



## Univariate Time Series Prediction of Wind speed with a case study of Yanbu, Saudi Arabia

Yunus Parvej Faniband<sup>1</sup>, S. M. Shaahid<sup>2</sup>

<sup>1</sup> King Fahd University of Petroleum and Minerals, Center for Communication & IT Research, Research Institute, Dhahran, Saudi Arabia, yparvej@kfupm.edu.sa

<sup>2</sup> King Fahd University of Petroleum and Minerals, Center for Engineering Research, Research Institute, Dhahran, Saudi Arabia, mshaahid@kfupm.edu.sa

### ABSTRACT

Wind energy is a promising alternative for renewable source of energy pursued world-wide to reduce carbon emissions for a green future. The prediction of wind speed is a challenging subject and plays an instrumental role in development of wind power systems (particularly grid connected renewable energy systems where predicting wind speed facilitates manipulation of the load on the grid). Modern machine learning techniques including neural networks have been widely utilized for this purpose. Literature indicates availability of several models for estimation of the wind speed one hour ahead and the hourly wind speed data profile one day ahead. This paper considers the prediction of wind energy as a univariate time series (UVT) prediction problem and employs major prediction algorithms including the K-Nearest Neighbors (kNN), Random Forest (RF), Support Vector Regression (SVR), Holt-Winter and ARIMA method. Forecasting a univariate time series depends only on past wind speed data values, rather than use of external data attributes like wind direction or weather forecast for prediction algorithm. In the present study (as a case-study), 13 years of hourly average wind speed data (of the period 1970-1982) of Yanbu, Saudi Arabia has been utilized to evaluate the performance of selected algorithms. Yanbu is an industrial city that plays a major role in the economy of Saudi Arabia. The findings showed that SVR, RF and ARIMA methods exhibit a better forecasting performance in relation to four evaluation parameters of Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (sMAPE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE).

**Key words:** Artificial Intelligence, Forecasting, Modeling, Neural networks, Wind speed, Wind power systems.

### 1. INTRODUCTION

Wind power, which is non-depletive, non-polluting, environmentally sustainable, economically competitive and socially beneficial, seems to have become the inevitable source of electricity generation in the world from renewable energy sources. Nevertheless, because of the nonlinear properties of wind speed across different times and across

different sites, precise information about dynamics of wind in the wind farm is very critical to handle various operations of wind energy conversion systems (WECS) for preload sharing, power system management, energy trading, and maintenance scheduling of the wind turbines. A wind power forecast or prediction corresponds to an estimate of the expected power generation from one or more wind turbines (clustered into a wind farm). Based on power system operation requirements, long-term (1–7 days) wind speed predictions[1][2] are essential for maintenance scheduling of the wind turbines, while short-term and medium-term wind speed forecasts are essential to increase the efficiency of wind energy generation systems[3] and to facilitate the integration of wind energy into utility grid[4][5].

The wind power forecasting is divided into four types depending on the information available and the time requirements of power system operations[6][7]. The time scales for *very short-term* forecasts vary from few seconds to 30 min, whereas *short-term* forecasts used for preload sharing, system management and maintenance schedules, requires the prediction time scale of 30 min to 6 hours. *Medium-term* (6–24 h) wind speed forecasts address the issue of power system management, energy trading and scheduling, whereas the *long-term* (1–7 days) forecasts are essential for maintenance scheduling and resource planning of the wind turbines [8][9].

Forecasting future values of observed time series is a crucial requirement in all science and engineering fields including finance, economics, industrial applications and business intelligence. Furthermore, with the rapid growth of big data, time-series forecasting algorithms need to analyze ever-increasing massive data sets, covering different domains including wind power generation.

The primary aim of this article is to employ four major machine learning algorithms in order to predict the short-term wind speed, using a univariate time series (UVT) method. Forecasting a univariate time series depends only on past wind speed data values, rather than use of external data attributes like wind direction or weather forecast for prediction algorithm. This study conducts the evaluation of the algorithms considering the accuracy, training time, and

forecasting time. This method is useful in the wind energy systems specifically for preload sharing, system management and maintenance schedules.

