Article

Not peer-reviewed version

Introduction to Wind Power Forecasting Using Hybrid VMD–GPR Models with Vedic Time Alignment

Krish Jalwal and Bhanu Prakash Joshi *

Posted Date: 11 May 2026

doi: 10.20944/preprints202605.0661.v1

Keywords: wind power prediction; vedic calendar; gregorian calendar; variational mode decomposition; Gaussian process regression; time series forecasting; renewable energy

Article

Introduction to Wind Power Forecasting Using Hybrid VMD–GPR Models with Vedic Time Alignment

Krish Jalwal [1] and Bhanu Prakash Joshi [2,*]

[1] B.Tech Student, Artificial Intelligence & Data Science, USAR, GGSIPU
[2] Assistant Professor, Department of Physics, USAR, GGSIPU, University School of Automation and Robotics (USAR), Guru Gobind Singh Indraprastha University, East Delhi Campus, Surajmal Vihar, Delhi – 110032, India
* Correspondence: bhanu.usar@ipu.ac.in

Abstract

This project checks methods in wind power forecasting by comparing Gregorian calendar based on seasonal alignments with the vedic lunisolar calendar parallely. Rather than using timestamps like most forecasting methods, this project seeks to determine whether periodic cycles based on nature's cosmos could reveal correlational patterns of wind activity surges and enhance accuracy. This study exploits the SOLETE dataset from SYSLAB, Denmark, which consists of 15 months of power generation alongside weather data. The dataset underwent processing with the CleanTS tool (an R package) and it was transformed into Gregorian and Vedic time frameworks. Within both time frameworks, the forecast approaches a hybrid forecasting model integrating "Variational Mode Decomposition (VMD) with Gaussian Process Regression (GPR)" was designed and assessed [11][12]. The Vedic forecasting approach is slightly better as it gives RMSE of 2.5519 and MAE of 2.0763, while the Gregorian forecasting approach gives RMSE of 2.6123 and MAE of 2.1424. The MAE correlation analysis over months revealed differing patterns within the two forecasting approaches with vedic giving better correlation than gregorian. This suggests that the Vedic calendar forecasting approach is better than the gregorian calendar system, which is based on natural cycles and is lunisolar, it is more accurate in capturing the chaotic signal of wind patterns than the arbitrary gregorian forecasting approach. This project helps in research, questioning the standard time representation in forecasting models which uses the gregorian timestamps and gives idea that if we put natural cycles through alternative calendar systems will it enhance the accuracy of energy predictions, potentially updating grid integration and operational planning.

Keywords: wind power prediction; vedic calendar; gregorian calendar; variational mode decomposition; Gaussian process regression; time series forecasting; renewable energy

I. Introduction

A. Topic Introduction

The need to predict winds is vital since this helps to keep the power grid consistent, stable and efficient. The use of wind for generating power continues to increase across the globe and hence forecasting will be increasingly vital. Traditional methodologies of wind power forecasting have used Gregorian calendars for forecasting wind behaviors. The Gregorian calendar may not necessarily be aligned to the real natural processes that affect the behavior of the winds. [4][5]

There is plenty of literature in relation to rainfall predictions and tides that indicates that most weather phenomena are more aligned to lunisolar and lunar cycles rather than the Gregorian calendar [1]. Rainfall and tidal prediction studies have been carried out using lunar calendars [1].

 doi:10.20944/preprints202605.0661.v1

Studies in such areas indicate that the relationship between lunar calendar and the prediction of such phenomena has resulted in significant gains in forecast accuracy [1]. This has helped us realize the importance of calendars in time-series models [1].

On that note, this research focuses on predicting the behavior of winds using a Vedic calendar [3]. The Vedic calendar, contrary to the Gregorian calendar, consists of months of equal length where there are seasonal adjustments [3]. Using the Vedic calendar may reveal some patterns in the winds behavior that the Gregorian calendar cannot bring out.

In this study, a hybrid method that utilizes the Variational Mode Decomposition (VMD) to analyze non-stationary signal wind power is combined with Gaussian Process Regression (GPR) [11][12]. In other words, two models are created based on these two types of calendar structures.

B. Problem Statement

Wind prediction is anything but simple. The nature of wind phenomena is characterized by its chaotic behavior and strong nonstationarity. Moreover, the stochasticity of wind complicates its integration into the large-scale power system. It affects the way dispatch is done, the quality of power generation, and stability throughout the power grid.

There are multiple obstacles when dealing with the wind power forecasting problem. Among them is the necessity of cleansing the time series. Inadequate handling of structural errors, missing values, incorrect timestamp, and outliers may affect the accuracy of the forecast significantly. To overcome the issues of structural errors, timestamp anomalies, missing values, and outliers in data, CleanTS, an R-package designed for data cleansing, can be used [2]. The preprocessing of wind power data results in clean data and improves input quality for the forecasting algorithm [2].

One of the core problems in time-series research is the representation of time. Time is divided into unequal periods using the Gregorian calendar, which is unrelated to solar heating or the movement of lunar tides affecting the atmosphere [1][9].

Calendars around the world are placed into different categories such as "as solar, lunar, or lunisolar based on their consideration of the Sun, Moon, or both." The Hindu Vedic calendar is lunisolar because it "incorporates both solar and lunar motions" [3]. A solar year and a lunar month are aligned by placing "an additional month (Adhikmaas) after 30 months ensuring all months are of equal 30 days" [3]. This design is intended towards the uniform segmentation of time to align with natural phenomenon like tides, seasons, winds, and other phenomena governed by celestial bodies [1][3].

The use of the Vedic calendar as a benchmark for time series analysis could be extremely beneficial. For the Hindu calendar unlike the Gregorian one, time could be subdivided into evenly distributed periods, thus eliminating assumptions in evaluations. It can also uncover concealed patterns by accounting for tithis which vary in length and have an impact on weather and wind for the day [3]. "These features make the Hindu Vedic calendar a promising substitute for time series forecasting, wind power prediction included" [1][3].

Linking the wind power statistics with Vedic lunisolar time allows for an analysis of the effect that periodic changes in the sun and moon have on the wind patterns. This approach improves not only the interpretation of the statistical results but also the accuracy of the forecasting process [1][9]. With this new method available, the future predictions will benefit greatly from using Vedic time stamps.

C. Objective of the Project

The objective of the current study can be defined as follows:

1. Timestamp Modification: Converting the timestamps of the wind power data set from "Gregorian calendar to the Vedic calendar" through using the VedicDateTime package [3].

2. Forecasting Model Construction: Constructing a forecasting model based on the integration of VMD (VMD-based decomposition of the wind power and meteorological factors) with GPR

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 May 2026 doi:10.20944/preprints202605.0661.v1

(Gaussian Process Regression) for predictions, based on a component-based decomposition strategy [11][12].

3. Evaluation: Analyzing the performance of VMD-GPR forecasting model under the Gregorian and Vedic calendar contexts as well as evaluating month-on-month correlation for a metric-driven approach.

