerations were made regarding the use of covariates and using target variables only for prediction. Models representing linear, decision trees, boosting algorithms, and neural networks were implemented to span our comparison across different model algorithm types. The algorithms used in the project are discussed below and link to details of the trained models are available in Appendix 3.

Linear Regression Model

The Linear Regression model in Darts is a univariate model that uses a linear equation to predict the target variable based on its past values. It learns these coefficients through ordinary least squares (OLS) regression, minimizing the sum of squared errors between predicted and actual values. To avoid overfitting, the model supports L1 and L2 regularization, which add penalty terms to the OLS objective function. Additionally, the model can incorporate lagged values of external time series (covariates) to enhance its predictive performance [16, 14]. In the case of our project, we used both the simple regression using only date and target variable; and multiple linear regression method which has the addition of all the covariates. The algorithm for a simple linear regression explains the linear relationship between the dependent (output) variable y and the independent (predictor) variable X [16].

$$\mathbf{Y_i = \beta_0 + \beta_1 X_i}$$

where Y_i = Dependent variable, β_0 = constant/Intercept, β_1 = Slope/Intercept, and X_i = Independent variable

On the other hand, the multiple linear regression uses the technique to understand the relationship between a single dependent variable and multiple independent variables which has a formula like simple linear regression, except that instead of having one beta variable, we now have for all the variables used as shown below:

$$\mathbf{Y = B_0 + B_1X_1 + B_2X_2 + \ldots + B_nX_n + \varepsilon}$$

Linear-regression models have been chosen in this study since it helps to understand the linearity, strength of predictors using a simple procedure and it has been a proven algorithm for prediction.

Random Forest Algorithm

Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is implemented in the RandomForest class in Darts, which allows the user to specify hyperparameters such as the number of trees to include in the model, the maximum depth of each tree, and the number of features to consider when splitting nodes. Additionally, RandomForest in Darts supports parallelization to speed up training and can be used for both univariate and multivariate time series forecasting tasks [17, 14].

Random Forest has been chosen in this study to represent decision tree algorithm, and due to its high accuracy, robustness, feature importance, adaptability, and scalability. It also reduces overfitting by averaging multiple decision trees and is less sensitive to noise and outliers available in the data [17].

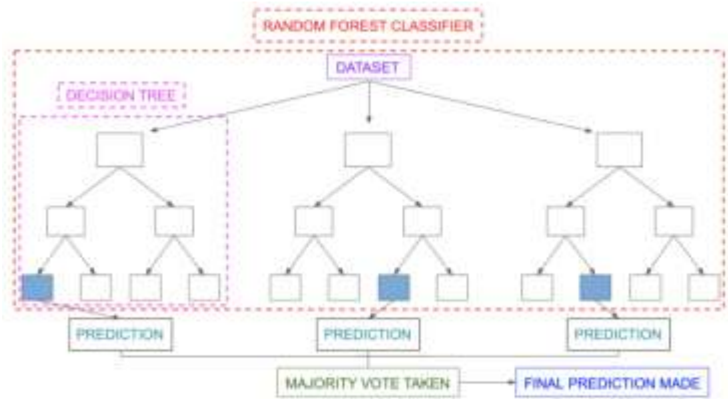

Figure 3: Diagrammatic representation of Random Forest Model Implementation [18]

Gradient Boosting Algorithms (CatBoost, LightGBM and XGBoost)

Gradient boosting on decision trees is a machine learning technique used by CatBoost and it was chosen as it is known for its ability to handle categorical characteristics and reduce the need for data pre-processing, making it a high-performance tool. Darts offers CatBoost as one of its machine learning algorithms for time series data prediction. A series of decision trees are generated, with each tree created to correct the flaws of the previous tree in the chain. To handle categorical variables with numerous levels, CatBoost employs techniques such as ordered boosting and gradient-based One-Group Selection [14].

Light Gradient Boosting Machine is another gradient boosting system that employs tree-based learning techniques. It is intended to be efficient and capable of handling huge datasets, which is known to be generally faster and more memory efficient. Darts' LightGBM model is a time series forecasting application of the LightGBM algorithm. It uses previous values of the target variable and historical covariates to anticipate future values of the target variable. The LightGBM model in Darts also lets you to tune hyperparameters like learning rate, number of trees, and tree depth to improve the model's performance [14].

XGBoost short for extreme gradient boosting, is another gradient boosting machine learning method used in Darts that creates a sequence of weak models that improve on each other to generate a final prediction. Although based on decision trees, it advances other methods such as random forest and gradient boost, with its ability to work well with large, complicated datasets by using various optimization methods. It uses gradient descent to optimize the objective function and can handle missing values, numeric and categorical data. It includes a regularization term to minimize overfitting and has been shown to be useful in a variety of machine learning applications, including univariate and multivariate time series forecasting [14].

Block Recurrent Neural Network (BRNN)

In Darts, the BRNN model is a customized recurrent neural network (RNN) that employs long short-term memory networks (LSTM) cells and a novel input layer architecture. Its input layer is made up of numerous blocks of LSTM cells, allowing it to capture both short-term and long-term relationships in the data. To boost efficiency, the outputs of these blocks are concatenated and transferred to the next layer, and the model also permits skip connections between LSTM blocks, allowing previous outputs to be used as inputs while having hidden states. The architecture of a traditional RNN is shown below:

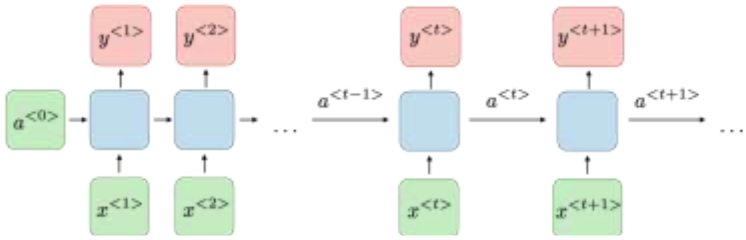

Figure 4: Diagrammatic Representation of BRNN Architecture [19].

For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:
$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$ and $y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$
where W_{ax}, W_{aa}, W_{ya}, b_a, b_y, are coefficients that are share temporally and g_1, g_2 are activation functions.

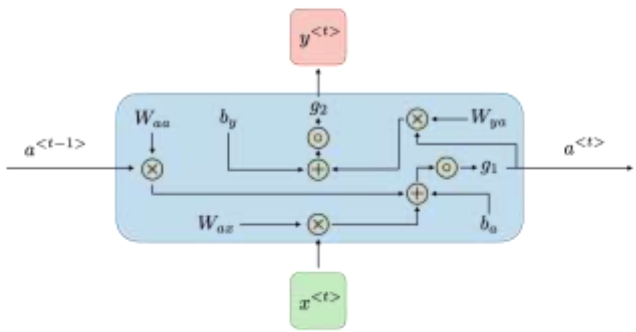

Figure 5: Diagrammatic Representation of a specific Block in BRNN[19]

Overall, the Darts BRNN model is well-suited for time series forecasting applications, taking advantage of the characteristics of LSTM cells, and supporting variable-length input sequences.

Neural Basis Expansion Analysis Time Series Forecasting (N-BEATS)

Darts' N-BEATS model is a deep learning system for time series forecasting. It is a deep neural architecture based on backward and forward residual links and a very deep stack of fully connected layers with several required properties, being explainable, relevant without modification to a wide array of target domains, and fast to train. The diagram below shows the architectural representation of N-BEATS.