This paper employs four major prediction algorithms including the K-Nearest Neighbors (k-NN), Random Forest (RF), Support Vector Regression (SVR), Holt-Winter Method to address the problem of short-term wind speed forecasting using univariate time series. One step ahead forecasting is estimated by determining optimal parameters for each algorithm. Four evaluation metrics, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (sMAPE) and Mean absolute percentage error (MAPE) are used to calculate the performance accuracy of the wind speed forecasting. The prediction time and time taken for training in each approach are also discussed.

Subsequently, this paper is organized as follows: Section 3 discusses the wind speed forecasting problem. Section 4 summarizes the time series forecasting algorithms used in this paper, which are derived from the machine learning approaches. Details of data set used are provided in the Section 5 followed by discussion of results in Section 6 and finally Conclusion with recommendations (Section 7).

## 2. LITERATURE REVIEW

There has been extensive research on developing efficient methods of wind forecasting. According to the literature, there are two mainstream methods that are used for wind speed forecasting [10]. The first method called the Physical approach utilizes the manufacturer's power curve to propose an estimation of the wind power output, with details of the wind flow around and inside the wind farm site [11]. The second approach is the statistical or data-driven prevision method, which depends on creating statistical models including machine learning technique, built by recognizing the complex relationships between weather predictions, previous historical measures and power generation [11].

Starting with traditional statistical approaches followed by a number of physical approaches, there has been a recent trend of using artificial intelligence with machine learning techniques and a number of other new methods [6][12]. Some studies in literature showed that it is possible to provide satisfactory precise forecasting by combining several physical aspects by adopting Numerical Weather Forecast (NWF) and MesoScale models [13], which are considered as physical methods. Few studies [14][15][16] employed traditional statistical approaches like Auto- Regressive Integrated Moving Average (ARIMA) and its derived methods. More recently new methods are adopted using Machine Learning algorithms to tackle time-series forecasting problems. Recent Studies mainly proposed Artificial Neural Network (ANN), compared to conventional model-based approaches, as ANNs are self-adaptive reliable predictive models [17]. ANN based

studies that are significantly investigated in the literature more recently used the following techniques : Multi-Layer Perceptron (MLPs) [18][17], K-nearest neighbor regression [19][20], Random Forest [21], Support vector regression [22][23], and Gaussian processes, Radial Basis Function [24][25][17] and the Recurrent Neural Networks [26][27][28] and the Fuzzy Logic [29][30][31].

According to the number of characteristics attributes used for forecasting, wind prediction methods can be divided into two major categories, namely, univariate model-based methods and multivariate model-based methods. Univariate methods take into account the value of only either past or current wind speed data in time series, rather than use of other external data attributes. The multivariate methods generally take into account other weather parameters that influence wind speed value such as wind direction, air temperature, atmospheric pressure, relative humidity and surface roughness. Univariate approaches seem to be cost-effective because it does not require the procurement and maintenance of a weather station or other measurement instruments. Furthermore, univariate methods are more effective in the case of short-term forecasts used for preload sharing, system management and maintenance schedules, as they do not depend on an extended process of data collection. However Multivariate time series model-based methods aims to yield more accurate forecasting compared to Univariate forecasting method.

Many researchers have investigated the multivariate wind speed forecasting methods. Authors of the article [32] used three parameters (temperature, wind speed and direction) to compute short-term wind power forecasting on a wind farm. Researchers of another study [33] proposed that wind speed and wind direction as inputs gives better prediction with multivariate model consisting of 4 characteristics (temperature, wind speed, wind direction and humidity) for a day ahead prediction of wind speed. Authors of the article [34] used six fundamental variables in wind power output for a day ahead forecasting of wind power generation.