It should be noted that the present project represents the latest research trend in exploring the influence of the Gregorian calendar on the time-series forecasting context.

II. Literature Survey

Wind power forecasting is divided into three groups depending on the period under consideration: "long-term (from current days to six days in advance), short-term (from an hour to one day in advance), and ultra-short-term (from five minutes to an hour in advance)" [4][8]. Each of them covers particular purposes of forecast generation and planning processes of wind energy production. Thus, in order to cope with particular requirements, a huge number of forecasting approaches were elaborated and are "grouped into four main categories: Time Series Analysis, Machine Learning Models, Deep Learning Models, and Hybrid Models" [4][5][9]. Each approach depends on data, such as historical data, meteorological data, and computation methods that increase its accuracy [5][8]. In most cases, the current approaches to wind power forecasting are represented graphically showing the newest methods used at present time [4][9].

This paper will be devoted to the systematization of methods of "long-term wind power forecasting. Long-term wind power forecasts are crucial for making decisions related to the maintenance and operation of wind turbines" [4][8]. The accuracy of short-term forecasts is very important, while in the case of long-term forecasting, it can be relatively low [5][9].

A. Time Series Analysis

"Time series analysis is considered a reliable foundation for predicting wind power generation." [4] The model makes use of historical data to determine the parameters needed and generate forecasts [4]. There is a notable method known as the Polynomial Autoregressive (PAR), which extends the basic idea of an autoregressive model using Volterra-series expansion [4]. Specifically, the degree-2 PAR model has proven to perform better for longer forecasting periods (more than 12 hours) than other nonlinear models including MLP, MLFF, ANN, and ANFIS [4]. The PAR models provide advantages in terms of reduced parameters and efficient computations. Evaluations using Global Energy Forecasting Competition 2012 dataset produced impressive results with NRMSE and NMAPE as measures of performance [4].

Besides the common forecasting purpose, time series analysis can be used for "ramp events prediction, i.e., sudden changes in wind power production that threaten the stability of the power grid" [6]. A hybrid model combines wind-power curve modeling using NWP data and a local correction model based on multi-prediction models [8]. Detection of the ramp events is made with high accuracy using the Swinging Door Algorithm [6].

B. Machine Learning Models

"Machine learning models are becoming more popular as long-term forecasters of wind power production due to their ability to identify nonlinear interactions between meteorological variables and wind power" [5][7]. "Support Vector Machines (SVM), Extreme Learning Machines (ELM), and various types of neural networks are some of the most prominent methods used" [5][7]. One case is a Conjugate Gradient Neural Network (CGNN) which improves the accuracy and training time of a Backpropagation Neural Network (BPNN) [4] by adding a CG optimization method [7]. This model uses real-time data from Chinese and Mongolian Wind Farms and incorporates meteorological data like air pressure, the temperature, humidity alongside wind speed and direction. In the same way, an SVM-based model had also been designed which used a hybrid kernel made out of polynomial

 doi:10.20944/preprints202605.0661.v1

and radial basis functions for local and global data correlation better capture. The parameters of the hybrid kernel were optimized to improve performance using a modified particle swarm optimization, and thus outperform standard models like ARMA, SVM with RBF, and Echo state neurons [5].

C. Deep Learning Models

"The ability of deep learning techniques to manage high-dimensional and nonlinear datasets has made them particularly useful for renovating long-term wind power data" [7]. One method is the use of stacked autoencoders (SAE) [7] to pull structural features from wind power data. In this method, data increments are created and processed through a two-layer autoencoder to obtain deep features which are later used by a cluster-based ensemble regression model. This approach achieved a prediction accuracy which is 12.63% better than models relying on statistical feature-based predictions [7]. One additional deep learning technique which stands out is the one which uses a "wavelet neural network (WNN) with a Morlet wavelet activation function" [8] for feature generation, selection, and forecasting. After feature extraction, the MDMRMR algorithm is applied to retrieve features considered most pertinent using the Maximum Dependence, Maximum Relevance, and Minimum Redundancy criteria, to train a shallow 2D CNN which is also optimized by particle swarm optimization [8].

D. Hybrid Model Approach

Through the combination of features of several techniques, hybrid models ensure better accuracy in predictions [9]. For instance, a hybrid model approach utilizing the combination of CEEMD technique along with the Sigma Point Kalman Filter can be more efficient when making long-term predictions because of the decomposition and reconstruction capabilities of the algorithm regarding the input series [9]. Daily patterns were extracted using wind power output and NWP (Numerical Weather Prediction) data through a clustering method known as k-means [8]. The Generalized Regression Neural Network (GRNN) [7] captures spatial dependencies in the forecast by training on the most similar data subsets [7]. The same principle was applied to enhance long term wind forecast accuracy using multi wind farms with a Bagging Neural Network (BaNN) [8] where IEMD (Improved Empirical Mode Decomposition) optimized k-means clustered BaNN [9]. A more sophisticated daily and hourly forecast model was created by combining VMD (Variational Mode Decomposition) with LSTM networks [11], outperforming EMD based models showing enhanced data stability and decline in noise with numerous layers of LSTMs trained for each decomposed mode.

E. Calendar Alignment with Natural Cycles

More modern attempts at time series forecasting in the context of climate change have shown that "aligning data with natural cycles, such as the lunar calendar, is more beneficial than following human calendars like the Gregorian" [1]. One investigation aimed at "forecasting rainfall in Bogor City using the Bi-directional Long Short-Term Memory (Bi-LSTM) model," where lunar-based data and Gregorian-based data were assessed with calendar data-based models to evaluate performance [1]. Daily rainfall data from the years 2000 to 2022 was aggregated to a monthly frequency on "both calendar systems. The research found that the lunar calendar significantly enhanced predictive accuracy relative to the Gregorian calendar with a lower MAPE of 14.82% for a three-month forecast relative to 35.12% using the Gregorian calendar" [1].

With the expanding scope of the synchronization of cycles concept, it only makes sense to consider how time series could be further improved by including in them the Vedic Hindu Calendar, whose structure combines solar years with both solar and lunar months [3]. The paper entitled "Natural Time-Series Analysis & Vedic Hindu Calendar System" [3] looks into how such a calendar

 doi:10.20944/preprints202605.0661.v1

system can facilitate forecasting in areas such as weather, climate, and energy due to uniform month lengths enabling the identification of cycles [3].

The research further illustrates the notion that time series analysis using the Vedic calendar generated stronger inter-month relationships which resonate better for accurate cyclical pattern detection. Those alterations sustain relationships with natural events such as the moon and sun's effect on weather and climate likely being essential to better forecasting models.

After going through the above literature survey, I will work on wind power prediction by forecasting the power values using a gregorian timestamp and by forecasting the power values using a vedic timestamps. We will compare how the both approaches have effect on the forecasting of wind power. Wind power being affected by moon because of ocean tides can have more effect if we use a lunar calendar. So vedic calendar being lunisolar can give higher accuracy in predicting the wind power. The inter-month relationship in vedic approach can give better results as it have fixed size months of 30 tithis, so I would find the month correlation in gregorian and vedic and compare them and try to find the reveal patterns in the data which are missed using the gregorian calendar.