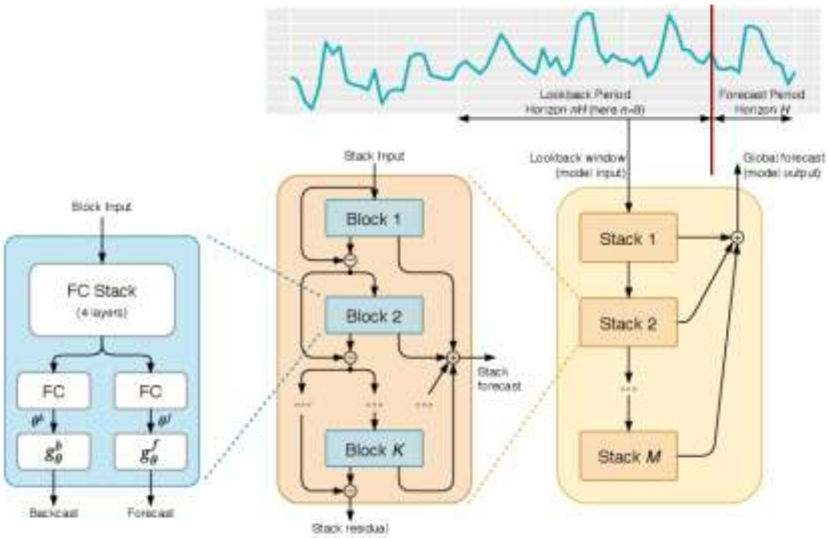

Figure 6: Diagrammatic Representation of N-BEATS Architecture [20]

N-BEATS generates a collection of fundamental functions that are coupled linearly to provide predictions for future time steps using a completely parallelized and interpretable architecture consisting of fully linked layers with gating mechanisms. It may be trained on several time series and manage a variety of temporal granularities. It also allows for the use of exogenous input information to improve predicting accuracy. Overall, Darts' N-BEATS model combines the benefits of deep learning with interpretability and flexibility, making it a useful tool for time series forecasting problems [14, 20].

Temporal Fusion Transformer

The Darts Temporal Fusion Transformer (TFT) model is a deep learning technique for time series forecasting. It employs the Transformer architecture and includes attention techniques as well as autoregressive modelling. It is made up of encoder and decoder blocks, with numerous layers of attention mechanisms and feed-forward neural networks in each. The encoder analyses past data and develops a latent representation of the input sequence, while the decoder makes predictions based on the encoder's output and future covariate inputs. A gating mechanism is also used by the TFT model to dynamically modify its weights based on the importance of each input characteristic. The TFT model is a useful tool for time series forecasting applications because to its ability to handle complicated data structures and deliver reliable forecasts[14].

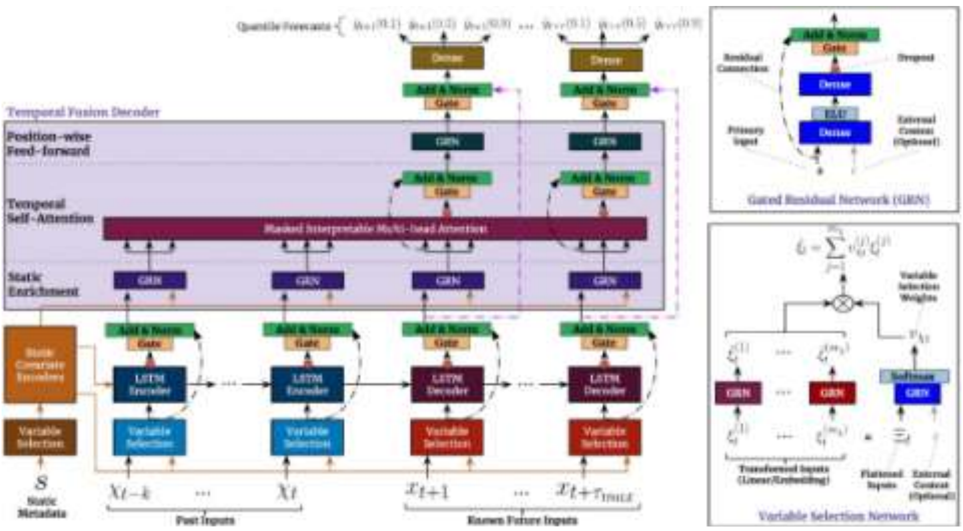

Figure 7: Diagrammatic Representation of TFT Architecture [21]

Temporal Convolutional Network (TCN)

The TCN model in Darts is a deep learning model that extracts features from time series data using 1D dilated convolutional layers. It overcomes the constraints of classic recurrent neural network models and is ideal for univariate and multivariate time series forecasting. It is made up of many blocks that have dilated causal convolutional layers, residual connections, and normalization. To capture information at different sizes, the dilation rate grows exponentially, and its outputs are fed into a fully linked layer for final predictions. Regularization and scaling can be accomplished using dropout regularization and activation functions. Darts' TCN model is a strong time series forecasting method, particularly for long-term forecasts[14].

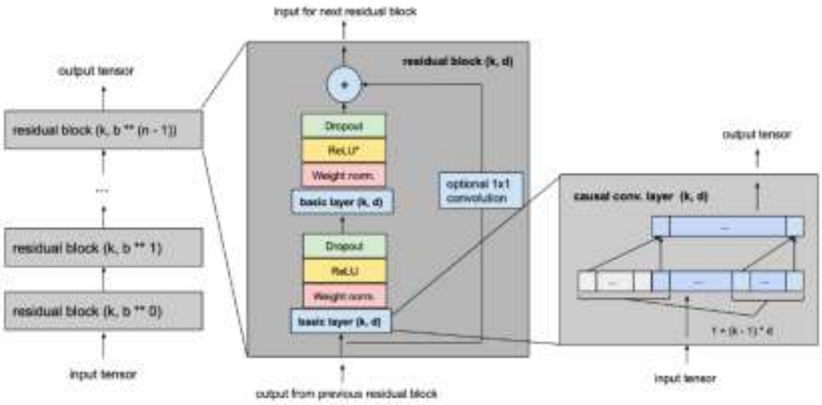

Figure 8: Darts TCN Model Framework [14].

Transformer Model

Darts' Transformer model is a deep learning model for time series forecasting. To avoid information leakage, it employs self-attention mechanisms, a position-wise feedforward network with residual connections, a causal mask, and seasonal embeddings to capture periodic patterns. Backpropagation over time may be used to train it, and gradient descent can be used to optimize it. Across several datasets, the Transformer model outperformed standard forecasting algorithms [22, 14].

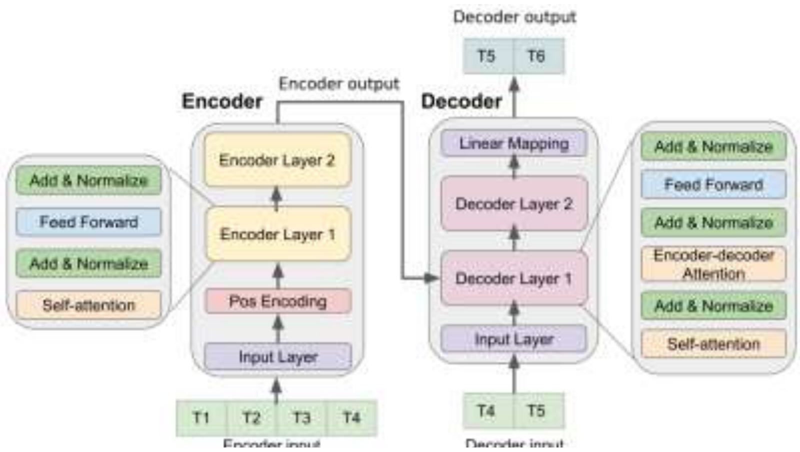

Figure 9: Architecture of transformer-based forecasting model [22]

Model Training and Evaluation for Performance Comparison

To ensure objectivity, models were trained using default parameters and without hyperparameter tuning. Their performance was evaluated on a separate validation set as mentioned earlier, to ensure predictions are done on data the model has never seen before; using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Results, including metrics, were presented in a table and graph for fair comparison.

