MAJOR PROJECT REPORT

ON

"WIND POWER PREDICTION"

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
In
Artificial Intelligence & Data Science
[2021-25]

Submitted By:
B.Dhruv
03519011921

Guided By:
Dr. Manisha Parlewar
Assistant Professor

UNIVERSITY SCHOOL OF AUTOMATION AND ROBOTICS, GGSIPU

GURU GOBIND SINGH
INDRAPRASTHA
UNIVERSITY
NEW DELHI

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY EAST DELHI CAMPUS, SURAJMAL VIHAR

DECLARATION BY THE CANDIDATE

I hereby declare that the work, which is being presented in this project entitled **"WIND POWER PREDICTION",** is an authentic record of my work under the supervision and guidance of **Dr. Manisha Parlewar, Assistant Professor, University School of Automation of Robotics, GGSIPU.**

This project was undertaken as a part of the major project report for the partial fulfillment of the Bachelor of Technology (Artificial Intelligence and Data Science).

I have not submitted the matter embodied here in this project for the award of any other Degree.

B.Dhruv
03519011921
B. Tech (AI&DS)

CERTIFICATE

This is to certify that this MAJOR PROJECT REPORT "**WIND POWER PREDICTION**" is submitted by "**B.Dhruv**" who carried out the project work under my supervision. I approve this project for submission to the Bachelor of Technology (B.Tech., AIDS) at the University School of Automation and Robotics (USAR), Guru Gobind Singh Indraprastha University, Delhi.

Date:

Dr. Manisha Parlewar
Assistant Professor
USAR, GGSIPU

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout this project. First and foremost, I am deeply thankful to **Prof. A. K. Saini, Director,** East Campus, GGSIPU, for providing the necessary academic environment and resources that made this project possible. I would also like to extend my heartfelt gratitude to **Prof. Arvinder Kaur**, Dean, USAR, for her constant encouragement and support during this academic journey.

I sincerely thank **Prof. Kriti Batra**, Program Coordinator, for his/her valuable guidance in structuring the project and ensuring smooth progress. I am especially grateful to **Prof. Chiranjit Pal**, Teacher-in-Charge, for the consistent academic supervision, helpful insights, and feedback that significantly enhanced the quality of my work.

A very special thanks to my project mentor, **Dr. Manisha Parlewar**, for her unwavering support, expert guidance, and constant motivation throughout every phase of this project. Their dedication and mentoring were crucial to the completion of this work.

Lastly, I would like to thank all my faculty members, friends, and family who contributed in any way toward the success of this project.

Thanking You

B.Dhruv

03519011921

PLAGIARISM REPORT

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

SIMILARITY REPORT

turnitin Page 2 of 53 - Integrity Overview Submission ID trn:oid:::1:3253257737

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- **124** Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- 2 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
9% Publications
3% Submitted works (Student Papers)

turnitin Page 2 of 53 - Integrity Overview Submission ID trn:oid:::1:3253257737

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The project investigates an unorthodox method in wind power forecasting by comparing Gregorian calendar-based seasonal alignments with the Vedic lunisolar calendar. 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.

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 contributes to research, questioning standard time representations in forecasting models which uses the gregorian timestamps and suggests that if we incorporate natural cycles through alternative calendrical systems will enhance the accuracy of renewable energy predictions, potentially improving grid integration and operational planning.

Keywords: Wind Power Prediction, Vedic Calendar, Gregorian Calendar, Variational Mode Decomposition, Gaussian Process Regression, Time Series Forecasting, Renewable Energy

CHAPTER 1: INTRODUCTION

1.1 Topic Introduction

Forecasting wind power is critical for ensuring the dependability, consistency, and operational effectiveness of power grids, particularly as the use of renewable energy sources ramps up worldwide. Traditionally, the bulk of forecasting methods—predicting wind power inclusive—operate based on the Gregorian calendar. This 'solar' calendar, however, does not always correspond with underlying atmospheric and celestial rhythms that dictate the wind's behavior.

Research in adjacent fields such as rainfall forecasting [1] demonstrates that numerous natural phenomena bear a closer relationship to a lunar or lunisolar calendar instead of maimed subdivisions of time such as the Gregorian calendar. It is now well-known that rainfall and tidal occurrences coincide with several phases of the moon, and this correlation has led to explorations of forecasting these events using lunar calendars. The findings from these studies emphasized the enhancement of prediction precision facilitated by the alignment of temporal analysis with natural cycles, which underscores the utter importance of calendar structure in time series modeling.

Drawing from this perspective, our study tries to take a different approach of forecasting wind power by employing the Vedic calendar, which combines both the solar and lunar systems. Unlike the Gregorian calendar, the Vedic calendar has standard equal-length months with periodic corrections for synching with celestial movements which is done to provide a more natural division of time. This could reveal some wind behaviour periodicities that would otherwise be hidden behind a Gregorian-based analysis.

As part of this project, we construct a hybrid model that incorporates Variational Mode Decomposition (VMD) for separating non-stationary wind power signals with Gaussian Process Regression (GPR) for prediction. We analyze and evaluate the forecasting accuracy of this model on datasets based on the Gregorian and Vedic calendars.

1.2 Problem Statement

Wind forecasting is by no means an easy task, as the dynamics of wind are chaotic and highly nonstationary. The random nature of winds poses severe complications for large-scale grid integration, impacting dispatch operations, power quality, and stability of the grid.

Wind power prediction has a multitude of challenges to address, one of which is maintaining quality and consistency in time series data. Forecasting precision can be severely hindered due to structural errors, missing values, incorrect timestamps, and anomalies present in the raw wind power data. To mitigate structural error and timestamp anomalies as well as missing data imputation and outlier handling, the CleanTS R package [2] can be utilized for efficient data cleaning. Effective wind power data preprocessing translates to improving the input data quality for the predictive models, making them more robust and accurate.

The representation of time, itself, is among the most elementary challenges in time series analysis. The Gregorian calendar splits up time into arbitrary months of unequal lengths, serving no purpose for the atmospheric or astronomical cycles. These timekeeping methods can muddle important periodic patterns associated with wind speed and power generation, such as those governed by both solar heating and lunar tidal forces.

Calendars around the world are placed into different categories such 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. 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. 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.

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. These features make the Hindu Vedic calendar a promising substitute for time series forecasting, wind power prediction included.

Aligning wind power data with the Vedic lunisolar calendar allows us to analyze solar and lunar periodical influences, greatly improving the interpretability and accuracy of forecasting models. This innovative approach opens new avenues in energy forecasting. It motivates us to use Vedic timestamps for forecasting natural energy.

1.3 Objective of the Project

The primary objectives of this project are as follows:

1. **Timestamp Adjustment:** Using the VedicDateTime package, adjust the timestamps of the wind power dataset from the Gregorian calendar to the Vedic calendar.
2. **Model Creation:** Wind power will be forecasted using a hybrid model VMD (wind power and meteorological parameters were decomposed) and GPR (Gaussian Process Regression employed for prediction) with the component-based decomposition approach.
3. **Evaluative Analysis:** Analyze and evaluate the forecasting accuracy of the VMD-GPR model in both Gregorian and Vedic calendar systems and assess correlations over several months for metric-based analysis.

This project is part of an emerging research area that explores the implications of the Gregorian calendar on time series forecasting. We integrate the domain of calendar science and modern machine learning alongside advanced signal decomposition techniques to determine the impact of natural cyclic alignment—via the Vedic Calendar—on the accuracy of wind power forecasting.

CHAPTER 2: LITERATURE SURVEY

Based on the lead time, wind power prediction is classified into three forecasting horizons: long-term (day up to six days ahead), short-term (one hour to a day ahead), and ultra short-term (five minutes to one hour ahead). Each of these forecasting horizons fulfills specific operational as well as wind energy systems planning requirements. To tackle the problems unique to each horizon, researchers formulated numerous prediction methodologies which can be classified into four major categories: Time Series Analysis, Machine Learning Models, Deep Learning Models, and Hybrid Approaches. These methodologies make use of historical data, meteorological data, and computation techniques to make the prediction more precise. The figure below summarizes recently published methods within each prediction horizon, providing an overview of the different approaches documented in the literature.