III. Methodology

A. Data Preprocessing

"Data preprocessing entails converting raw data into a suitable format that can be used in analytical processing." Data preprocessing involves techniques such as filling out missing values, timestamp correction, outlier detection, noise reduction, and normalization [2]. Effective data preprocessing results in accurate and reliable data inputs for modeling, which in turn enhances performance [2].

1) CleanTS

"CleanTS is an open-source R software tool designed explicitly for time series data preprocessing" [2]. The software is automated and capable of detecting and correcting several common problems associated with time series data such as missing timestamps, duplicate timestamps, outliers, and temporal inconsistencies [2]. The software ensures that time series have optimal structural and statistical properties that ensure high-quality forecast models [2].

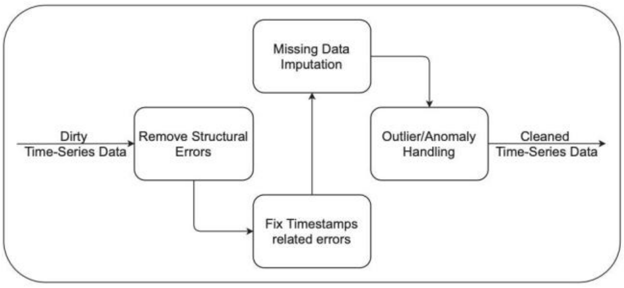


Figure 1. CleanTS.

2) Aspects of Data Preprocessing

Time Series data preprocessing involves the following critical components:

Outlier Detection and Removal: Removing extreme data points that differ substantially from other data points within the same sample [2].

Missing Timestamps: Detecting missing timestamps in a data set and inserting them where necessary to ensure consistency during time series data modeling [2].

Timestamp Alignment: Consistent and regular recording of data within a data set to guarantee accurate temporal analysis [2].

Feature Extraction: Conversion of "raw data into features that can be used" in modeling [2].

B. Variational Mode Decomposition (VMD)

 doi:10.20944/preprints202605.0661.v1

The Variational Mode Decomposition technique is an adaptive approach used to analyze the time series data and decompose it into a finite number of components, referred to as Intrinsic Mode Functions (IMFs)" [11]. Each IMF relates to a different frequency band [11].

1) Theory

"VMD formulates the decomposition as a constrained optimization problem. It seeks to find a set of modes whose bandwidths are minimized while reconstructing the original signal accurately." The decomposition is done in the frequency domain using Hilbert transforms and frequency shifting.

2) Formula

"The Variational Mode Decomposition (VMD) aims to decompose a signal f(t) into a set of K modes $\{u_k(t)\}$, each with a specific sparsity in the frequency domain" [11]. The decomposition is achieved by solving the following constrained variational problem [11]:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\}$$
$$\text{s.t.} \quad \sum_k u_k = f$$

Where:

f(t) = the original signal to be decomposed

$u_k(t)$ = the k^{th} mode (IMF)

ω_k = the center frequency of the k^{th} mode

* = convolution operator

δ(t) = Dirac delta function

$||\cdot||_2^2$ = L2 norm (squared energy of the analytic signal)

Subject to: $\Sigma_k u_k = f$. This process ensures that each mode is compact in the frequency domain and reconstructs the original signal when summed together [11].

C. Gaussian Process Regression (GPR)

"GPR is a non-parametric, Bayesian approach to regression. It assumes that the observed data are samples from a multivariate Gaussian distribution" and models the underlying function as a Gaussian process [12]. GPR is known for producing "not only point predictions but also confidence intervals around those predictions" [12].

1) Theory

"A Gaussian process defines a distribution over functions, fully specified by a mean function and a covariance function (kernel)" [12]. Using the available training data set, Gaussian Process Regression learns about the function range consistent with the observations. GPR produces a prediction mean and prediction variance for each test input, thus being suitable for applications where uncertainties must be accounted for [12].

2) Kernels

The kernel is also referred to as a covariance function that determines how similar any two points are within a Gaussian process regression [12]. Kernels determine how the predicted function would behave in terms of smoothness, periodicity, or complexity [12]. Popular kernels include:

RBF Kernel: smooth behavior assumption.

Matern Kernel: more complex behavior allowed.

Periodic Kernel: identifies repeating structures.

Linear Kernel: identifies linear relationships.

It is important to select and combine kernels to reveal hidden properties within datasets [12].

D. Evaluation Criteria

Evaluation criteria provide a systematic method for assessing how well predictive models perform in their ability to accurately predict results. In other words, evaluation criteria assess the

 doi:10.20944/preprints202605.0661.v1

deviation of the model predictions from the real data to allow an objective assessment and further improve the model [5][12]. Following criteria were used within this study:

1) Mean Absolute Error (MAE)

"Mean Absolute Error represents the average magnitude of prediction errors without considering the direction." This metric provides a linear score [5][12].

$$\frac{1}{n}\sum_{i=1}^{n}\left|y_i - x_i\right|$$

Where: x_i = Actual value, y_i = Predicted value, n = Number of observations.

2) Root Mean Square Error (RMSE)

"This measure is the square root of the average of the squared difference between predicted and observed values." The error is more sensitive to larger values than MAE [5][12].

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

3) Coefficient of Determination (R^2 Score)

"Coefficient of determination, or R^2, indicates how well the predictions correlate to the actual data." The higher the R^2 value, the closer predictions will be to the true data, where 1 implies perfect prediction, and 0 suggests that the model performs no better than the mean [5][12].

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

4) Correlation Heatmap

In order to examine consistency with respect to time, correlation heatmaps were generated based on the MAE calculated daily. Correlation coefficient between daily MAE values for each pair of months (Gregorian or Vedic) was computed, resulting in the symmetric matrix depicted as a heatmap [5][12]. The "correlation coefficient is determined by the following equation:" [5][12]

$$\rho_{xy} = \frac{cov(X,Y)}{\sigma_x \sigma_y}$$

Where: cov(X,Y) = Covariance between two variables. $\sigma_x\sigma_Y$ = Standard deviations of variables X and Y.

IV. Proposed Work

A. Flowchart

The second step includes transforming the raw wind power data into a forecastable time series. This paper seeks to analyze how the use of Vedic (lunisolar) time rather than Gregorian time impacts long-range wind power forecast accuracy [3].

Load Wind Power Data: The data set consists of wind power, humidity, temperature, wind speed, and wind direction. The data set includes timestamps in the Gregorian calendar. The data is loaded for further processing in both Python and R.

Data Extraction: From the timestamp in the Gregorian calendar, the day, month, year, and timezone information is extracted. These will be necessary when converting the Gregorian time to Vedic calendar through astronomical computation.