Some studies focused on forecasting using univariate time series data using either linear or nonlinear regressive models [35]. These research efforts addressed the prediction time scale of the next 100 minutes of the day using the input of wind speeds of the previous 5 days, while some of the studies used univariate time series forecasting method to compare their approach with multivariate model [36][37].

## 3. PROBLEM FORMULATION

The problem of wind speed forecasting is aimed at seeking an approximation  $f(t+k)$  of the wind vector  $y(t+k)$  based on previous  $n$  measurements  $y(t)$ ,  $y(t+1)$ , ...,  $y(t-m+1)$ . 'k' is chosen to be small in order to provide precise wind speed forecast and this is known as short-term wind speed forecasting [4].

We consider this problem as a classical univariate forecasting problem with discrete time points. The goal is to use  $y = (y(t_1) \dots y(t_T))$  observations of a single time series observed up to time point ' $t_T$ ' in order to find a forecaster that can allow reliable time forward predictions  $\hat{y}(h_j) = \hat{f}(h_j)$  for the time points  $h_1 \dots h_H$  of the forecasting horizon.

The forecast or prediction error ( $E_t$ ) is defined as the difference between the actual value and the predicted value for the corresponding time ( $t$ ) and given by below equation:

$$E_t = \text{Actual value } (Y_t) - \text{Forecasted Value } (F_t) \quad (1)$$

Error measures have a significant role to play in calibrating the refining process of forecasting models. Since the principal goal of forecasting is to improve the prediction accuracy, error measures or evaluation metrics are very significant from the point of view of the forecaster. There are a variety of different error measures definitions of forecasting in the literature, few of them are listed below:

- 1) Mean Squared Error (MSE):

$$MSE = \frac{\sum_{t=1}^N E_t^2}{N} \quad (2)$$

- 2) Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{\sum_{t=1}^N \left| \frac{E_t}{Y_t} \right|}{N} \quad (3)$$

- 3) Mean Absolute Error(MAE):

$$MAE = \frac{\sum_{t=1}^N |E_t|}{N} \quad (4)$$

- 4) Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{t=1}^N E_t^2}{N}} \quad (5)$$

- 5) Symmetric Mean Absolute Percentage Error (sMAPE):

$$sMAPE = \text{mean}\left(\frac{200 |Y_t - F_t|}{(Y_t + F_t)}\right)$$

- 6) Mean Absolute Scaled Error (MASE):

MASE =  $\text{mean}(|q_t|)$ , here  $q_t$  is scaled error

$$q_t = \frac{E_t}{\frac{1}{n-1} (\sum_{i=2}^N |Y_i - Y_{i-1}|)} \quad (6)$$

This paper uses Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), MASE (mean absolute scaled error) and sMAPE (symmetric mean absolute percentage error), as error measures for the exercises that are discussed in the Section 6. MASE and sMAPE are scale-independent metrics and are thus ideal for comparing forecasting algorithms across various series[38].

## 4. TIME SERIES FORECASTING MODELS

Forecasting models are the mathematical models which capture certain characteristics or features and include certain assumptions of time series, in order to predict the next unknown value. The forecasting algorithms used in this paper are derived from the machine learning approaches and this section summarizes the algorithms used. These algorithms have also been used in other prediction applications including but not limited to domains of Weather forecasts, energy load forecasting, domain of coastal and marine applications.

### 4.1 Naïve or Persistent

Naïve method is one of the extremely simple and surprisingly effective forecasting methods, where the current forecast value is set to be the value of the last observation. In other words the prediction for the next minute's wind speed equals to the current wind speed, i.e.,  $\hat{y}(t + 1) = y(t)$ . This method is considered as a base line for algorithm comparisons.