I have systematically investigated WPP methods based on Long-term horizon. Long-term wind power prediction plays a crucial role in strategic planning activities such as wind turbine maintenance and energy management. While short-term forecasts demand high accuracy, long-term forecasts can tolerate comparatively lower precision but must effectively capture trends and patterns over extended horizons.

2.1 Time series analysis

Time series analysis is a well-established approach for wind power prediction, where historical data is used to estimate model parameters and generate forecasts. One notable method is the Polynomial Autoregressive (PAR) model [3], which extends the traditional AR model using Volterra series expansion. In particular, a degree-2 PAR model has shown superior performance for long-term horizons (exceeding 12 hours) in comparison with other nonlinear models like MLP, MLFF, ANN, and ANFIS. PAR models not only require fewer parameters but are also computationally efficient. Evaluations using datasets from the Global Energy Forecasting Competition 2012 have demonstrated strong performance metrics such as NRMSE and NMAPE. Moreover, time series approaches have also been applied to ramp event prediction—sudden changes in wind power output that can disrupt grid stability. For this, a hybrid method combining wind power curve modeling using NWP data and a local correction model based on multivariate prediction algorithms has been proposed. The ramp detection acheives high accuracy in detecting of ramp events by leveraging the Swiging Door Algo.

2.2 Machine learning models

The use of Machine learning algorithms as a tool for carrying out long-term wind power forecasting has been on the rise lately due to their ability to understand complex, nonlinear relationships between the meteorological data and the power generated. Support Vector Machines (SVM), Extreme Learning Machines(ELM), and various types of neural networks are some of the most prominent methods used. 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. 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 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. In another study, SVMs fused with the Swinging Door [5] Algorithm optimized with Genetic Algorithms were used for the detection of wind ramp events by first finding the best ramp and non-ramp window.

2.3 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. One method is the use of stacked autoencoders (SAE) [6] 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. One additional deep learning technique which stands out is the one which uses a wavelet neural network (WNN) with a Morlet wavelet activation function 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. This technique has performed well for multi-time horizon prediction (1 hour, 1 day, and 2 days ahead) with two datasets.

2.4 Hybrid Approach

By merging the various strengths of several methods, hybrid models further enhance accuracy in prediction. For instance, the model which integrates Complementary Ensemble Empirical Mode Decomposition (CEEMD) with a Sigma Point Kalman Filter is able to perform long-term forecasting predicting with sophisticated decomposing and reconstruction of input series steady components. Daily patterns were extracted using wind power output and NWP (Numerical Weather Prediction) data through a clustering method known as k-means. The Generalized Regression Neural Network (GRNN) [7] captures spatial dependencies in the forecast by training on the most similar data subsets. 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. A more sophisticated daily and hourly forecast model was created by combining VMD (Variational Mode Decomposition) with LSTM networks, outperforming EMD based models showing enhanced data stability and decline in noise with numerous layers of LSTMs trained for each decomposed mode.

2.5 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. 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 signicantly 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. The study concluded that some constituents of the environment such as water rely heavily on lunar cycles for control; thus, weaved frameworks regulated by such rhythms are more precise in depicting certain environmental elements.

As the idea of aligning data with natural cycles expands, the potential for improving time series analysis with the Vedic Hindu calendar, which integrates years with solar months and lunar months with solar years, comes to mind. The study titled Natural Time-Series Analysis and Vedic Hindu Calendar System[9] investigates the calendrical Vedic's forecasting applications with a focus on weather, climate, and energy forecasting. The Vedic calendar's structure balances solar and lunar influences, providing uniform month durations which aid in periodic pattern recognition. Unlike the Gregorian calendar which allocates month length without any regard to natural phenomena, the Vedic system permits more relevant time divisions that facilitate the exposure of correlations otherwise masked by Gregorian months.

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. The paper advocates constructing bespoke tools and methodologies around forecasting specialized to the Vedic calendar which can boost climate science, energy, and other nature-dependent time series analysis.

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 use a lunar calendar. So vedic calendar being lunisolar can give higher accuracy in predicting the wind power. The inter-month reslationship 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.

CHAPTER 3: METHODOLOGY

3.1 Data Preprocessing

Data preprocessing refers to the process of transforming raw data into a clean and usable format for analysis or modeling. This includes handling missing values, correcting timestamps, removing outliers, smoothing noisy data, and aligning time intervals. Effective preprocessing ensures that the input data is accurate, consistent, and ready for model training, ultimately improving model performance and reliability.

3.1.1 CleanTS

CleanTS is an R package specifically designed for preprocessing time series data. It automates the detection and correction of common data issues such as missing timestamps, duplicates, outliers, and temporal inconsistencies. CleanTS ensures structural and statistical smoothness of time series, which is essential for high-quality forecasting models.

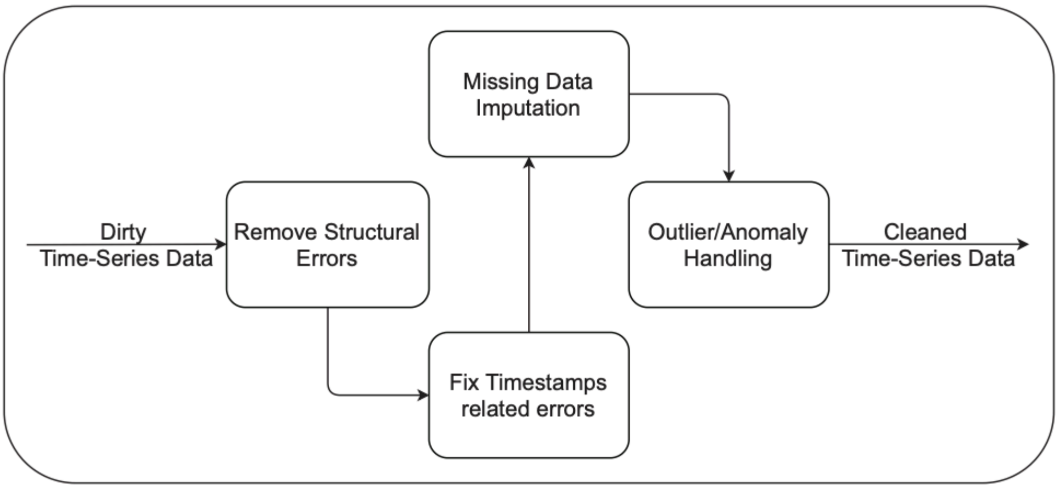


Figure 1 - CleanTs

3.1.2 Aspect of Data Preprocessing

Key aspects of time series data preprocessing include:

- **Outlier Detection and Removal:** It is the process of identifying and eliminating data points that significantly deviate from expected patterns, which can otherwise distort analysis and reduce model accuracy.

- **Missing Timestamps:** It refers to detecting and handling gaps in time-based data by introducing missing time points and imputing values, ensuring continuity and consistency for time series modeling.
- **Timestamp Alignment:** It is the process of ensuring that data is recorded at consistent, regular intervals across the dataset, which is essential for accurate temporal analysis and modeling.
- **Feature Extraction:** It is the process of transforming raw data into informative variables (features) that capture essential patterns or characteristics, making it easier for models to learn and make accurate predictions.

3.2 Variational Mode Decomposition (VMD)

VMD is an adaptive signal decomposition technique that breaks down a time series into a finite number of components known as **Intrinsic Mode Functions (IMFs)**. Each IMF represents a mode with a specific frequency band, enabling better isolation of trends, cycles, and noise in non-stationary signals.

3.2.1 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.

3.2.2 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. The decomposition is achieved by solving the following constrained variational problem:

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

Where;

$f(t)$ = the original signal to be decomposed

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

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

$*$ = convolution operator

$\left(\delta(t) + \frac{j}{\pi t}\right)$ = Hilbert transform kernel to compute analytic signal

$e^{-jw_k t}$ = frequency shifting to baseband

∂_t = time derivative

$||.||_2$ = L2 norm (squared energy of the analytic signal)

This process ensures that each mode is compact in the frequency domain and reconstructs the original signal when summed together.

3.3 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. GPR is known for producing not only point predictions but also confidence intervals around those predictions.