Extract Vedic Features – Tithi & Sunrise Time (in R): Tithi and sunrise time in the R environment will be computed from the Gregorian date and timezone information obtained from above using the VedicDateTime package [3].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 May 2026 doi:10.20944/preprints202605.0661.v1

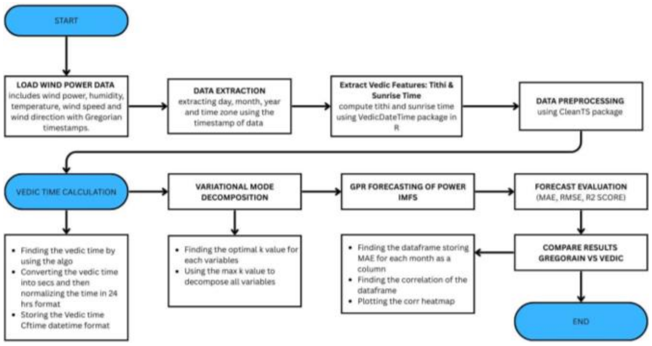


Figure 2. Flowchart.

Data Preprocessing (in Python): In preparation for decomposition and modeling, the raw wind power data set will be preprocessed with the help of CleanTS R package [2]. Outliers will be censored to zero and zeros will be imputed by averaging the next five values.

Vedic Time Calculation: The tithi output in CSV obtained from R in Step 3 will be imported into Python. The algorithm calculates the end time of the first tithi. Subsequent start and end times of each successive tithi will be computed. Conversion of Vedic time into seconds, normalization to a 24-hour clock, and storing into Cftime format is done for easy handling.

Variational Mode Decomposition (VMD): Each variable (Wind Power, Temperature, Humidity, Wind Speed, Wind Direction) is subjected to variational mode decomposition. This will break down each variable into various intrinsic mode functions (IMF) [11].

Determine Optimal k for Decomposition: An experiment will be carried out to determine the optimal decomposition of the IMFs. Based on the results, the maximum number of modes (k) will be chosen to perform uniform decomposition on each variable [11].

GPR Forecasting of Power IMFs: "Gaussian Process Regression (GPR) is performed on each of the wind power IMFs" and exogenous variables are inputted as their IMFs [12]. Monthly forecasting will be done.

Forecast Evaluation (MAE, RMSE, R^2 score): "Performance of the model will be assessed using metrics such as MAE, RMSE, and R squared" [5][12]. This will give insights into the accuracy of forecast values in the two calendar systems.

Compare Results – Gregorian-Time vs Vedic-Time: After performing all the steps above, the performance of the model between Gregorian-time and Vedic-time alignment will be compared.

B. Data Preprocessing

1) Outlier Handling

In this experiment, "the time-series plot of the data for power, temperature, humidity, wind speed, and wind direction was visually examined for the presence of outliers." From the time-series plots, abrupt variations that were inconsistent with the trends of the data were observed for the detection of outliers. Visual examination played an important role in validating whether the chosen methods of correction were suitable for the given data.

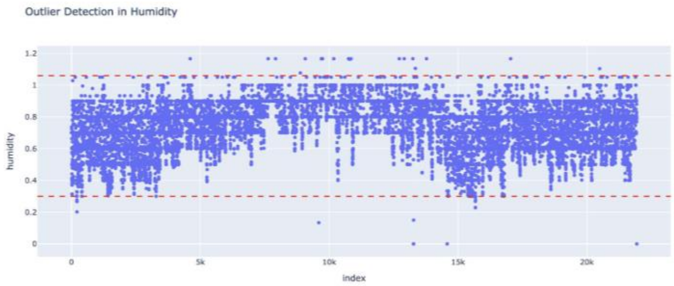


Figure 3. Outlier Detection.

When gaps are detected, CleanTS marks the gaps with temporal and spatial precision by replacing the outlier values with zeros. These zeros are replaced by averaging out five numbers before the zero placeholder and five numbers after the zero placeholder. Such a windowed averaging technique minimizes the risk of introducing significant distortion to the trend and upholds the structural order of the time sequence data. Moreover, in order to test the adequacy of the preprocessing steps, CleanTS offer pre-processed/post-processed contrasts allowing smoothness to be scrutinized together with the consistency of corrected time series data.

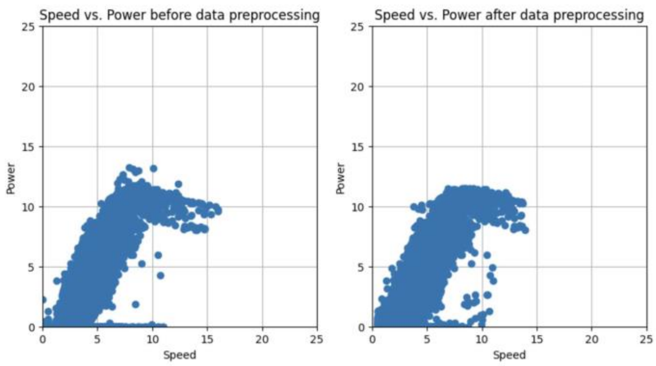


Figure 4. Data Before vs After Preprocessing.

2) Feature Extraction

To aid in the computation of Vedic calendrical components processes with the R package VedicDateTime, new features "such as Day, Month, Year, and Time Zone were created from the existing datetime column" [3]. The dt.day, dt.month and dt.year functions enabled the extraction of the individual components of the date from the timestamp. Furthermore, "a Time Zone column was created whereby rows for the winter months (November to March) were given a value of 1 and summer months (April to October) were given a value of 2 based on Denmark's seasonal time zone practice." The VedicDateTime R package was required to determine the accurate sunrise times and tithis, which enable the time series analysis of the Vedic calendar based on Vedic chronological components [3]; the extracted features were vital.

C. Vedic Time Computation

This dataset has a datetime entry which is in Gregorian calendar form. The dataset also contains the following power variables: power, temperature, wind direction, humidity, and wind speed. In order to organize the data in the form of Vedic structure, it was necessary to create additional features such as Day, Month, Year, and timezone based on the datetime feature.

The above mentioned features were important for using the R library called VedicDateTime in order to calculate the two most important features of the Vedic calendar which are the time of sunrise "and the tithi along with its ending time" [3]. The geographical coordinates of the data collecting site were also required as an input parameter.

"Using specific functions in the VedicDateTime package (as shown in the referenced figures), the tithi and its end time were computed for each date after converting it to julian day number" [3]. Below is the sample R code to find tithi:

```r
# Julian day number
jd <- 2459778
# Latitude, Longitude, and timezone of the location
place <- c(15.34, 75.13, +5.5)
tithi(jd,place)
#> [1] 20 20 55 35
```

Figure 5. VedicDateTime Tithi-1.

In some cases, where two tithis occurred in a single day, the output included eight digits representing both tithis and their respective end times. Below is the sample R code to find tithi:

```
tithi(gregorian_to_jd(17,6,2022),c(15.34, 75.13, +5.5))
#> [1] 18  6 11 26 19 26 59 58
```

Figure 6. VedicDateTime Tithi-2.