MAE measures the average absolute difference between predicted and actual values. This helps us understand the magnitude of difference between the prediction of an observation and the true value of that observation which is also referred to as L1 loss function. It is an easy-to-understand quantifiable measurement of errors in regression problems and it is widely used due to its resiliency to outliers or extreme values, coming down to the degree at which we want to be able to penalize large errors. Below shows mean absolute error equation:

$$\mathbf{MAE = (1/n)\ \Sigma(i=1\ to\ n)\ |y\_i - \hat{y}\_i|}$$

n = the number of observations; Σ = summation symbol (which means "add them all up"); y_i is the true value; ŷ_i is the predicted value.

RMSE on the other hand, computes the square root of the mean of the squared differences. It is more sensitive to outliers compared to RMSE which helps to penalize large errors more than MAE since the errors are initially squared. It is interpreted as the average weighted performance of the model, where a larger weight is added to outlier predictions.[52] Root means square error can be expressed as:

$$\mathbf{RMSE} = \sqrt{[\Sigma (Pi - Oi)^2 / n]}$$

Pi is the predicted value for the ith observation in the dataset; Oi is the observed value for the ith observation in the dataset; n is the sample size.

Although both metrics are similar, but MAE returns values that are more interpretable as it is simply the average of absolute errors. In the project, we want to be able to see the differences in the metrics, while trying to penalize large errors, and at the same time, treating all errors equally to return a more interpretable value, hence the choice of using both RMSE and MAE. Generally, both metrics can go from 0 to infinity, with lower values indicating better performance, while values close to zero indicates high accuracy and precision in predictions.

Furthermore, the impact of time lags on model performance was explored by varying lag values for target variables and past covariates in all models except neural networks. Lag helps identify patterns in data, therefore, lag values of 1, 2, 6, 12, 24, and 48 hours were compared. Two evaluation approaches were used: one with target variables only and the other incorporating covariates. This aimed to determine optimal lag values and assess covariate impact on predictions.

Computational efficiency was also evaluated by measuring prediction time for each model on the validation data. This analysis, using the Python time library, identified differences in speed and computational requirements, crucial for real-world applications. This is important to help understand the model that would be more efficient by saving time, cost for computing resources.

Lastly, to enable us to explain what happened in our models from input to output, model explain-ability was investigated using the Shap Explainer in Darts. The SHAP method determined each feature's contribution to predictions, considering past lags of targets and covariates. SHAP values created a summary plot, ranking features by importance, aiding interpretation of the model's behavior.

III. Results And Discussion

This section aims to provide a comprehensive analysis of the study's outcomes and contribute to the understanding of the research problem. The results of the experiments conducted is presented, including model performance, explain-ability, and their significance. The link to all code repositories is available in Appendix 3.

Descriptive Statistics of the Dataset

As shown in figure 10 below, the analysis conducted on wind speed and direction revealed that their distributions were right skewed for both the original and resampled data. Resampling resulted in higher frequency counts at the minimum and maximum wind direction values, but the statistical properties of wind speed remained stable. However, the maximum wind speed value decreased from 24.1 to 10.4 after resampling, and most samples fell within the range of 0-5. Wind direction exhibited a slight increase in mean and median values with an evenly distributed sample.

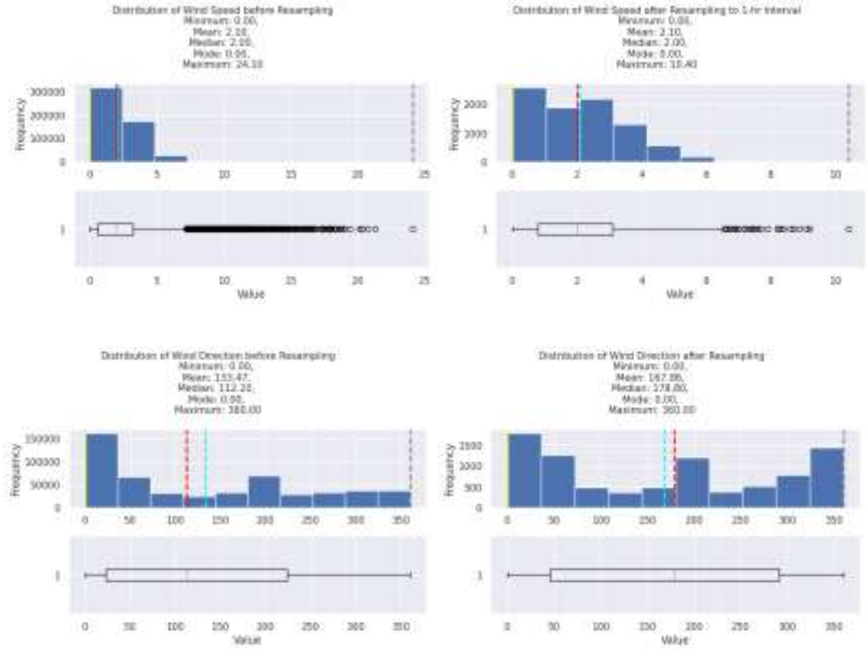

Figure 10: Distribution of Wind Speed and Direction

The analysis also examined the influence of time-related factors, such as the hour of the day and the month of the year, on wind speed and wind direction. The findings revealed significant patterns in both variables. Figure 11 and 12 clearly illustrates that certain periods of the day and specific months of the year have a substantial impact on wind speed and wind direction.

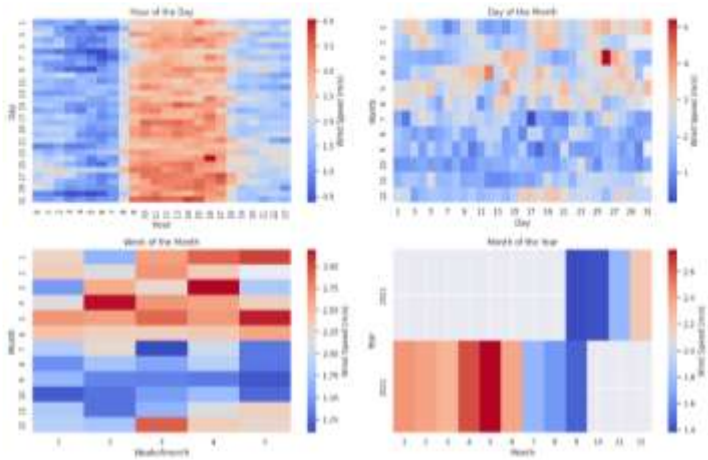

Figure 11: Wind Speed (m/s) Patterns Across Time

Specifically, the analysis demonstrated that wind speed tends to increase predominantly between the 9th and 17th hour of the day. Additionally, the increase in wind speed is more pronounced during the months of December to June, while it remains comparatively lower during the rest of the year. These observations strongly influenced the selection of time features extracted from the data's timestamp. Overall, the analysis highlights the importance of considering time-related factors when studying wind speed and wind direction, with the hour of the day and the month of the year emerging as significant variables affecting these meteorological phenomena.