3.3.1 Theory

A Gaussian process defines a distribution over functions, fully specified by a **mean function** and a **covariance function (kernel)**. Given training data, GPR estimates the distribution of functions that are consistent with the observed data. It returns a predictive mean and variance for each test point, making it well-suited for uncertainty-aware modeling.

3.3.2 About Kernels

Kernels (or covariance functions) define the similarity between data points in GPR. They control the smoothness, periodicity, and complexity of the predicted function. Common kernels include:

- **RBF (Radial Basis Function)**: Assumes smooth and continuous functions.
- **Matern Kernel**: Allows for more flexible and rougher functions.
- **Periodic Kernel**: Captures repeating patterns.
- **Linear Kernel**: Models linear relationships.

Choosing or combining kernels is critical to capturing the underlying structure in the data.

3.4 Evaluation Metrics

Evaluation metrics provide a systematic way to assess the accuracy and reliability of forecasting models. These metrics help quantify the difference between predicted and actual values, enabling objective comparisons between models and guiding model improvements. The following metrics were used in this study:

3.4.1 Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is a linear score, meaning all individual differences are weighted equally.

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

Where:

x_i = Actual value

y_i = Predicted value

n = Number of observations

3.4.2 Root Mean Square Error (RMSE)

RMSE is the square root of the average of squared differences between prediction and actual observation. It penalizes larger errors more than MAE.

$$\text{RMSE}=\sqrt{\sum_{i=1}^{n}\frac{(\widehat{y_i}-y_i)^2}{n}}$$

3.4.3 Coefficient of Determination (R^2 Score)

The R^2 score indicates how well the predictions approximate the actual data. A value of 1 indicates perfect prediction, while 0 indicates that the model does no better than the mean.

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

3.4.4 Correlation Heatmap

To evaluate consistency across time, correlation heatmaps were generated using the daily MAE values. Pearson correlation coefficients were computed between daily MAE values of different months (Gregorian or Vedic), forming a symmetric matrix visualized as a heatmap. This helped in identifying seasonal or periodic patterns in forecast accuracy.

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

Where:

$cov(X, Y)$ = Covariance between two variables

$\sigma_x \sigma_y$ = Standard deviations of variables X and Y

CHAPTER 4: PROPOSED WORK

4.1 Flowchart

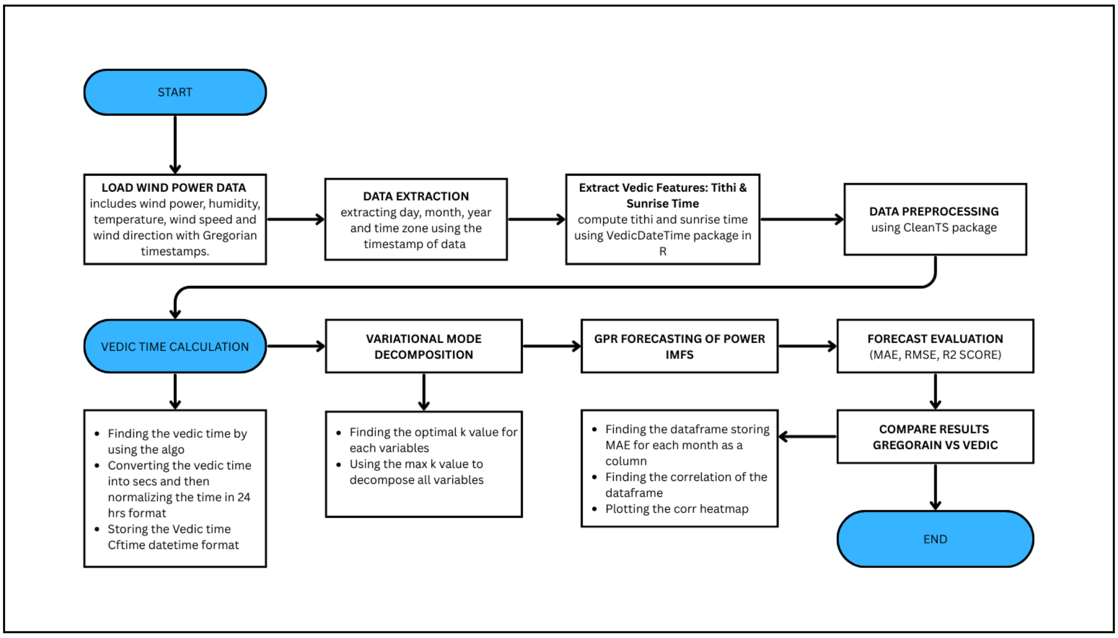


Figure 2 - Flowchart

1. **Start:**

   The forecasting process begins by transforming raw wind power data into meaningful predictions. The process evaluates how using Vedic (lunisolar) time instead of Gregorian time may impact long-term wind power forecasting accuracy.

2. **Load Wind Power Data:**

   The dataset includes wind power, humidity, temperature, wind speed, and wind direction, all timestamped using the Gregorian calendar. The data is loaded for processing in both Python and R environments.

3. **Data Extraction:**

   From the Gregorian timestamp, the day, month, year, and time zone are extracted. These are essential for mapping corresponding Vedic calendar features using astronomical calculations.

4. **Extract Vedic Features: Tithi & Sunrise Time (in R):**

   The VedicDateTime package in R computes tithi (lunar day) and sunrise time using the extracted date and time zone. This facilitates the transition from Gregorian to Vedic time for deeper temporal analysis.

5. **Data Preprocessing (in Python):**
   The raw dataset is cleaned using the CleanTS R package. Outliers are capped at 0, and zero values are imputed with the average of the surrounding five values. This ensures smooth, gap-free data suitable for decomposition and modeling.
6. **Vedic Time Calculation:**
   The tithi CSV exported from R is imported into Python. An algorithm finds the end time of the first tithi in the dataset and derives the start and end times for subsequent tithis. The Vedic time is converted into seconds, normalized to a 24-hour format, and stored using the Cftime format, which supports a 360-day calendar structure.
7. **Variational Mode Decomposition (VMD):**
   VMD is applied to each variable (wind power, temperature, etc.) individually to decompose them into Intrinsic Mode Functions (IMFs). These IMFs help isolate long-term trends, seasonal patterns, and high-frequency components for enhanced model learning.
8. **Determine Optimal k for Decomposition:**
   The optimal number of VMD components (k) is determined through experimentation. The maximum k value is then used to uniformly decompose all variables, ensuring balanced complexity and accuracy in the modeling phase.
9. **GPR Forecasting of Power IMFs:**
   Gaussian Process Regression (GPR) is used to forecast the IMFs of wind power. The exogenous features (temperature, humidity, wind speed, direction) are also input as IMFs to enhance prediction accuracy. Forecast results are stored month-wise.
10. **Forecast Evaluation (MAE, RMSE, R^2 Score):**
    Model performance is quantitatively assessed using evaluation metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score. These metrics reflect the accuracy of the predicted values in both calendar systems.
11. **Compare Results: Gregorian vs Vedic:**
    The final step compares model performance under Gregorian time alignment and Vedic time alignment. This comparison aims to determine whether natural

time segmentation via the Vedic lunisolar system provides any advantage in improving wind power forecast accuracy.

4.2 Data preprocessing

4.2.1 Outlier Handling

For this project, a visual inspection of the time series plots of power, temperature, humidity, wind speed, and wind direction was conducted for outlier identification. From time series graphs, sharp changes which were not consistent with the overall trend were noted for outlier prediction for different features. This visual inspection was essential to confirm that the selected data corrections were appropriate with respect to the true data patterns.