### 4.2 Auto Regressive Integrated Moving Average

Autoregressive Integrated Moving Average algorithm (ARIMA) is a statistically sophisticated model and is one of the traditional algorithms for time series forecasting. Autoregressive models reliably predict short-term temporal structures, but accuracy drops for long-term high-level structures as these models assume certain stationary time series properties. ARIMA is represented with a standard notation ARIMA(p,d,q), where p, d and q are the order of the three main components AR, I, and MA, respectively. Within the ARIMA method, also known as the Box-Jenkins method[39], the Auto Regressive (AR) component uses the dependent relationship between an observation and some lagged observations. Part I model the number of differences required to turn the time series into a stationary time series. Moving Average (MA) component models the contribution of noise terms arising from the reliance on lagged measurement between a measurement and residual errors from a moving average model.

ARIMA method requires to tune the parameters (p,d and q) in order to minimize the MAE for selected model and can produce high prediction accuracy.

### 4.3 Holt-Winter Method

The Holt-Winter[40] approach is a well-known forecasting techniques for the time series and takes into account both seasonality and trend components. Additive and Multiplicative are two variants of this algorithm based on the seasonality. If the degree of seasonal variability increases as the mean time series level increases, then the seasonality is called multiplicative. If the seasonal effect is independent of the current mean level of the time series, it is classified as Additive.

Considering an exponential smoothing of level ( $S_t$ ), seasonal index ( $I_t$ ) and trend ( $T_t$ ), then mathematically the Holt-Winter forecasting with Multiplicative seasonality is represented as:

$$S_t = \alpha \left( \frac{D_t}{I_{t-p}} \right) + (1 - \alpha)(S_{t-1}) + T_{t-1}$$

$$I_t = \gamma \left( \frac{D_t}{S_t} \right) + (1 - \gamma)(S_{t-1}) + I_{t-p}$$

$$T_t = \beta (S_t - S_{t-1}) + (1 - \beta)T_{t-1}$$

Here  $\alpha$ ,  $\gamma$  and  $\beta$ , are considered as smoothing constants and  $p$  is the number of observation points in a cycle, for example  $p = 4$  for quarterly data.

The Holt Winter's forecasting method is simple exponential smoothing, which is a special case of ARIMA models, namely the IMA (0,1,1) model. This paper use sktime[41], a Python library which is a unified framework for machine learning with time series and compatible with scikit-learn[42] (an open-source machine learning library).Sktime provides algorithm for Holt-Winters triple exponential smoothing method to select the model order and estimate the parameters.

**4.4K-Nearest Neighbor (k-NN)**

K Nearest Neighbor [43]is an algorithm that creates a database of all available features and forecasts the numerical target, which is computed from the similarity measure calculations (e.g., distance functions). The initial step for forecasting time-series in K-NN approach involves development of database with a set of features to be used relative to 'present' conditions. For a uni-variate dataset like the one used in this paper, examples of features used that can be identified are time series values, average time series values, cumulative time series sum, entropy of time series, etc.

Brief explanation of the implementation of k-NN algorithm is as follows. Consider a vector  $p$ , where  $n$  features are patterns for time  $t$  and having components  $p_i, i = 1, \dots, n$ . Assume that matrix  $A_{ij}$  consists of features for historical data. In this matrix each row corresponds to the vector of features for each time in the historical data set.

The following two steps are required to determine the k nearest neighbor to  $p_i$ :

1. Compute the distance between  $p_i$  and all the rows from  $A_{ij}$  from Equation6. There are different methods that are used to calculate the distance such has Euclidian distance or Manhattan distance. In this paper we used the Euclidian distance.

$$d_j = \sum \left( \sqrt{(p_i - A_{ij})^2} \right) \tag{6}$$

2. From Equation 7, extract the k indices returning the smallest element of  $d_j$  and its related k time stamps  $(\tau_1, \dots, \tau_k)$

$$Z_t + \Delta_t = \frac{\sum_{i=1}^k \alpha Z_{\tau_i + \Delta_t}}{\sum_{i=1}^k \alpha_i} \tag{7}$$

This means that the prediction parameter  $Z_t + \Delta_t$  is determined from a linear combination of time series values following the nearest neighbor $\tau_i$ . The weights  $\alpha_i$ are also depending on the distance  $d_j$ .