"Similarly, the sunrise time for each day was calculated" [3]. It also returns 4 numbers: first is the julian day number followed by the sunrise time. Below is the sample R code for finding sunrise time:

```
> library(VedicDateTime)
> sunrise(gregorian_to_jd(17,6,2022), c(55.7478, 12.0800, 1))
[1] 2459748       3       28       46
```

Figure 7. VedicDateTime Sunrise.

This enriched Vedic information was stored in a new DataFrame and exported for further use in Python.

"In the Python environment, a custom algorithm was developed to compute the Vedic time corresponding to each Gregorian timestamp. Initially, all rows were skipped until the end of the first tithi." Afterwards, the algorithm determined the time span of each tithi (usually ranging from 19 to 26 hours), normalized this time span by converting it into seconds, and finally mapped it into a 24hr clock format. In order to record these numbers precisely and in accordance with the Vedic calendar system, the Cftime data type was employed. This data type allows one to work with a calendar that consists of 360 days, a perfect fit for the Vedic 12 months consisting of 30 tithis [3].

1) Algorithm

The pseudocode of the Algorithm is given below:

Part 1: Initialize – Load dataset, import Vedic tithi and sunrise CSV from R, set initial pointers.

Part 2: Assign Vedic Timestamps – For each row, compute Vedic time based on tithi start/end boundaries.

Part 3: Store Vedic Timestamps in cftime.Datetime360Day format – Map normalized Vedic time into the 360-day Cftime calendar structure.

2) Calendar Shift

In order to analyze the temporal structure of Vedic months as compared to Gregorian months, a plot was created. It was aimed at demonstrating how data alignment is impacted by the Vedic lunisolar system which could be relevant for evaluating its effectiveness in pattern detection in time series forecasting.

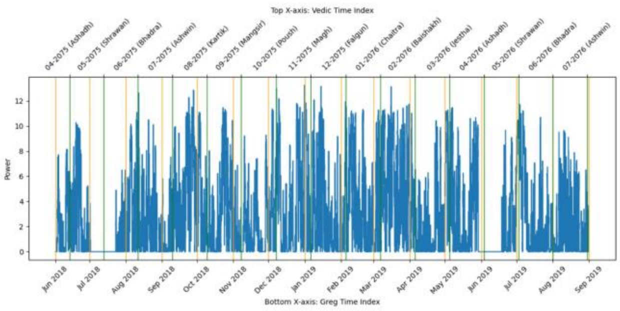


Figure 8. Time Shift.

D. Variational Mode Decomposition

"In this project, Variational Mode Decomposition (VMD) was applied to preprocess the time series data" for five important variables: Power, Temperature, Wind Speed, Humidity, and Wind

doi:10.20944/preprints202605.0661.v1

Direction [11]. By using VMD on each variable, the goal is to obtain distinct Intrinsic Mode Functions (IMFs) [11]. IMFs are components of the original signal that contain different frequency information, including the trend, cyclic pattern, and noise [11].

"The optimal number of modes (k) for each signal was determined through experimentation in order to achieve the best possible decomposition result." After obtaining the optimal k value for each variable, the highest one among them was selected and applied to all variables to maintain consistency. This guarantees the same number of IMFs for each signal, which is vital for organizing the input data in GPR.

In this case, each variable was processed independently using the VMD algorithm [11]. The hyperparameters alpha (bandwidth constraint), tau (noise-tolerance fidelity), and convergence tol were adjusted to optimize the results. As a result, the algorithm generates k IMFs for each signal, which were then stored as a DataFrame indexed by the datetime index of the original data. Such representation makes it possible to correctly align all decomposed signals and use each IMF as an input feature for predicting the corresponding power IMF.

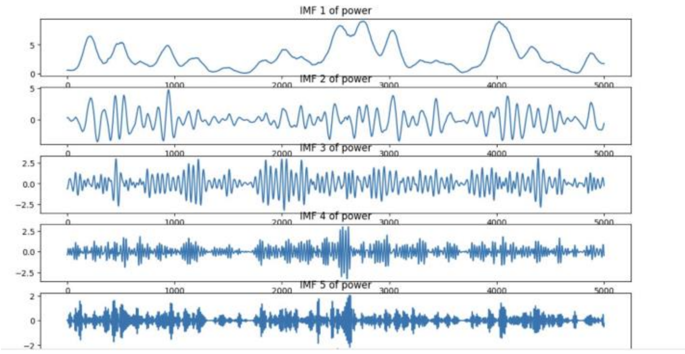


Figure 9. VMD.

E. Gaussian Process Regression (GPR)

In the current work, GPR is used for predicting the wind energy by considering each intrinsic mode function or IMF of the power signal as a separate entity [12]. The power signal together with the exogenous variable including wind speed, temperature, humidity, and direction of wind are firstly broken down into multiple IMFs using VMD (Variational Mode Decomposition) [11]. Each IMF represents a different frequency band of the power signal [11][12].

For each IMF of the power signal, a feature matrix is constructed using the corresponding IMFs of the exogenous variables. These form the input X, while the target output Y is the respective power IMF. This approach allows the "model to understand the contribution of each exogenous variable to the specific component of the power signal it is forecasting."

A Sparse Gaussian Process Regression model is then trained for each IMF [12]. To reduce computational complexity, 100 inducing points are used. The model uses a composite kernel consisting of "a Radial Basis Function (RBF) kernel and a Bias kernel." Training of the GPR model is done in two stages: in the first stage, the Gaussian noise variance is fixed, and the kernel parameters are optimized. In the second stage, the noise variance is unfixed and jointly optimized along with other parameters to improve model flexibility.

"Once trained, the GPR model is used to predict the value of the power IMF at the next time point" using the corresponding future values of the exogenous IMFs [12]. The forecasts for all power IMFs are then summed to reconstruct the final forecasted power value. The prediction value of power is then compared to its real value by applying evaluation criteria such as the root mean squared error (RMSE) [5][12].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 May 2026 doi:10.20944/preprints202605.0661.v1

F. Forecasting Approach

Within this study, a one-day-ahead forecasting scheme was employed, utilizing GPR models trained in advance [12]. The task involved assessing forecasting performance using two different time-referencing schemes – Gregorian and Vedic calendars [3]. Thus, the purpose was to discover the impact of different time referencing on renewable energy forecasting accuracy.

1. Forecasting Based on Gregorian Calendar: The forecast was conducted at 10:00 AM Gregorian time. Data from the past 60 days at 10:00 AM was used for training. Forecasting was performed for several months. Forecasting accuracy was assessed in terms of Mean Absolute Error (MAE). Grouping results according to Gregorian months was performed to investigate correlations.

2. Forecasting Based on Vedic Calendar: Forecasts were performed at 5 hours after sunrise according to Vedic time [3]. Training was performed based on data collected over the last 60 days in the same Vedic-time period. Predictions were conducted in order to predict next day data at the same Vedic-time period. MAE assessment was followed by grouping results according to Vedic months.

This comparative framework enabled a novel analysis of forecasting performance from both Gregorian and Vedic calendar perspectives, offering insights into temporal structures best suited for renewable energy prediction.