Like wind speed, wind direction displayed distinct patterns in relation to these variables, as depicted in figure 12. In the early hours of the day (the first 7 hours), the data consistently indicated that the wind direction predominantly ranged from 200° to 360°N (eastward). For the remaining hours of the day, the wind direction tended to be less than 200°N (eastward). This pattern emphasizes a consistent shift in wind direction as the day progresses.

Additionally, the analysis revealed a discernible pattern in wind direction concerning the months of the year. The data indicated a higher degree of wind direction within the 4th to 9th months, while the degree of wind direction was comparatively lower in the other months of the year. This finding highlights a seasonal influence on wind direction, suggesting that specific months exhibit a more consistent trend in wind direction than others. These findings underscore the significance of both the hour of the day and the month of the year in understanding wind direction variations. By considering these temporal factors, a more comprehensive understanding of wind patterns and their relationship to meteorological phenomena can be obtained.

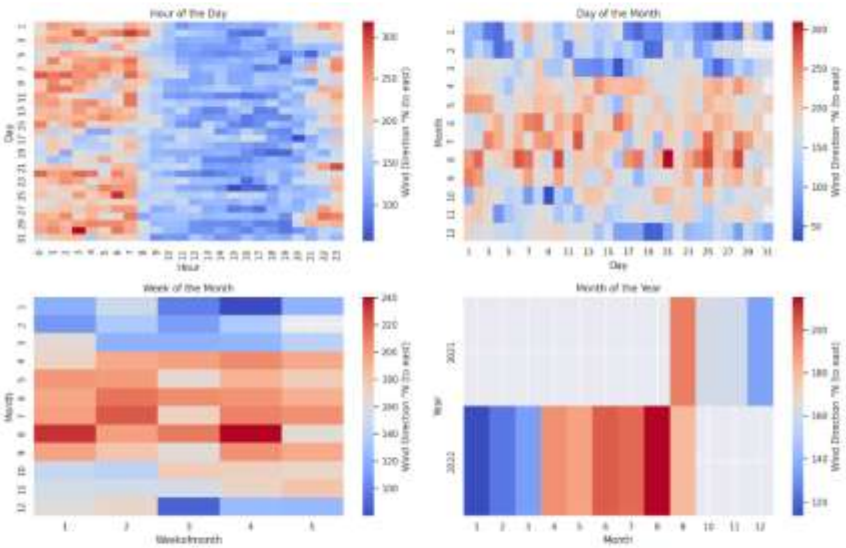

Figure 12: Wind Direction ^{0}N (to east) Patterns Across Time

A correlation matrix was constructed to investigate the relationships between various variables. As displayed in figure 13, the matrix revealed interesting findings regarding the associations between wind speed, wind direction, and other covariates. Firstly, wind speed exhibited a slight positive correlation with most of the other variables, except for relative humidity and wind direction, which demonstrated a negative correlation. This implies that as wind speed increases, the other variables tend to exhibit a positive trend, except for relative humidity and wind direction, which show a negative relationship with wind speed.

Conversely, wind direction displayed a negative correlation with most of the covariates, except for relative humidity. This suggests that as wind direction changes, the other variables tend to exhibit a negative trend, except for relative humidity, which shows a positive relationship with wind direction. Moreover, the correlation analysis revealed that barometric pressure and precipitation exhibited the lowest or no significant correlation with both wind speed and wind direction. This implies that these variables have minimal or negligible influence on the variations observed in wind speed and wind direction.

Overall, the correlation matrix analysis provided insights into the relationships between wind speed, wind direction, and the other examined variables. It indicates that certain covariates have either positive or negative associations with wind speed and wind direction, highlighting the complex interplay between these meteorological factors.

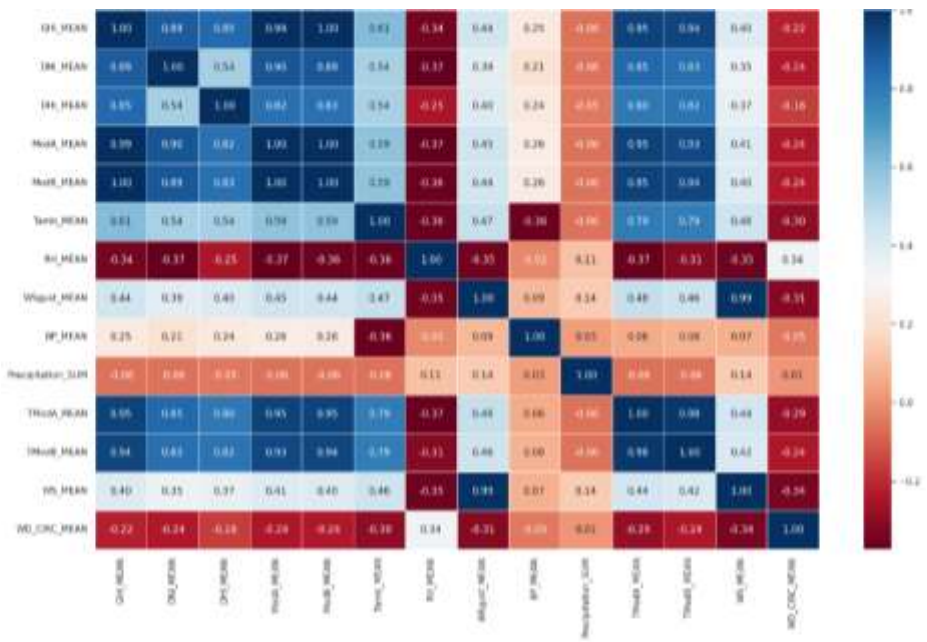

Figure 13: Correlation Matrix to view relationship between features in Bauchi State Solar Measurement Data

Performance Comparison of the Machine Learning Algorithms

The performance comparison of machine learning algorithms involved training and evaluating ten different models on a previously unseen validation set. To investigate the impact of covariates and lag parameters on model performance, the models were trained in two ways: univariate training using only wind speed and direction as target variables, and incorporating covariates with different lag parameters (1, 2, 6, 12, 24, and 48-hour lags). This resulted in a total of 140 model iterations from the ten algorithms. The neural network models, which lacked lag attributes, were trained without incorporating lags.

From the numerous trained models, the top 25 models, determined based on their RMSE (Root Mean Square Error) values, were selected for further discussion in this results section. The comprehensive table, available in appendix 1 and 2, provides details on all the models trained, including their usage of covariates, lag parameters, as well as the corresponding RMSE and MAE (Mean Absolute Error) values.

Wind Speed Performance

The chart below displays the top 25 models, ranked by their RMSE scores, for predicting wind speed based on evaluation on the validation datasets. The best-performing model among them was the CatBoost model with a 12-hour lag, achieving an RMSE score of 0.877. The findings highlight the importance of incorporating covariates into model training to improve performance. CatBoost and LightGBM demonstrated excellent performance compared to other models, while XGBoost did not make it into the top 25. None of the neural network models ranked in the top 10 for wind speed prediction, although TCN and Transformer models showed potential.