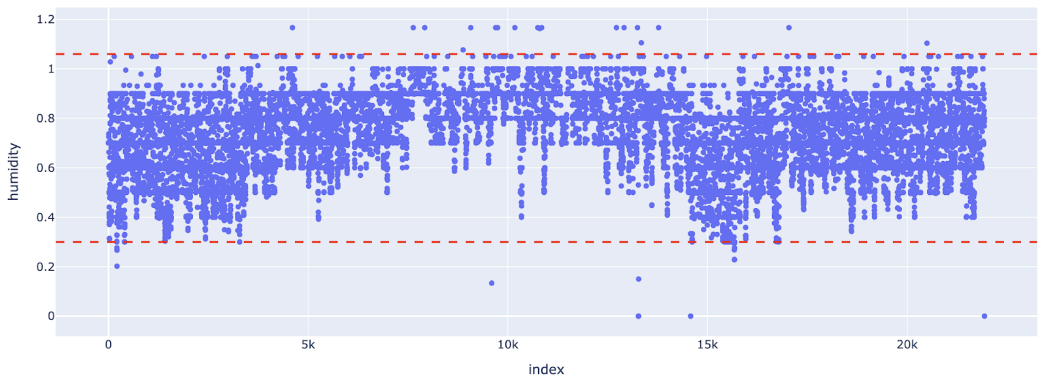


Figure 3 - Outlier Detection

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 off corrected time series data.

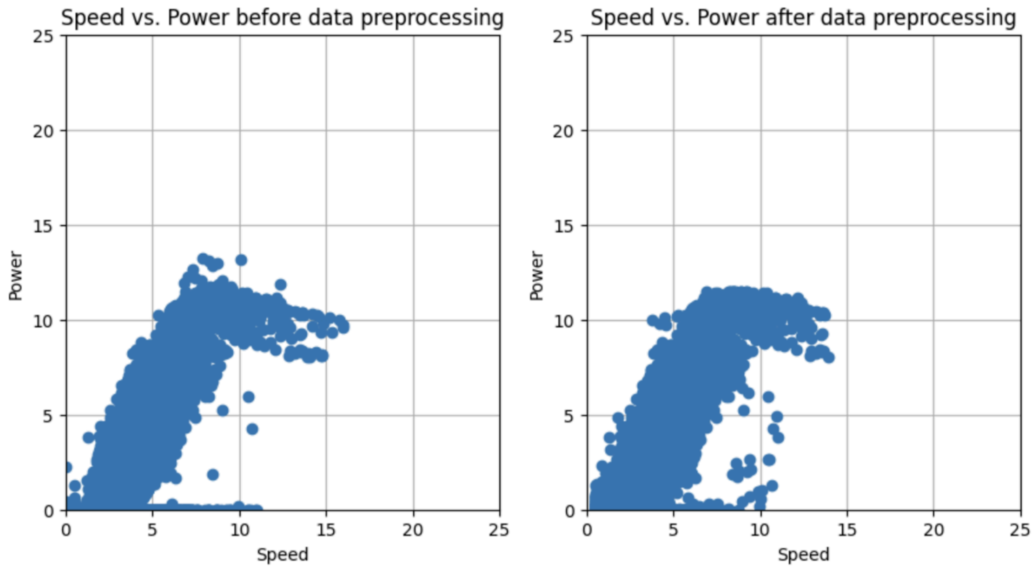


Figure 4 - Data Before vs After Preprocessing

4.2.2 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. 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, the extracted features were vital.

4.3 Vedic Time Computation

This dataset has a datetime entry which is in Gregorian calender form. The dataset also contains the following power variables: power, temperature, wind direction, humidity, and wind speed. In order to align the data into Vedic format, we had to derive new features like Day, Month, Year, and time zone from the datetime column. These attributes were critical for leveraging the VedicDateTime R package to compute two essential Vedic calendar elements: the time of sunrise and the tithi (lunar day) along with the end time of the tithi. The geographical coordinates of the data collection site were also needed as inputs to these calculations.

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. Below is the sample R code to find tithi:

```
# 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. 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. From that point onward, the algorithm calculated the duration of each tithi (ranging typically between 19 to 26 hours), normalized the duration by converting it into seconds, and then mapped it to a 24-hour format. To store these values accurately and in a format consistent with Vedic calendrical structure, the

Cftime data type was used. This type supports a 360-day calendar—perfectly accommodating the 12 months of 30 tithis each—thus preserving the Vedic temporal structure for downstream modeling.

4.3.1 Algorithm

The pseudocode of Algorithm is given below:

Part 1: Initialize

```
None
Input: Gregorian timestamps, tithi_data, sunrise_data, location_coordinates
Output: Vedic timestamps corresponding to each Gregorian timestamp
Initialize empty array vedic_timestamps
```

Part 2: Assign Vedic Timestamps

```
None
For each row in data:
  Find the tithi (lunar day) active at current Gregorian timestamp
  Calculate duration of current tithi in seconds
  Find elapsed time since start of current tithi
  Normalize elapsed time to fit 24-hour format
  Compute Vedic date components (year, month, day, hour, minute, second)
  Store in vedic_timestamps array
```

Part 3: Store Vedic Timestamps in cftime.Datetime360Day format

```
None
Convert vedic_timestamps to cftime.Datetime360Day objects
Return the array of Vedic timestamps
```

4.3.2 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.

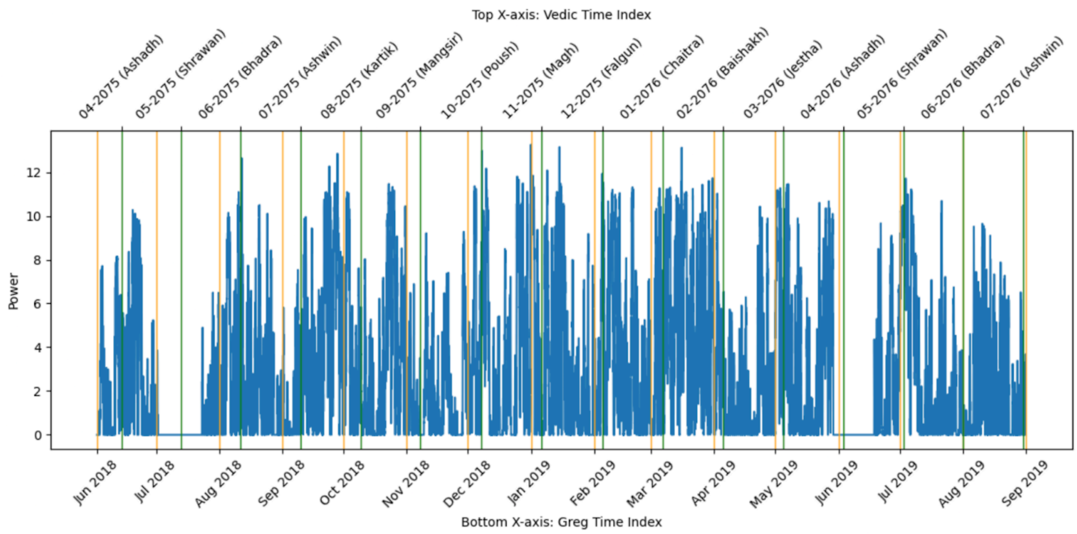


Figure 8 - Time Shift

4.4 Variational Mode Decomposition

Pseudocode of VMD:

```
None
For each variable (power, temperature, wind speed, humidity, wind direction):
  Apply VMD to each variable and find the optimal k val
  find the max_k val out of all optimal val
  Apply VMD to extract max_k IMFs for each variable
  Store IMFs with datetime index
```

In this project, Variational Mode Decomposition (VMD) was used to preprocess the time series data of five key variables: power, temperature, wind speed, humidity, and wind direction. The purpose of applying VMD was to extract distinct Intrinsic Mode Functions (IMFs) from each signal, which represent different frequency components such as trends, cyclic patterns, and noise. This decomposition enhances the forecasting model's ability to capture complex temporal dynamics by analyzing the signal in its individual components.

To ensure the most effective decomposition, an optimal number of modes (k) was determined separately for each variable through experimentation. Once the optimal k values were obtained, the maximum of these values was chosen and uniformly applied across all variables. This ensured a consistent number of IMFs for each variable, which is crucial for creating aligned and structured input data for downstream forecasting models like GPR.

For each variable, the VMD algorithm was run with carefully tuned hyperparameters, including alpha (bandwidth constraint), tau (noise-tolerance fidelity), and convergence tol. The decomposition yielded k IMFs per signal, which were stored in a DataFrame format and indexed using the datetime index of the original data. This allowed for proper time-based alignment across all decomposed variables and ensured that each IMF could be directly used as an input feature for predicting the corresponding power IMF.

Visualizations were also generated to compare the original signal and the reconstructed signal (sum of IMFs), along with individual plots for each IMF. This helped verify the accuracy of the decomposition and provided insight into how the signal was broken down into interpretable components. Overall, VMD served as a powerful preprocessing step to transform the raw meteorological and power signals into frequency-resolved features, improving the learning capability of the subsequent forecasting models.

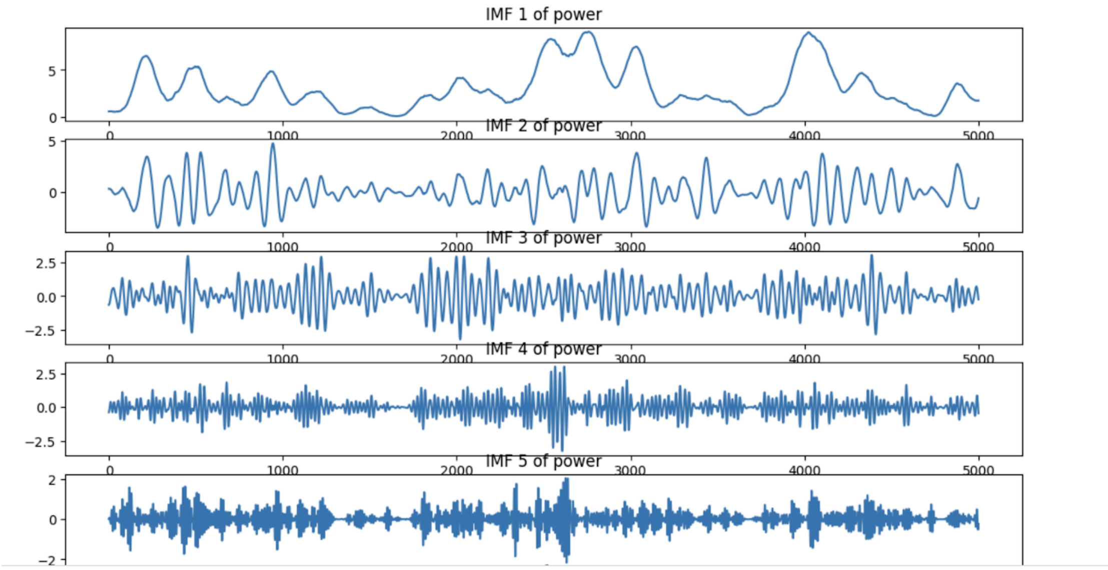


Figure 9 - VMD

4.5 Gaussian Process Regression (GPR)

In this project, Gaussian Process Regression (GPR) is employed to forecast wind power by modeling each decomposed component (Intrinsic Mode Function or IMF) of the power signal separately. The power signal, along with the exogenous variables—wind speed, temperature, humidity, and wind direction—are first decomposed into multiple IMFs using a signal decomposition method like VMD (Variational Mode Decomposition). Each IMF captures a distinct frequency component of the original signal, allowing the model to learn complex patterns at multiple time scales.

Pseudocode of GPR:

```None
For each power IMF:
   Create feature matrix X from exogenous IMFs
   Set target Y as the power IMF values
   Initialize GPR with RBF kernel and inducing points
   Optimize kernel parameters in two stages
   Store trained model
```

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. 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. The forecasts for all power IMFs are then summed to reconstruct the final forecasted power value. This predicted power is compared against the actual power value using evaluation metrics like Root Mean Squared Error (RMSE). The process is repeated for multiple days, and all forecasts are stored in a DataFrame for analysis.

In summary, GPR is applied in a component-wise manner, using the IMF framework to model power dynamics at different frequencies, while leveraging meteorological variables to provide accurate, data-driven forecasts of wind power.

4.6 Forecasting Approach

In this project, a one-day-ahead forecasting framework was implemented using the trained Gaussian Process Regression (GPR) models. The objective was to evaluate forecasting performance under two distinct time-referencing systems: the Gregorian calendar and the Vedic lunisolar calendar. This dual approach aimed to explore how cultural and astronomical time alignment impacts renewable energy forecasting accuracy.

1. Gregorian-Based Forecasting:
    - Forecasts were made at 10:00 AM Gregorian time each day.
    - The previous 60 days data at 10:00 AM was used for training.
    - Forecasting was repeated across multiple months.
    - Forecast accuracy was measured using Mean Absolute Error (MAE), and results were grouped by Gregorian months for correlation analysis.

2. Vedic-Based Forecasting:
    - Forecasts were made at 5 hours post-sunrise each day, based on Vedic time.
    - The previous 60 days of aligned post-sunrise data were used for training.
    - Forecasting was done for the next day at the same Vedic-relative time.
    - MAE was computed and grouped by Vedic months to analyze seasonal behavior in the traditional calendar system.