V. Results

A. Model Training

To assess the performance of the hybrid VMD-GPR model, a rolling one-day ahead forecasting scheme was implemented. The main idea of the scheme was to predict the wind power value for a particular day based on the training performed on data for the past 60 days. This procedure was iterated over 30 consecutive days.

During each iteration: Data for 60 days, aligned to a certain reference time (Gregorian or Vedic), is collected. As inputs used the following parameters: wind direction, temperature, humidity, and wind speed. The output was the power value for each IMF. Each IMF is predicted separately based on a Sparse Gaussian Process Regression (GPR) model [12] trained on a combined kernel: RBF + Bias.

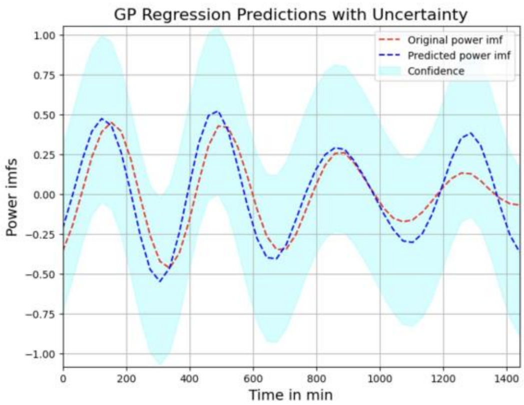
The training was carried out with help of the train_sparse_gp function, performing the following actions:

1. Initialization of the SparseGP model, specifying 100 inducing points.

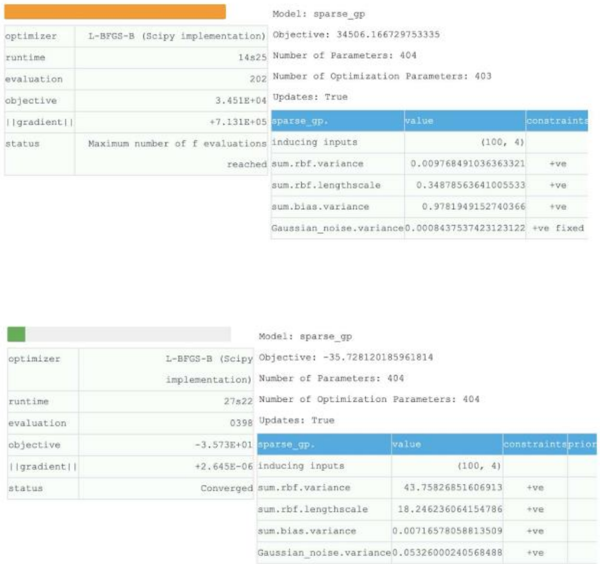
2. Fixation of the noise variance, then performing an optimization of the model (for 200 iterations).

3. Unfixing the noise variance and optimizing the model (up to 5000 iterations).

After completing the training: The GPR predicts IMF for the next day [12]. All the IMFs' predictions are added up to form the final forecast for the wind power. The accuracy of the forecast is checked by several criteria: RMSE, MAE [5][12]. All the operations mentioned above are applied to each new day in the testing period, advancing the date each time.



Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 May 2026 doi:10.20944/preprints202605.0661.v1



optimizer	L-BFGS-B (Scipy implementation)
runtime	14s25
evaluation	202
objective	3.451E+04
||gradient||	+7.131E+05
status	Maximum number of f evaluations reached

Model: sparse_gp
Objective: 34506.166729753335
Number of Parameters: 404
Number of Optimization Parameters: 403
Updates: True

sparse_gp.	value	constraints
inducing inputs	(100, 4)	
sum.rbf.variance	0.009768491036363321	+ve
sum.rbf.lengthscale	0.34878563641005533	+ve
sum.bias.variance	0.9781949152740366	+ve
Gaussian_noise.variance	0.0008437537423123122	+ve fixed

optimizer	L-BFGS-B (Scipy implementation)
runtime	27s22
evaluation	0398
objective	-3.573E+01
||gradient||	+2.645E-06
status	Converged

Model: sparse_gp
Objective: -35.728120185961814
Number of Parameters: 404
Number of Optimization Parameters: 404
Updates: True

sparse_gp.	value	constraints	prior
inducing inputs	(100, 4)		
sum.rbf.variance	43.75826851606913	+ve	
sum.rbf.lengthscale	18.246236064154786	+ve	
sum.bias.variance	0.00716578058813509	+ve	
Gaussian_noise.variance	0.05326000240568488	+ve	

Figure 10. Model Training.

B. Gregorian Calendar-Based Forecasting Results (10 AM Forecast)

The forecast for wind energy in this methodology was conducted each day at 10:00 AM based on data that was organized according to the Gregorian calendar system. The training of the hybrid model VMD-GPR used the sliding windows methodology as follows:

Training window: Previous 60 days of data.

Forecast target: Wind power at 10 AM of the next day.

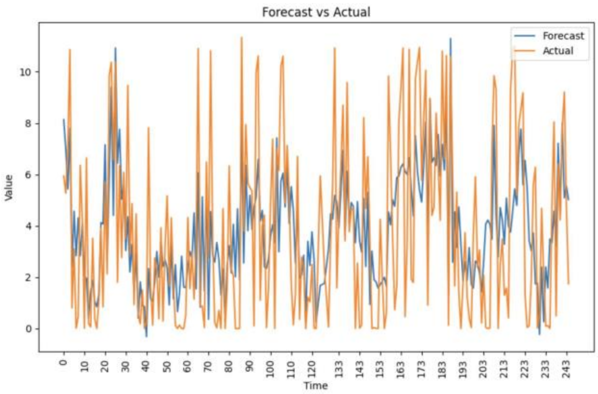


Figure 11. Gregorian Forecast vs Actual Plot.

1) Forecast Storage and Daily Evaluation

For each forecast: The predicted power values were recorded along with the actual values observed in a DataFrame. The Mean Absolute Error (MAE) was calculated for each predicted timestamp. Each day's MAE values were grouped into a matrix of size (31 × N months). This allowed the assessment of the accuracy of forecasts over days and months.

2) Correlation Heatmap Analysis

The correlation matrix of daily MAE for each Gregorian month was calculated and presented in a heatmap form. In particular: The axes correspond to months (e.g., 10 = October, 11 = November, and so on). The cells denote the Pearson correlation coefficient between two months' MAE curves. High correlation (dark red color) means a comparable pattern of daily error values.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 May 2026 doi:10.20944/preprints202605.0661.v1

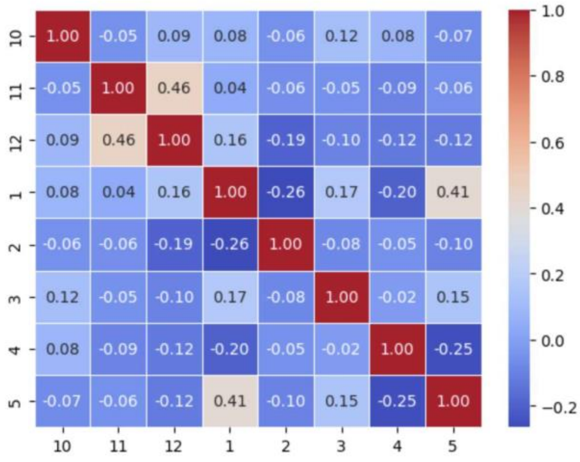


Figure 12. Gregorian Correlation Heatmap.