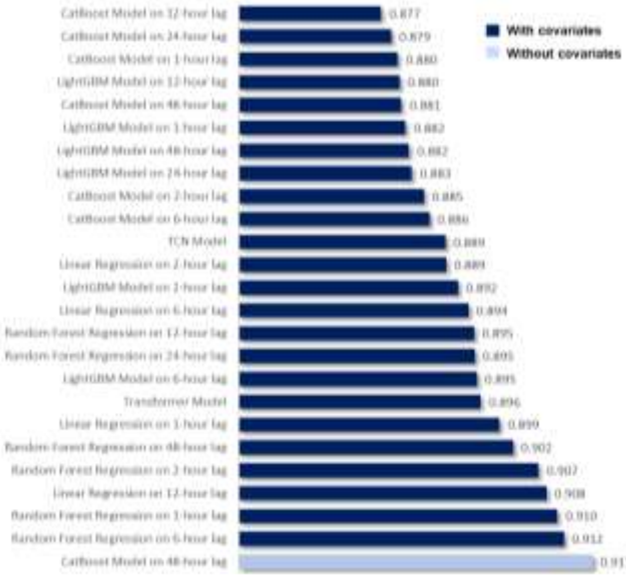

Figure 14: RMSE value of top 25 models for wind speed prediction

The top two models, both trained on CatBoost with 12-hour and 24-hour lags, suggest the presence of a seasonal effect on model performance. The graphical output of the top model displaying actual and predicted values, is presented below.

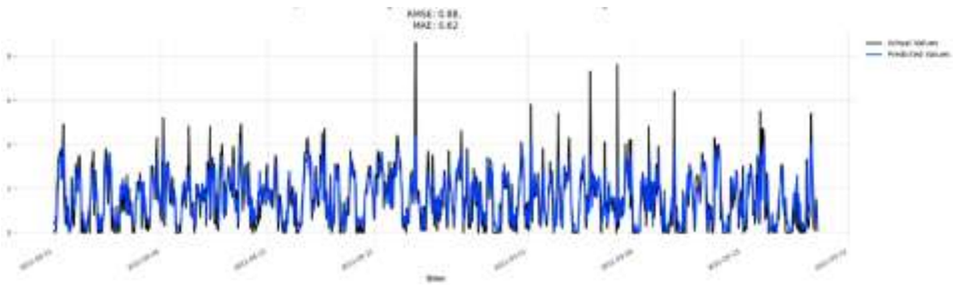

Figure 15: Wind Speed Prediction Using CatBoost Model with Covariates on 12-hour Lag

In addition to evaluating model performance, the computational efficiency of the models was examined by calculating how long each fitted model took to predict the validation data. The graph below depicts the elapsed time for the top ten wind speed forecast models. The LightGBM model with covariates on a 24-hour lag had the lowest prediction time of the top ten models, requiring only 14 seconds. Even our top-performing model, with an elapsed duration of 18 seconds, maintained a reasonable performance. It's worth noting that the Catboost model, which was trained on a 48-hour lag, had the highest forecast time of the top ten models, clocking in at 25 seconds. The chart below depicts the predicted elapsed time in visual form.

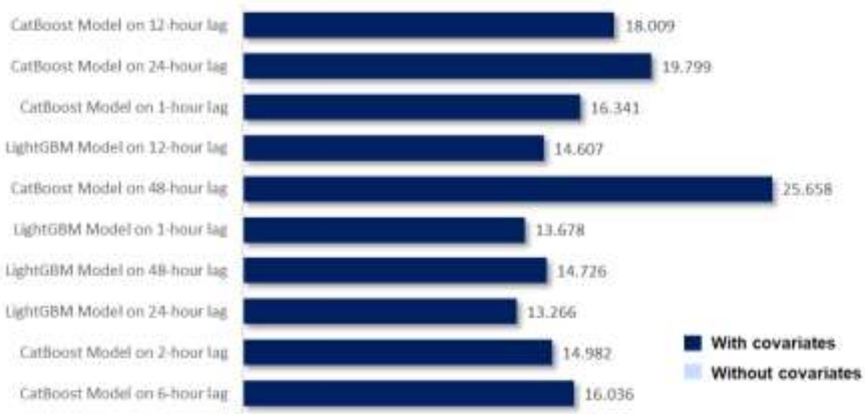

Figure 16: Prediction Elapsed Time in seconds of the top 10 models for wind speed prediction

Wind Direction Performance

The chart below presents the top 25 models, along with their characteristics, for the prediction of wind speed based on the evaluation of RMSE scores on the validation datasets. Similarly, for wind direction prediction, CatBoost achieved the best performance with an RMSE value of 93.8, but with a 2-hour lag. Considering the wide range of wind direction values (0 – 360), one might expect a higher level of performance.

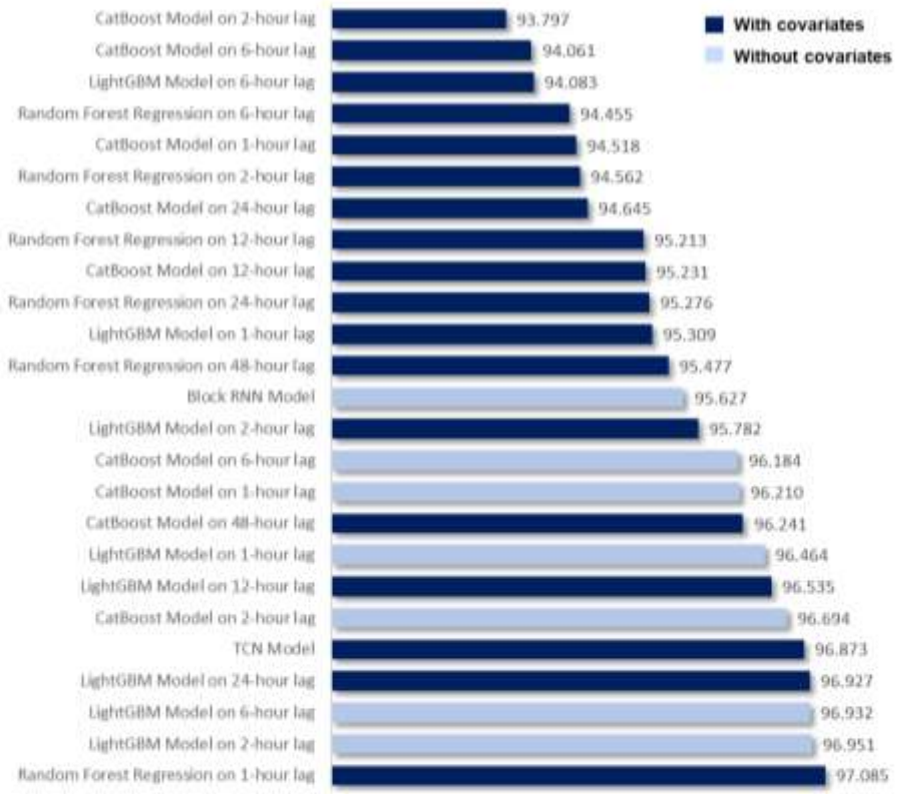

Figure 17: RMSE value of top 25 models for wind direction prediction

However, it is important to note that, in general, only models incorporating covariates made it into the top 10. This suggests that the inclusion of covariates in the model training had a significant impact on performance improvement. The graphical output of the top model, CatBoost with covariates on a 2-hour lag, displaying actual and predicted values, is presented in the chart below.