```
None
For each day to forecast:
   Extract previous 60 days of aligned data
   For each power IMF:
      Predict using corresponding GPR model
   Sum predicted IMFs to obtain final power forecast
   Calculate error metrics
   Store results
```

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.

CHAPTER 5: RESULT

5.1 Model Training

To evaluate the performance of our hybrid VMD-GPR model, we implemented a rolling one-day-ahead forecasting strategy. The key idea was to forecast the wind power value for a given day by training the model using data from the previous 60 days. This process was repeated in a loop across 30 consecutive days to simulate continuous daily forecasting.

For each forecast iteration:

- We extracted 60 days of historical data aligned to a specific time reference (Gregorian or Vedic)
- The input features included wind direction, temperature, humidity, and wind speed. The corresponding output variable was the power value for each decomposed IMF.
- Each Intrinsic Mode Function (IMF) was modeled individually using a Sparse Gaussian Process Regression (GPR) model with a composite kernel:
  RBF kernel + Bias kernel.

The training was done using the train_sparse_gp function, which:

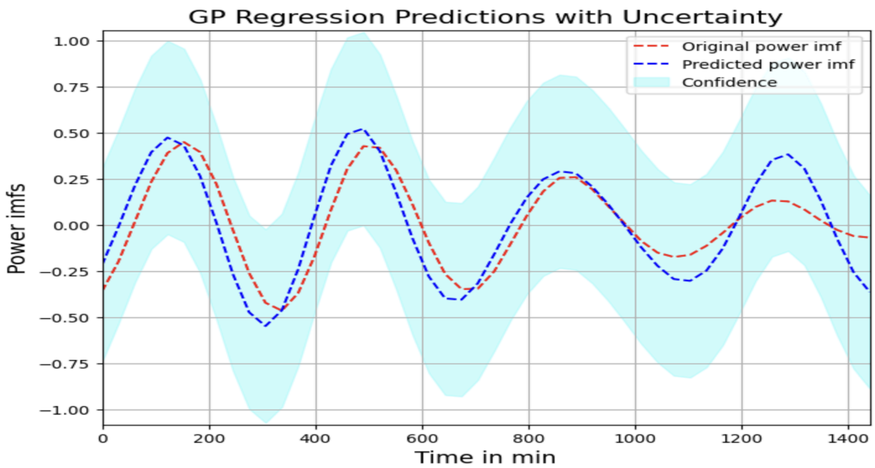
1. Initialized a SparseGP model with 100 inducing points
2. Fixed the noise variance for initial optimization (200 iterations) to stabilize training.
3. Unfixed the variance and performed full model optimization (up to 5000 iterations) to fit the GP.

After training:

- The GPR model predicted the IMF value for the next day.
- All IMF forecasts were summed to reconstruct the final wind power prediction.
- Forecast accuracy was evaluated using multiple error metrics:
    1. Root Mean Squared Error (RMSE)

2. Mean Absolute Error (MAE)

This entire process was repeated for each day in the test window by incrementing the date, simulating a real-world rolling forecast system. Forecast results and errors for each day were stored in a DataFrame and saved to a CSV file for further analysis.



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

5.2 Gregorian Calendar-Based Forecasting Results (10 AM Forecast)

In this approach, wind power forecasting was performed for 10:00 AM daily using data aligned to the Gregorian calendar. The hybrid VMD-GPR model was trained and evaluated using a sliding window approach with the following configuration:

- Training window: Previous 60 days of data
- Forecast target: Wind power at 10 AM of the next day

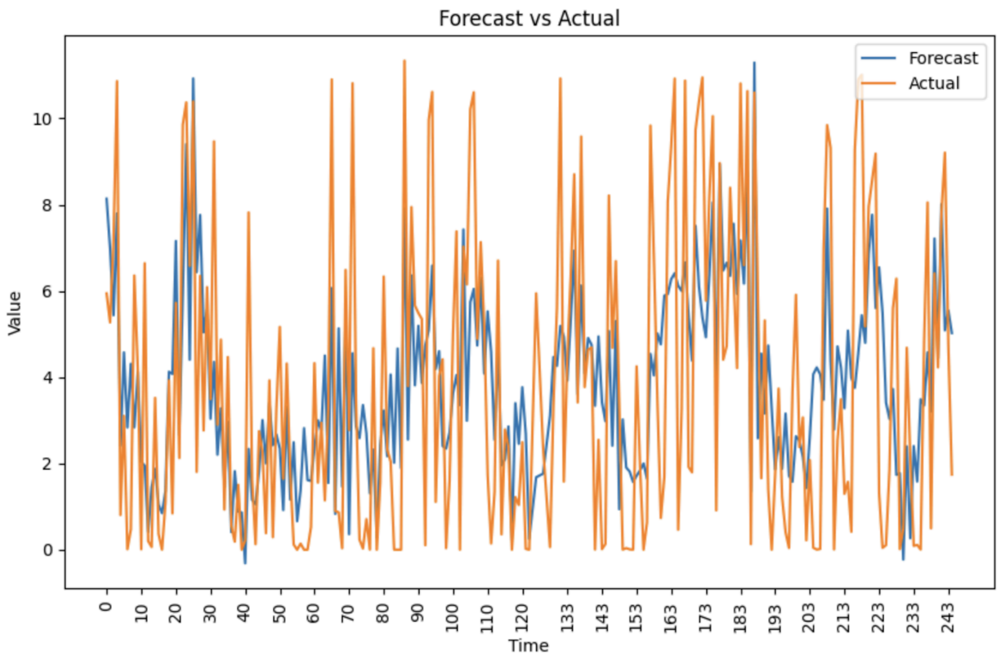


Figure 11 - Gregorian Forecast vs Actual Plot

5.2.1 Forecast Storage and Daily Evaluation

For each prediction:

- Forecasted power values were stored alongside actual observed values in a structured DataFrame
- The Mean Absolute Error (MAE) was computed at each forecasted timestamp
- Daily MAE values were grouped across the forecast period to form a (31 × N months) matrix for monthly analysis

This enabled observation of forecast accuracy trends over days and comparative performance across months.

5.2.2 Correlation Heatmap Analysis

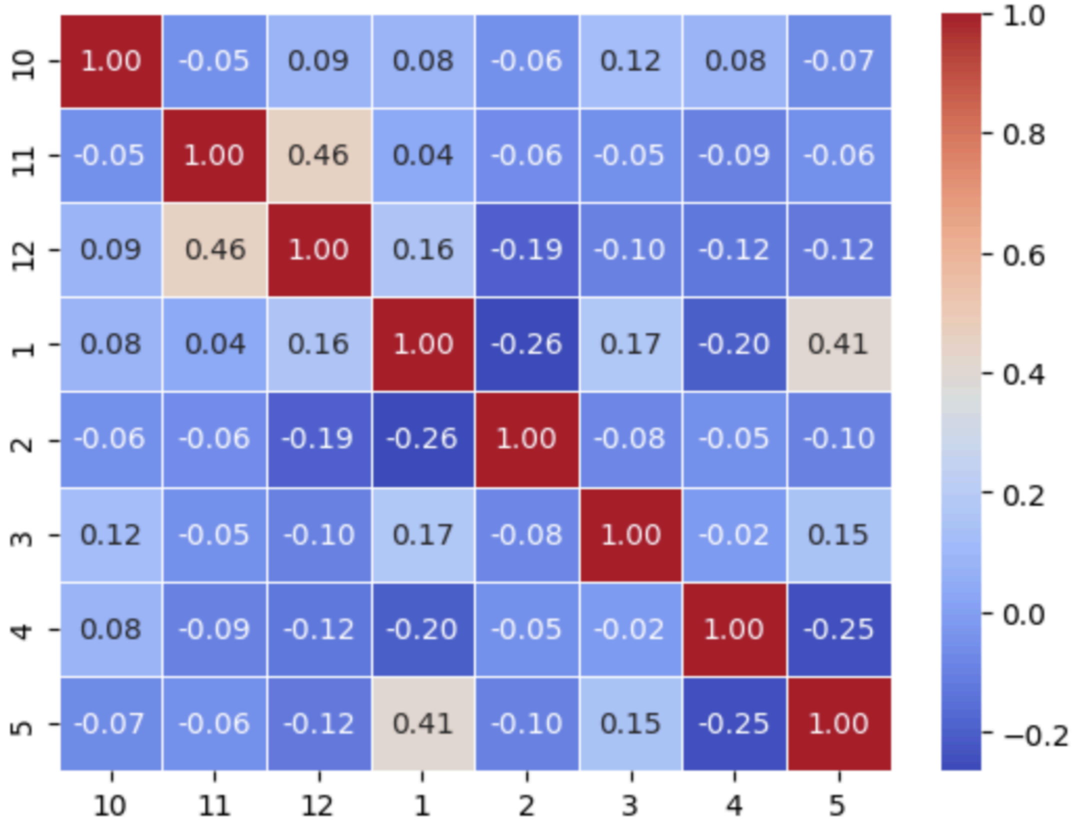


Figure 12 - Gregorian Corr Heatmap

A correlation matrix of daily MAE values across Gregorian months was computed and visualized as a heatmap. Here:

- Each axis denotes months (e.g., 10 = October, 11 = November, etc.)
- Cells represent the Pearson correlation coefficient between two months' MAE profiles
- Higher correlation (deep red) indicates similar day-wise error behavior

5.2.3 Key Observations:

- A relatively high correlation (e.g., 0.46) is observed between months like November–December and December–January, showing localized temporal similarity
- However, the majority of months exhibit low or even negative correlation, suggesting inconsistent forecasting behavior across time

- For example, February, March, and April show weak or poor correlation with other months, highlighting instability in performance

Metrics	Result
RMSE	2.6123
MAE	2.1424
R2 Score	0.4414

Table 1 - Gregorian Results

5.2.4 Interpretation

While the Gregorian approach provides moderately accurate forecasts, the inter-month inconsistency indicated by the correlation matrix suggests that the model struggles to generalize across different times of the year. This limits its reliability for long-term or seasonally adaptive forecasting.

5.3 Vedic Calendar Based Forecasting Results (Sunrise + 5 Hours Forecasting)

This approach utilizes the Vedic calendar system by targeting wind power values 5 hours after sunrise each day — a dynamic time that respects traditional tithi-based timekeeping. This aligns forecasts more closely with lunar-solar rhythms, potentially offering more natural temporal segmentation.

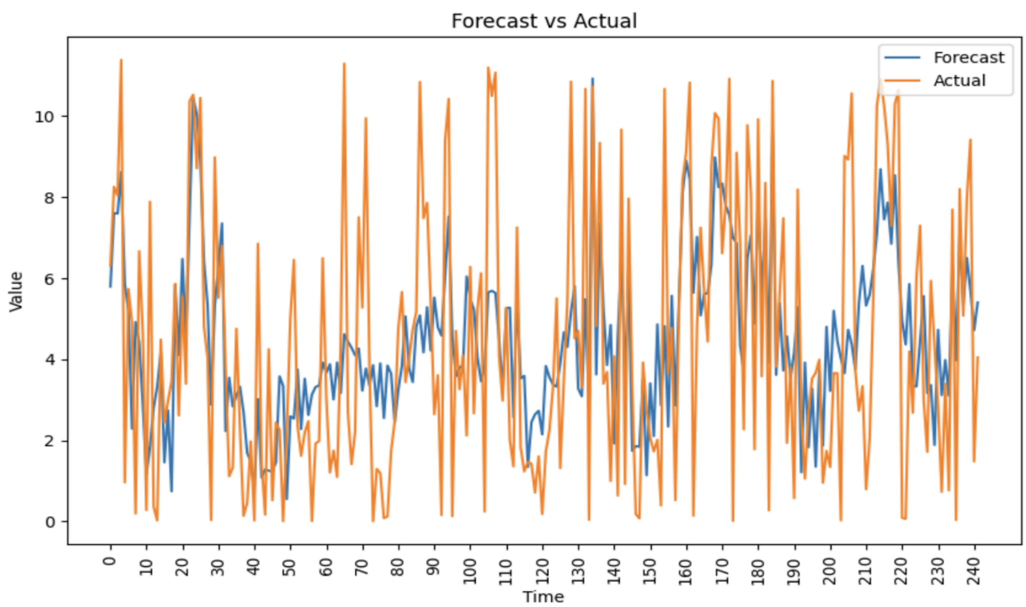


Figure 13 - Vedic Forecast vs Actual Plot

5.3.1 Data Processing and Evaluation