3) Major Findings

There exists a fairly high correlation (e.g., 0.46) between months such as November and December and December and January, implying temporal similarity. Yet, most months have a low or even negative correlation, implying irregular forecasting patterns over time. Months such as February, March, and April have low or even poor correlation with other months.

Table 1. Gregorian Results.

Metrics	Result
RMSE	2.6123
MAE	2.1424
R^2 Score	0.4414

4) Conclusion

Even though the Gregorian model makes fairly precise predictions, the variability within the months implied by the correlation matrix shows that the model lacks generalizing ability. Hence, it is not suitable for long-term or seasonal forecasting.

C. Vedic Calendar-Based Forecasting (Forecasting 5 Hours After Sunrise)

The method involves using Vedic calendar where forecasts are made on daily basis taking wind power 5 hours after sunrise into account, thus accounting for time according to tithis which have been traditionally used for time measurements in ancient India.

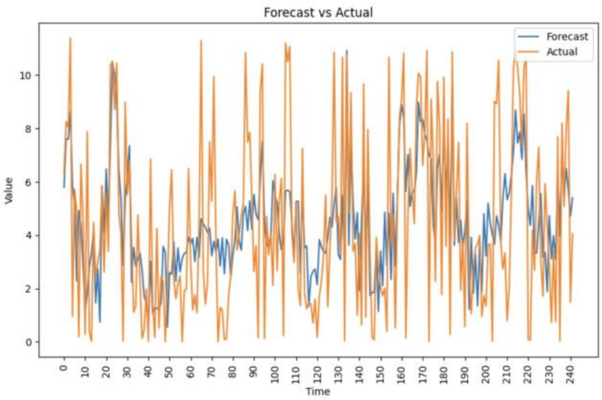


Figure 13. Vedic Forecast vs Actual Plot.

1) Data Processing and Analysis

Prediction target: Wind power 5 hours after sunrise calculated via the VedicDateTime module. Error aggregation: MAE calculated for every tithi and then aggregated according to Vedic month. Output matrix dimensions: 30 × 12. This means the resulting error will be presented in matrix form.

2) Correlation Heatmap of Vedic Months

A correlation matrix was created using the monthly tithi-wise MAE values. Axes: Vedic months (e.g., Chaitra = 1, Vaishakha = 2, etc.). Cell values: Pearson correlations of error profiles.

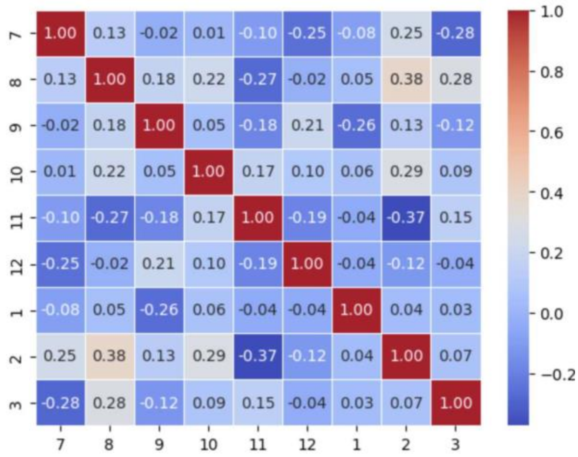


Figure 14. Vedic Correlation Heatmap.

3) Important Points

In contrary to Gregorian calendar, Vedic heatmap has a fairly high correlation value for all months due to more predictable model behavior. For example, Vaishakha (2) has 0.38 correlation with Shravana (8), and such behavior can be seen in other pairs of months. It is important that there are no dark colors on the heatmap, which means that there are no strong negative correlations, in contrary to Gregorian heatmap.

Table 2. Vedic Results.

Metrics	Result
RMSE	2.5519
MAE	2.0763
R^2 Score	0.4224

4) Interpretation

The Vedic-based model performs slightly better in terms of metrics (lower MAE and RMSE values), but, more importantly, has higher robustness and consistency of forecasting for different months. This is demonstrated by the correlation heatmap, showing relatively stable behavior for the whole period. Lack of sharp drops makes the model even more reliable.

D. Discussion on Results

Vedic-based model has lower MAE and RMSE metrics values, indicating higher precision. Month-wise correlation matrix demonstrates consistency of the Vedic model, in contrary to

Gregorian one, having disruptive jumps with high correlations in some months. This suggests that Vedic calendar splitting fits better for forecasting long-term wind patterns.

VI. Simulation Environment

A. Hardware Requirements

1) CPU Cores: Multi-core processor is necessary for parallel computations and quick performance of decomposition and forecasting algorithms. Recommended: A processor with at least 4 to 8 cores (e.g., Intel Core i7, AMD Ryzen 7); 16+ cores is optimal for best possible performance.

2) RAM: Minimum: 8 GB. Recommended: 16 GB and above (to efficiently handle large datasets and multiple components of decomposed series).

3) Storage: SSD with 256 GB or higher to ensure fast read/write operation and efficient access to big data and processing of Python and R scripts.

4) Internet Connection: Necessary in the initial stage to download libraries. Necessary for execution of the code and notebooks on cloud platforms such as Kaggle.

B. Software Requirements

1) Operating System: 64-bit Windows 10/11, macOS, or Ubuntu Linux 20.04 and above for full compatibility with Python and R languages.

2) Programming Languages: Python 3.8+ used for data preprocessing, VMD decomposition, GPR modeling, forecasting, and visualizations. R 4.0+ used for conversion of timestamps to Vedic lunisolar calendar timestamp using vedicdatetime package.

3) Python Libraries: pandas, numpy, scipy – for data manipulation, numerical computing, and imputation. matplotlib, seaborn, plotly – for time series plots, forecast accuracy, and visualizations. vmdpy – for performing VMD decomposition. GPy – for implementing GPR wind energy forecasting model.

4) R Packages: CleanTS – for preprocessing dataset and handling missing timestamps, outliers, and erroneous timestamps. vedicdatetime – for calculating timestamps according to Vedic lunisolar calendar.

5) Development Environment / IDEs: Kaggle – the development and execution environment used for Python-based models including forecasting. RStudio – used for R programming for Vedic time calculations and preliminary data preprocessing.