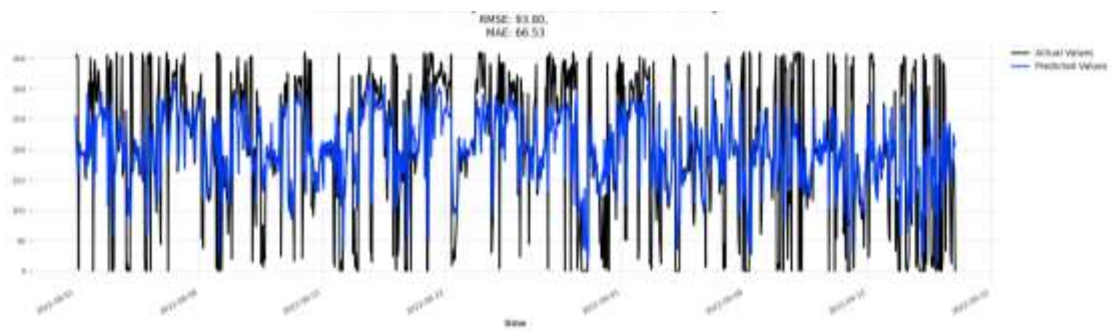

Figure 18: Wind direction prediction using CatBoost Model with Covariates on 2-hour lag.

In addition to evaluating model performance, the computational efficiency of the models in predicting wind direction was also assessed. Like the prediction of wind speed, the LightGBM model with covariates on a 6-hour lag achieved the shortest prediction time, taking approximately 13 seconds. However, our top-performing model still demonstrated a favorable performance, with an elapsed time of around 16 seconds. The chart below provides a visual representation of the prediction elapsed time in seconds for the top 10 models in wind direction prediction.

Figure 19: Prediction elapsed time in seconds of the top 10 models for wind direction

Overall, the performance of the Darts time-series model framework in predicting both wind speed and wind direction shows promising prospects, considering that no hyperparameters were tuned and the models were trained on one year of data. The lowest RMSE achieved was 0.877 for the prediction of wind speed using CatBoost on a 12-hour lag, and 93.8 for wind direction. These results suggest that there is a likelihood of further improvement by tuning the hyperparameters and incorporating more data in future work for both variables. With proper hyperparameter optimization and a larger dataset, the predictive accuracy of the models could potentially be enhanced, leading to even more accurate predictions for both wind speed and wind direction.

Comparison of Performance against Previous Studies

In comparing the performance of the top model trained using Darts in this study with previous research conducted without Darts but utilizing similar models available in Darts, notable findings were observed. The use of Darts yielded comparable or even superior results, even when simpler models were employed with their default settings.

For example, the study by [13] achieved an RMSE of 0.8718 for 1-hourly data prediction using LSTM, which was on par with the CatBoost model trained in this study, yielding an RMSE of 0.877 without any tuning and with a smaller dataset. Additionally, [12], in their attempt to predict wind speed and wind direction, obtained an RMSE of 1.068 and 44.34, respectively. In contrast, our trained models showcased significantly improved performance for wind speed, with the best performing CatBoost model achieving an RMSE of 0.877 for wind speed, however lower for wind direction which was 93.8 in our case. Nonetheless, these comparisons highlight the efficacy of the Darts framework in producing an equally fair or better-performing models for wind speed and direction prediction without any hyper-parameter tuning, when compared with other studies.

Analysis of the Top Performing Model Features and Importance

An analysis was conducted to assess the overall impact of covariates and lags on the model performance. The findings revealed that the introduction of covariates had a positive effect on the performance of the models. On the other hand, increasing the lag parameter did not have a significant impact on performance.

Specifically, for wind speed prediction, a lag of 12 hours consistently yielded the best performance across all models. For wind direction prediction, a lag of 6 hours demonstrated the best performance. In general, it was observed that most of the neural network models trained in this study performed poorly compared to the classical models. This suggests that for the task of wind speed and wind direction prediction, the classical models outperformed the neural network models in terms of accuracy and predictive power.

These findings provide insights into the influence of covariates and lags on model performance, highlighting the importance of incorporating covariates and selecting appropriate lag parameters for achieving accurate predictions. Furthermore, the comparison between neural network models and classical models underscores the need to carefully consider the choice of model architecture to optimize performance for wind speed and wind direction prediction.

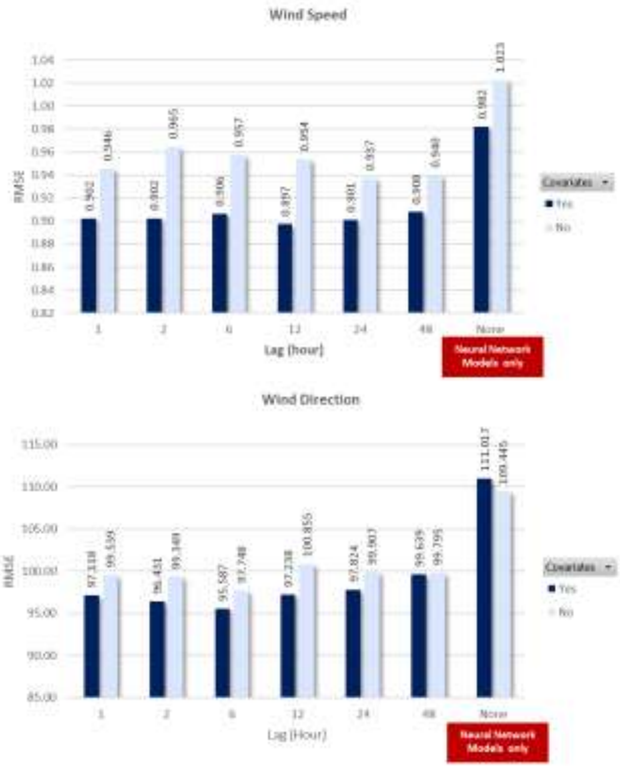

Figure 20: Assessing Average RMSE values by lag(hr) and Covariates for Wind Speed and Direction Prediction

Additionally, to gain insights into the features that significantly influenced the model performance, the Shap Explainer in Darts was employed. This allowed us to understand how the models' made predictions and the importance of each input feature in the prediction process.

The chart below presents the explainability of the top models for both wind speed and wind direction. For wind speed prediction, the top five features contributing to the model's performance were identified as the previous 1-hour lag of wind speed, gust, wind direction, GHI (Global Horizontal Irradiance), and the 2-hour lag of wind speed. These features played a crucial role in accurately predicting wind speed.

Conversely, for wind direction prediction, the model trained on a 2-hour lag of both the target variable and covariates (using the CatBoost model that yielded the best performance) identified the following top five features as most influential: 1-hour lag of wind direction, relative humidity, wind speed, and the 2-hour lag of both DNI (Direct Normal Irradiance) and RH (Relative Humidity). These features significantly contributed to the predictive performance of the wind direction model.

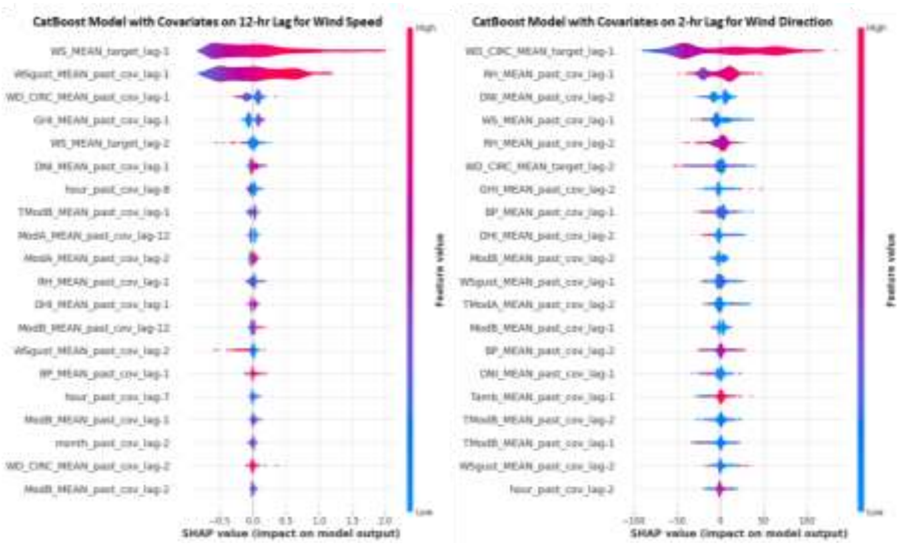

Figure 21: Model Explainability of the top models for wind speed and direction prediction

By analysing the explainability of the models, we gained valuable insights into the key features driving their predictions. This information can be utilized to further optimize and refine the models, ultimately improving their performance in wind speed and wind direction prediction tasks.