- Forecast target: Wind power 5 hours post-sunrise, computed via the VedicDateTime package
- Error aggregation: MAE computed for each tithi, grouped by Vedic month
- Resulting matrix: (30 rows × 12 columns), representing MAE per tithi per month

This allows study of forecast performance consistency across the Vedic calendar.

5.3.2 Correlation Heatmap of Vedic Months

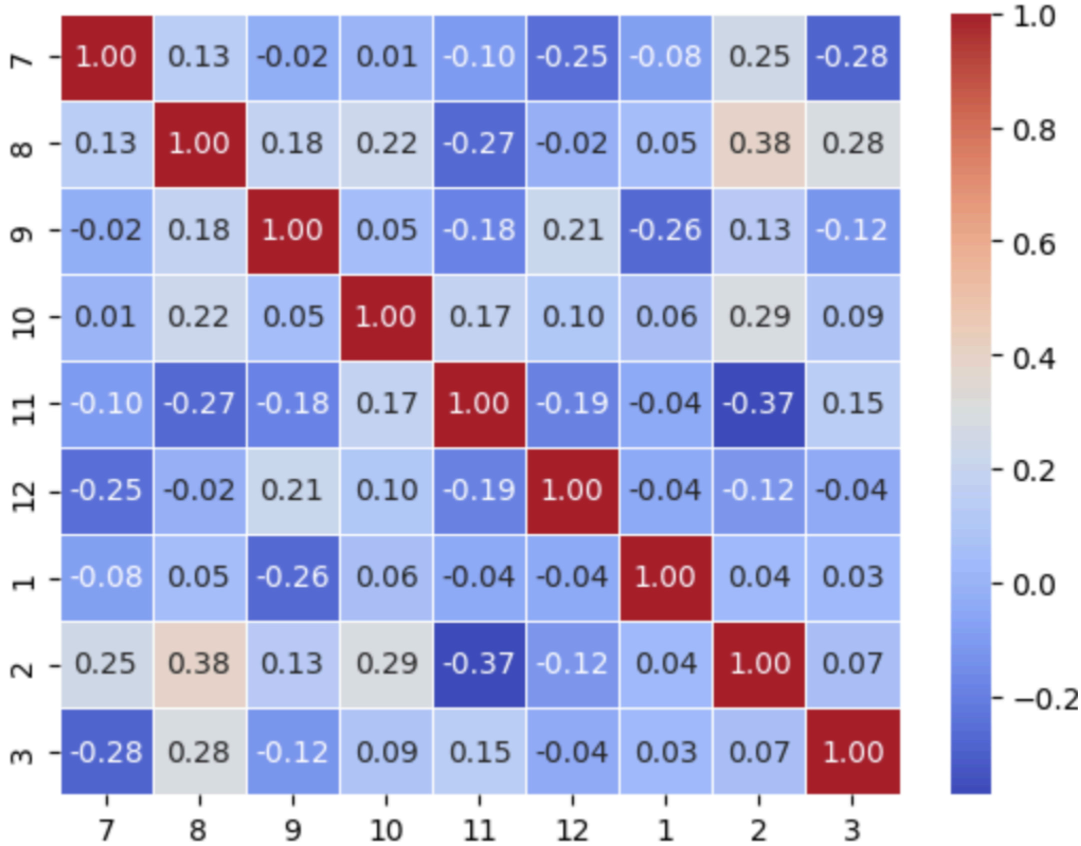


Figure 14 - Vedic Corr Heatmap

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

5.3.3 Key Observations:

- Unlike the Gregorian case, the Vedic heatmap shows moderate correlation across most months, reflecting more consistent model behavior
- For example, Vaishakha (2) shows positive correlation with Shravana (8) at 0.38, and similar behavior is observed in other month-pairs
- Importantly, the heatmap lacks extreme cold colors, indicating absence of sharply negative correlations, a contrast to the Gregorian counterpart

Metrics	Result
RMSE	2.5519
MAE	2.0763
R2 Score	0.4224

Table 2 - Vedic Results

5.3.4 Interpretation

The Vedic approach results in slightly better performance metrics (lower MAE and RMSE) than the Gregorian one, and more importantly, shows better stability and consistency in forecasting across months. This is clearly reflected in the correlation heatmap, where the error behavior remains reasonably aligned across the lunar calendar. The absence of large correlation drops reinforces its robustness.

5.4 Discussion on Result

- The Vedic-based method achieves lower MAE and RMSE, reflecting superior forecast precision
- The correlation matrix of the Vedic approach shows greater month-wise consistency, while the Gregorian approach exhibits disruptive variations with strong correlation only in isolated months
- These results suggest that Vedic calendar segmentation aligns better with the underlying wind patterns, making it more effective for long-term wind power forecasting

CHAPTER 6: SIMULATION ENVIRONMENT

6.1 Hardware Requirements

1) CPU Cores:

- A multi-core processor is essential for parallel processing and faster execution of decomposition and forecasting tasks.
- Recommended: A processor with at least 4 to 8 cores (e.g., Intel Core i7, AMD Ryzen 7); 16 cores or higher is ideal for optimal performance.

2) RAM:

- Minimum: 8 GB
- Recommended: 16 GB or more (to efficiently handle large datasets and multiple decomposed components during VMD and GPR processing)

3) Storage:

- SSD with at least 256 GB capacity for faster read/write operations, quick access to large datasets, and smooth execution of Python and R scripts.

4) Internet Connection:

- Required initially for downloading necessary libraries and dependencies.
- Also required for running code and notebooks on cloud platforms like Kaggle.

6.2 Software Requirements

1) Operating System:

- 64-bit Windows 10/11, macOS, or Linux (Ubuntu 20.04 or later) with full compatibility for Python and R-based libraries and tools.

2) Programming Languages:

- Python (version 3.8 or higher) – Used for data preprocessing, VMD decomposition, GPR modeling, forecasting, and visualization.
- R (version 4.0 or higher) – Used primarily for timestamp conversion from Gregorian to Vedic calendar using the vedicdatetime package.

3) Python Libraries:

- pandas, numpy, scipy – For data manipulation, numerical operations, and imputation.
- matplotlib, seaborn, plotly – For visualizing time series trends, forecast accuracy, and calendar alignment shifts.
- vmdpy – For performing Variational Mode Decomposition (VMD) to extract intrinsic mode functions (IMFs).
- GPy – For implementing Gaussian Process Regression (GPR) for wind power forecasting.

4) R Packages:

- CleanTS – For automated data preprocessing such as correcting timestamps, handling missing values, and detecting/removing outliers.
- vedicdatetime – For converting Gregorian calendar timestamps to Vedic lunisolar calendar format to explore time-aligned forecasting.

5) Development Environment / IDEs:

- Kaggle – Used as the primary development and execution platform for Python-based components, including model training and evaluation.
- RStudio – Used for R scripting related to Vedic time computation and preliminary data preparation.

6) Spreadsheet Tools (Optional):

- Microsoft Excel or Google Sheets – For analyzing and presenting final forecast reports, including error metrics (MAE, RMSE, R^2) and visual summaries of actual vs predicted values.

6.3 Screenshot of Simulation

Figure 15: Screenshot of Kaggle 1

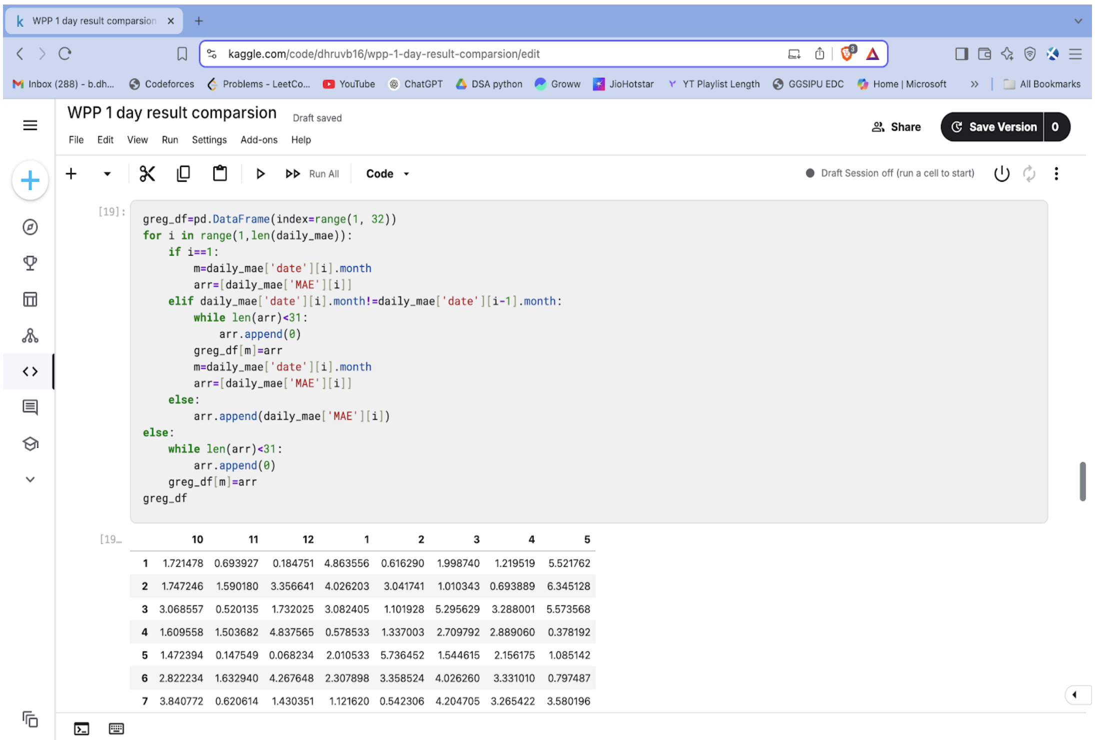


Figure 16: Screenshot of Kaggle 2

CHAPTER 7: CONCLUSION

This research illustrated the exceptional capabilities of a hybrid model incorporating Variational Mode Decomposition (VMD) and Gaussian Process Regression (GPR) in long-term wind power forecasting. The model's ability to capture intricate nonlinear dynamics along with temporal patterns in wind power data was achieved through signal decomposition combined with probabilistic regression, and its meteo-climatic features were captured with fair accuracy.