C. Screenshot of Simulation

Figure 15. Screenshot of Kaggle 1.

 doi:10.20944/preprints202605.0661.v1

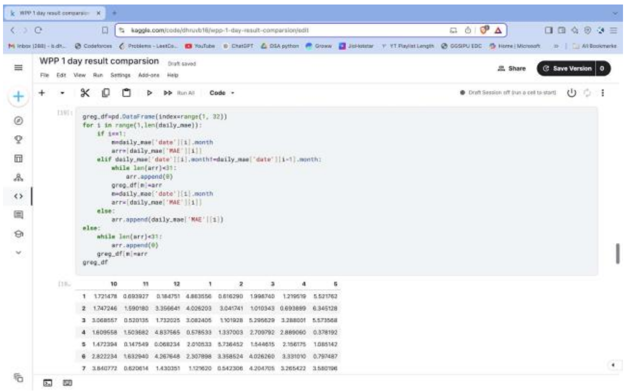


Figure 16. Screenshot of Kaggle 2.

VII. Conclusion

This research demonstrated the outstanding abilities of a hybrid algorithm that used "Variational Mode Decomposition (VMD) in combination with Gaussian Process Regression (GPR)" [11][12]. Capturing of complex nonlinearities as well as temporal patterns in wind power data using this approach was successful as a result of signal decomposition and probabilistic regression; the weather/climatic variables of interest were also adequately captured.

The following forecasting approaches were considered in the research:

1. Gregorian-Based Forecasting: An algorithm aimed at precise forecasting of wind power at 10 AM based on Gregorian calendar.
2. Vedic-Based Forecasting: A predictive algorithm grounded on Vedic chronology, designed for predicting wind power for a period of 5 hours after sunrise on the basis of Vedic tithi and month.

For both algorithms, VMD was utilized to derive the features relevant for this task from the signal of interest (wind power) as well as from external predictors in terms of decomposition into Intrinsic Mode Functions (IMFs) which included the signal's trends, seasonality and additional noise components [11]. Separate GPR models were constructed for each IMF, and then their forecasts were combined to construct the final one [12].

A. Key Findings

The Gregorian-based model produced MAE = 2.1424, RMSE = 2.6123, and $R^2 = 0.4414$. Slightly better results were attained by the Vedic-based model: MAE = 2.0763, RMSE = 2.5519, $R^2 = 0.4224$. Significantly, the MAE correlation heatmap showed a higher consistency of forecasts in time for the Vedic-based approach compared to Gregorian-based one.

B. Future Work

As the Vedic forecasting method provides higher forecasting accuracy than the Gregorian forecasting method according to the correlation heatmap, it demonstrates the possibility of applying the Vedic timestamp in enhancing forecast accuracy.

The forecasting technique that utilizes the Vedic timestamp could be applied to other forecasting areas like predicting solar power, demand energy forecasting, and rainfall prediction, as the Vedic calendar is lunisolar, and time alignment with natural phenomena might increase model accuracy.

More sophisticated deep learning algorithms such as "Bi-directional Long Short-Term Memory (Bi-LSTM)" could replace the GPR algorithm, taking advantage of its long-term memory capabilities and increasing the accuracy of predictions.

The development of an R package that converts Gregorian timestamps to Vedic timestamps could help analysts and researchers integrate the Vedic calendar in their workflows.

Proposing a standard format of the Vedic time could assist analysts and researchers in recording data in a structured way, adhering to the Vedic timeline.

 doi:10.20944/preprints202605.0661.v1

References

1. G. Darmawan, B. Handoko, D. Y. Faidah, and D. Islamiaty, “Improving the Forecasting Accuracy Based on the Lunar Calendar in Modeling Rainfall Levels Using the Bi-LSTM Method through the Grid Search Approach,” The Scientific World Journal, vol. 2023, pp. 1-8, Dec. 2023, doi: 10.1155/2023/1863346.
2. M. K. Shende, A. E. Feijoo-Lorenzo, and N. D. Bokde, “cleanTS: Automated (AutoML) tool to clean univariate time series at microscales,” Neurocomputing, vol. 500, pp. 155-176, May 2022, doi: 10.1016/j.neucom.2022.05.057.
3. N. D. Bokde, P. K. Patil, S. Sengupta, M. Sawant, and A. E. Feijoo-Lorenzo, “VedicDateTime: An R package to implement Vedic calendar system,” Multimedia Tools and Applications, vol. 83, no. 11, pp. 32141-32157, Sep. 2023, doi: 10.1007/s11042-023-16553-w.
4. O. Karakuş, E. E. Kuruoglu, and M. A. Altinkaya, “One-day ahead wind speed/power prediction based on polynomial autoregressive model,” IET Renewable Power Generation, vol. 11, no. 11, pp. 1430-1439, Jun. 2017, doi: 10.1049/iet-rpg.2016.0972.
5. D. Lee and R. Baldick, “Short-Term wind power ensemble prediction based on Gaussian processes and neural networks,” IEEE Transactions on Smart Grid, vol. 5, no. 1, pp. 501-510, Sep. 2013, doi: 10.1109/tsg.2013.2280649.
6. J. Li, G. Liu, and S. Zhang, “Smoothing ramp events in wind farm based on dynamic programming in energy internet,” Frontiers in Energy, vol. 12, no. 4, pp. 550-559, Nov. 2018, doi: 10.1007/s11708-018-0593-8.
7. S. Tasnim, A. Rahman, A. M. T. Oo, and Md. E. Haque, “Autoencoder for wind power prediction,” Renewables Wind Water and Solar, vol. 4, no. 1, Dec. 2017, doi: 10.1186/s40807-017-0044-x.
8. L. Dong, L. Wang, S. F. Khahro, S. Gao, and X. Liao, “Wind power day-ahead prediction with cluster analysis of NWP,” Renewable and Sustainable Energy Reviews, vol. 60, pp. 1206-1212, Mar. 2016, doi: 10.1016/j.rser.2016.01.106.
9. O. Abedinia, M. Lotfi, M. Bagheri, B. Sobhani, M. Shafie-Khah, and J. P. S. Catalao, “Improved EMD-Based Complex Prediction Model for wind power Forecasting,” IEEE Transactions on Sustainable Energy, vol. 11, no. 4, pp. 2790-2802, Feb. 2020, doi: 10.1109/tste.2020.2976038.
10. N. D. Bokde, P. K. Patil, S. Sengupta, M. Sawant, and A. E. Feijoo-Lorenzo, “VedicDateTime: An R package to implement Vedic calendar system,” Multimedia Tools and Applications, vol. 83, no. 11, pp. 32141-32157, Sep. 2023, doi: 10.1007/s11042-023-16553-w.
11. Q. Liu, Y. Xu, L. Zhang, and C. Liang, “Research on heart Sound Signal Denoising Algorithm based on variational mode decomposition and Wavelet threshold,” Journal of Computer and Communications, vol. 09, no. 10, pp. 110-121, Jan. 2021, doi: 10.4236/jcc.2021.910007.
12. H. Jin, L. Shi, X. Chen, B. Qian, B. Yang, and H. Jin, “Probabilistic wind power forecasting using selective ensemble of finite mixture Gaussian process regression models,” Renewable Energy, vol. 174, pp. 1-18, Apr. 2021, doi: 10.1016/j.renene.2021.04.028.