IV. Conclusion

In this paper, we highlighted the potential of machine learning algorithms within the Darts framework, particularly when combined with covariates, for accurate wind speed and direction prediction using solar radiation data. The findings provide important considerations for wind energy management and policymaking, particularly in the context of renewable energy planning in Nigeria. The study revealed that the inclusion of covariates significantly improved the performance of prediction models, highlighting the importance of considering additional factors such as gust, GHI, and relative humidity. It also highlighted the superior performance of classical models, particularly CatBoost, compared to neural network models for wind energy prediction and performed in par with models from previous studies even without hyperparameter tuning. This finding provides valuable guidance for investment decision-making, as it suggests that classical models using Darts model should be considered reliable tools for assessing the feasibility and profitability of wind energy projects. The findings of this study also have significant implications for wind energy management and policy, particularly in the areas of energy planning, resource assessment, grid integration, and investment decision making. These implications are not only relevant on a broader scale but also specifically applicable to renewable energy planning in any country. With the implementation of Darts model for wind energy prediction, there is potential to unlock wind energy resources efficiently, achieve greater energy security, reduce greenhouse gas emissions, and contribute to a sustainable and diversified energy mix. Finally, future research can be done by expanding the study to longer data periods, tuning hyperparameters, and considering regional variations, which can further enhance the robustness, reliability, accuracy, and applicability of wind energy prediction models using the Darts framework.

References

[1] Y. N. Chanchangi, F. Adu, A. Ghosh, S. Sundaram and T. K. Mallick, "Nigeria's Energy Review: Focusing on Solar Energy Potential and Penetration," *Environment, Development and Sustainability,* p. 5755–5796, 2022.

[2] M. Santhosh, C. Venkaiah and D. M. Vinod Kumar, "Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review," *Engineering Reports,* vol. 2, no. 6, 2020.

[3] W. O. Idris, M. Z. Ibrahim and A. Albani, "The Status of the Development of Wind Energy in Nigeria," *Energies,* vol. 13, no. 23, p. 6219, 2020.

[4] A. Al-Quraan, B. Al-Mhairat, A. M. A. Malkawi, A. Radaideh and H. M. K. Al-Masri, "Optimal Prediction of Wind Energy Resources Based on WOA—A Case Study in Jordan," *sustainability,* vol. 15, no. 5, p. 3927, 2023.

[5] M. Algren, W. Fisher and A. E. Landis, "Data Science Applied to Sustainability Analysis," in *Chapter 8 - Machine learning in life cycle assessment*, Elsevier, 2021, pp. 167-190.

[6] A.-N. Buturache and S. Stancu, "Wind Energy Prediction Using Machine Learning," *Low Carbon Economy,* vol. 12, no. 1, 2021.

[7] H. Guo, J. Wang, Z. Li and Y. Jin, "A multivariable hybrid prediction system of wind power based on outlier test and innovative multi-objective optimization," *Energy,* vol. 239, p. 122333, 2022.

[8] A. Alkesaiberi, F. Harrou and Y. Sun, "Efficient Wind Power Prediction Using Machine Learning Methods: A Comparative Study," *energies,* vol. 15, no. 7, p. 2327, 2022.

[9] D. Gielen, F. Boshell, D. Saygin, M. D. Bazilian, N. Wagner and R. Gorini, "The Role of Renewable Energy in the Global Energy Transformation," *Energy Strategy Reviews,* vol. 24, pp. 38-50, 2019.

[10] K. Blazakis, Y. Katsigiannis and G. Stavrakakis, "One-Day-Ahead Solar Irradiation and Windspeed Forecasting with Advanced Deep Learning Techniques," *energies,* vol. 15, no. 12, p. 4361, 2022.

[11] L. Lopez, I. Oliveros, L. Torres, L. Ripoll, J. Soto, G. Salazar and S. Cantillo, "Prediction of Wind Speed Using Hybrid Techniques," *energies,* vol. 13, no. 23, p. 6284, 2020.

[12] A. P. Sari, H. Suzuki, T. Kitajima, T. Yasuno and D. A. Prasetya, "Prediction Model of Wind Speed and Direction Using Deep Neural Network," *Journal of Electrical Engineering, Mechatronic and Computer Science,* vol. 3, no. 1, 2020.

[13] M. Ibrahim, A. Alsheikh, Q. Al-Hindawi, S. Al-Dahidi and H. ElMoaqet, "Short-Time Wind Speed Forecast Using Artificial Learning-Based Algorithms," *Computational Intelligence and Neuroscience,* pp. 1-15, 2020.

[14] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik and G. Grosch, "Darts: User-Friendly Modern Machine Learning for Time Series," *Journal of Machine Learning Research,* vol. 23, no. 124, pp. 1-6, 2022.

[15] R. Affolter, "Nigeria - Solar Radiation Measurement Data," 2022. [Online]. Available: https://energydata.info/dataset/nigeria-solar-radiation-measurement-data.

[16] K. Mali, "Everything you Need to Know About Linear Regression," 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/.

[17] A. Sharma, "Random Forest vs Decision Tree | Which is Right for You?," 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[18] E. Y. Boateng, J. Otoo and D. A. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, 2020.
[19] A. Amidi and S. Amidi, "VIP Cheatsheet: Recurrent Neural Network," 2018. [Online]. Available: https://github.com/afshinea/stanford-cs-230-deep-learning/blob/master/en/cheatsheet-recurrent-neural-networks.pdf.
[20] E. Stevenson, V. Rodriguez-Fernandez, E. Minisci and D. Camacho, "A deep learning approach to solar radio flux forecasting," *Acta Astronautica*, vol. 193, pp. 595-606, 2022.
[21] B. Lim, S. Arik, N. Loeff and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748-1764, 2021.
[22] L. K. Albin, "How to make a Transformer for time series forecasting with PyTorch," 2022. [Online]. Available: https://towardsdatascience.com/how-to-make-a-pytorch-transformer-for-time-series-forecasting-69e073d4061e.