The study implemented two forecasting methods:

1) Gregorian-Based Forecasting: A wind power forecasting method that attempts to predict wind power precisely at 10 AM based on the Gregorian calendar.

2) Vedic-Based Forecasting: A forecasting method based on traditional Vedic timekeeping, designed to provide predictions for a 5-hour period after sunrise, structured around Vedic days (tithis) and months.

For both methodologies, VMD was engaged to extract features pertinent to the problem from the wind power signal and its external predictors into Intrinsic Mode Functions (IMFs) which encapsulated its trends, seasonality, and miscellaneous noise components. For each component, individual GPR models were developed, and their forecasts were aggregated to form the ultimate wind power forecast.

7.1 Key Findings:

- The Gregorian-based model achieved a MAE of 2.1424, RMSE of 2.6123, and R^2 score of 0.4414.
- The Vedic-based model outperformed slightly with a MAE of 2.0763, RMSE of 2.5519, and R^2 score of 0.4224.
- Importantly, the correlation heatmaps for the MAE values across months revealed that the Vedic-based forecasting showed stronger temporal consistency than the Gregorian-based method.
- These results suggest that the Vedic-based approach is not only comparable but potentially more effective in capturing consistent forecast patterns.

7.2 Future Work:

The promising results from this study open several avenues for future exploration:

- Since the Vedic forecasting approach gives better forecasting accuracy than the greograin forecasting appproach, as evident from the correlation heatmap and evaluation metrics. This suggests the potential of using the vedic timestmap in improving forecasting accuracy.
- The Vedic forecasting approach can be extended to other forecasting domains such as solar power prediction, demand energy forecasting , and rainfall prediction, where time alignment with natural phenomena may enhance model performance as the vedic calendar is lunisolar.
- More advanced deep learning models like Bi-directional Long Short-Term Memory (Bi-LSTM) can be employed in place of GPR to capture long-term temporal dependencies and further improve prediction accuracy.
- A dedicated R package can be developed to convert Gregorian timestamps into Vedic timestamps, enabling researchers and analysts to incorporate the Vedic calendar into their workflows.
- A standardized Vedic time format can be proposed, allowing structured data recording and analysis aligned with Vedic principles. This would facilitate a new dimension of time-based data science rooted in traditional systems.

REFERENCES

[1] Darmawan, Gumgum, et al. "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 2023.1 (2023): 1863346

[2] Shende, Mayur Kishor, Andres E. Feijoo-Lorenzo, and Neeraj Dhanraj Bokde. "cleanTS: Automated (AutoML) tool to clean univariate time series at microscales." Neurocomputing 500 (2022): 155-176

[2] Bokde, Neeraj Dhanraj, et al. "VedicDateTime: An R package to implement Vedic calendar system." Multimedia Tools and Applications 83.11 (2024): 32141-32157

[3] Karaku¸s, O.; Kuruo ˘glu, E.E.; Altınkaya, M.A. One-day ahead wind speed/power prediction based on polynomial autoregressive model. IET Renew. Power Gener. 2017, 11, 1430–1439. [CrossRef]

[4] Lee, D.; Baldick, R. Short-term wind power ensemble prediction based on Gaussian processes and neural networks. IEEE Trans. Smart Grid 2013, 5, 501–510. [CrossRef]

[5] Ouyang, T.; Zha, X.; Qin, L.; Kusiak, A. Optimisation of time window size for wind power ramps prediction. IET Renew. Power Gener. 2017, 11, 1270–1277. [CrossRef]

[6] Tasnim, S.; Rahman, A.; Oo, A.M.T.; Haque, E. Autoencoder for wind power prediction. Renew. Wind. Water Sol. 2017, 4, 6. [CrossRef]

[7] Dong, L.; Wang, L.; Khahro, S.F.; Gao, S.; Liao, X. Wind power day-ahead prediction with cluster analysis of NWP. Renew. Sustain. Energy Rev. 2016, 60, 1206–1212. [CrossRef]

[8] Abedinia, O.; Lotfi, M.; Bagheri, M.; Sobhani, B.; Shafie-Khah, M.; Catalão, J.P. Improved EMD-based complex prediction model for wind power forecasting. IEEE Trans. Sustain. Energy 2020, 11, 2790–2802. [CrossRef]

[9] Bokde, Neeraj Dhanraj. "Natural time-series analysis and vedic hindu calendar system." arXiv preprint arXiv:2111.03441 (2021).

[10] Dragomiretskiy, Konstantin, and Dominique Zosso. "Variational mode decomposition." *IEEE transactions on signal processing* 62.3 (2013): 531-544.

[11] Jin, H.; Shi, L.; Chen, X.; Qian, B.; Yang, B.; Jin, H. Probabilistic wind power forecasting using selective ensemble of finite mixture Gaussian process regression models. Renew. Energy 2021, 174, 1–18. [CrossRef]