Appendix

Appendix 1.Algorithm Performance for Wind Speed Prediction

ML Technique	ML Algorithm	Lag (hr)	Covariates	RMSE	MAE	Elapsed Time(s)
With Covariates for Wind Speed Prediction						
Ensemble Technique	Random Forest	1	Yes	0.910	0.641	26.679
		2	Yes	0.907	0.638	26.773
		6	Yes	0.912	0.640	26.343
		12	Yes	0.895	0.637	28.159
		24	Yes	0.895	0.634	30.138
		48	Yes	0.902	0.641	28.971
Gradient Boosting	CatBoost	1	Yes	0.880	0.619	16.341
		2	Yes	0.885	0.620	14.982
		6	Yes	0.886	0.624	16.036
		12	Yes	0.877	0.619	18.009
		24	Yes	0.879	0.623	19.799
		48	Yes	0.881	0.629	25.658
	LightGBM	1	Yes	0.882	0.623	13.678
		2	Yes	0.892	0.625	13.772
		6	Yes	0.895	0.633	14.180
		12	Yes	0.880	0.623	14.607
		24	Yes	0.883	0.629	13.266
		48	Yes	0.882	0.633	14.726
	XGBoost	1	Yes	0.939	0.660	14.991
		2	Yes	0.940	0.662	14.419
		6	Yes	0.946	0.667	14.038
		12	Yes	0.927	0.659	14.886
		24	Yes	0.930	0.665	13.774
		48	Yes	0.932	0.667	15.699
Linear Regression	Linear Regression	1	Yes	0.899	0.639	14.068
		2	Yes	0.889	0.632	13.146
		6	Yes	0.894	0.642	14.261
		12	Yes	0.908	0.654	13.880
		24	Yes	0.918	0.662	13.713
		48	Yes	0.944	0.681	13.766
Neural Network Model	Block RNN	None	Yes	0.935	0.664	79.089

ML Technique	ML Algorithm	Lag (hr)	Covariates	RMSE	MAE	Elapsed Time(s)
	N-BEATS	None	Yes	1.020	0.730	438.231
	TCN	None	Yes	0.889	0.618	76.483
	TFT	None	Yes	1.170	0.830	435.246
	Transformer	None	Yes	0.896	0.642	97.169
Without Covariates for Wind Speed Prediction						
Ensemble Technique	Random Forest	1	No	0.946	0.674	27.060
		2	No	1.011	0.729	28.485
		6	No	0.976	0.711	28.010
		12	No	0.954	0.698	23.515
		24	No	0.930	0.670	27.404
		48	No	0.926	0.667	24.790
Gradient Boosting	CatBoost	1	No	0.944	0.672	13.723
		2	No	0.940	0.674	13.156
		6	No	0.941	0.679	11.032
		12	No	0.930	0.675	13.649
		24	No	0.921	0.660	13.091
		48	No	0.917	0.653	12.900
	LightGBM	1	No	0.939	0.673	12.313
		2	No	0.940	0.677	11.761
		6	No	0.942	0.677	11.111
		12	No	0.941	0.679	11.051
		24	No	0.927	0.661	11.174
		48	No	0.928	0.663	10.940
	XGBoost	1	No	0.952	0.674	11.914
		2	No	0.973	0.697	10.709
		6	No	0.969	0.705	11.247
		12	No	0.987	0.725	11.486
		24	No	0.977	0.703	11.418
		48	No	0.994	0.708	13.072
Linear Regression	Linear Regression	1	No	0.948	0.678	10.789
		2	No	0.959	0.683	10.348
		6	No	0.958	0.683	23.013
		12	No	0.958	0.683	10.817
		24	No	0.932	0.665	20.832
		48	No	0.933	0.661	11.791
Neural Network Model	Block RNN	None	No	0.965	0.712	76.429
	N-BEATS	None	No	1.020	0.750	422.333
	TCN	None	No	0.936	0.675	75.866
	TFT	None	No	1.240	0.890	423.322
	Transformer	None	No	0.952	0.687	97.170

Appendix 2. Algorithm Performance for Wind Direction Prediction

ML Technique	ML Algorithm	Lag (hr)	Covariates	RMSE	MAE	Elapsed Time(s)
With Covariates for Wind Direction Prediction						
Ensemble Technique	Random Forest	1	Yes	97.085	68.547	26.522
		2	Yes	94.562	66.476	26.705
		6	Yes	94.455	66.793	26.819
		12	Yes	95.213	67.973	25.283
		24	Yes	95.276	69.066	26.040
		48	Yes	95.477	69.016	24.511
Gradient Boosting	CatBoost	1	Yes	94.518	67.467	15.745
		2	Yes	93.797	66.526	15.874
		6	Yes	94.061	67.960	15.616
		12	Yes	95.231	69.125	17.547
		24	Yes	94.645	69.062	17.672
		48	Yes	96.241	70.709	24.074
	LightGBM	1	Yes	95.309	68.146	14.439
		2	Yes	95.782	68.990	13.289
		6	Yes	94.083	67.303	12.869
		12	Yes	96.535	69.299	13.466
		24	Yes	96.927	69.838	12.378
		48	Yes	97.911	70.635	14.834
	XGBoost	1	Yes	100.090	71.786	14.833
		2	Yes	99.523	71.781	14.526
		6	Yes	97.787	70.627	14.460
		12	Yes	99.412	71.306	16.309
		24	Yes	101.160	72.997	14.049
		48	Yes	103.713	74.761	15.492
Linear Regression	Linear Regression	1	Yes	98.586	71.726	14.422
		2	Yes	98.489	71.604	13.655
		6	Yes	97.551	72.492	13.836
		12	Yes	99.798	74.575	14.367
		24	Yes	101.112	76.338	14.454
		48	Yes	104.851	79.514	14.140
Neural Network Model	Block RNN	None	Yes	97.135	69.100	77.793
	N-BEATS	None	Yes	113.520	79.300	431.233
	TCN	None	Yes	96.873	71.265	75.893
	TFT	None	Yes	131.840	88.120	422.265
	Transformer	None	Yes	115.719	97.987	101.692
Without Covariates for Wind Direction Prediction						
Ensemble Technique	Random Forest	1	No	106.340	74.949	23.393
		2	No	103.333	70.957	22.988
		6	No	97.491	69.208	23.311
		12	No	99.769	72.349	24.965
		24	No	99.647	75.370	24.651

ML Technique	ML Algorithm	Lag (hr)	Covariates	RMSE	MAE	Elapsed Time(s)
		48	No	98.912	75.167	23.026
Gradient Boosting	CatBoost	1	No	96.210	67.132	13.965
		2	No	96.694	65.989	12.940
		6	No	96.184	67.651	12.462
		12	No	100.697	72.071	11.494
		24	No	97.683	70.866	10.495
		48	No	97.938	71.663	13.582
	LightGBM	1	No	96.464	67.246	12.623
		2	No	96.951	66.727	10.538
		6	No	96.932	68.082	11.612
		12	No	98.990	71.400	11.938
		24	No	98.240	71.324	10.870
		48	No	97.461	71.232	11.812
	XGBoost	1	No	99.866	68.722	12.463
		2	No	102.035	69.271	13.936
		6	No	99.915	69.814	12.170
		12	No	105.022	74.689	12.652
		24	No	103.954	76.039	11.477
		48	No	104.525	77.431	13.145
Linear Regression	Linear Regression	1	No	98.817	70.103	11.566
		2	No	97.730	69.188	12.028
		6	No	98.220	69.680	11.805
		12	No	99.798	72.532	10.495
		24	No	100.011	73.402	11.766
		48	No	100.138	74.844	10.040
Neural Network Model	Block RNN	None	No	95.627	68.275	79.750
	N-BEATS	None	No	113.530	79.460	421.221
	TCN	None	No	98.262	69.995	75.518
	TFT	None	No	123.190	82.510	420.634
	Transformer	None	No	116.617	99.001	97.474

Appendix 3. Supplementary Information: Data Sources and Code Repositories

Information	Link
Data Sources	https://energydata.info/dataset/nigeria-solar-radiation-measurement-data
Code Repositories	https://github.com/tosmartak/darts-model-for-wind-speed-direction-prediction