**4.5Random Forest model**

Random Forest model [44]make use of decision trees to capture the non-linearity in the data by dividing the space into small sub-spaces depending on the problem. The Random Forest is an ensemble machine learning algorithm called Bootstrap Aggregation or bagging and more specifically a randomized aggregated ensemble of decision tree. An ensemble technique incorporates the predictions from several machine learning algorithms to make predictions more accurate than any single model.

Decision tree model the data with high variance, which results in over fitting models and hence not effective in generalizing the learned rules. The Random Forest model addresses this limitation by training a collection of trees instead of a single tree. In order to ensure randomness, the RF approach, introduce randomness at different levels. For example, introducing random sampling of the training dataset for each tree, RF technique has shown good efficiency in both regression and classification problems.

This paper uses sktime[41]python library compatible with scikit-learn.'KNeighborsRegressor' class in scikit-learn provide APIs for constructing a forest of random trees.

**4.6Support Vector Regression (SVR)**

The adaptation of Support Vector Machines (SVM) concepts for regression is known as Support Vector Regression (SVR)[45]. The approach of SVR operates by finding the best possible line to minimize a cost function error, using data instances known as Support Vectors. It is achieved using a method of optimization that considers only those data instances that are nearest to line with the minimum cost, in the training dataset.

For the current problem of wind speed forecasting this paper employs a special approach known as  $\epsilon$ -SV regression algorithm (a derivative method of SVRalgorithm). Consider the training set  $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset \chi \times \mathbb{R}$ , where  $\chi$  denotes the space of the input patterns. The  $\epsilon$ -SV regression algorithm[46] operates by finding the function  $f(x)$  that has at most  $\epsilon$  deviation from the actually obtained targets  $y_i$  for all the training data, and is as flat as possible at the same time. Using  $\epsilon$ -SV, the current problem of wind speed forecasting is given by the following Equation 8, by first considering the case of linear functions  $f$ , taking the form  $f(x)$ , as follows:

$$f(x) = \langle \omega, x \rangle + b \tag{8}$$

With  $\omega \in c, b \in \mathbb{R}$ , where  $\langle \dots \rangle$  is the dot product in  $\chi$ .

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \tag{9}$$

$x(i)$ is the input vector, the terms  $\alpha_i, \alpha_i^*$  are Lagrange multipliers,  $b \in \mathbb{R}$  and  $G(x_i, x_j)$  is a kernel function.

As formulated by Equation 9,  $\omega$  is expressed as a linear combination of the training patterns  $x_i$ . It can be inferred from this equation that the complexity of the representation of a function by SVs depends only on the number of SVs and not

on the dimensionality of the input space  $\chi$ . Some of the kernels approaches that can be applied are polynomial, hyperbolic tangent and Gaussian radial basis function.

In this paper, we used `sktime`[41]`python` library compatible with `scikit-learn`, specifically SVR - a epsilon-Support Vector Regression from `scikit-learn`. The Polynomial Kernel was applied with exponent of 1 and value of 0.002 for the epsilon parameter of the epsilon insensitive loss function [47].

**4.7 Linear Regression (LR)**

LR approach only supports regression type problems and can be perceived as intersection between ML methodologies of RF, kNN, SVR, and statistically sophisticated model of ARIMA. LR algorithm computes the coefficients for a line or hyperplane which best fits the training data. This method proves to be efficient if the output variable for the data is a linear combination of the inputs [48]. In this paper, we used `sktime`[41]`python` library compatible with `scikit-learn`[42], specifically Linear Regression - Ordinary least squares Linear Regression class from `scikit-learn`.

**Table 1:** Information of site Yanbu, Saudi Arabia and wind speed characteristics

Information	Details	
Site	Yanbu	
Latitude (North)	24° 07'	
Longitude (East)	38° 03'	
Elevation (m)	10	
Wind speed details (m/s)	Mean	4.425
	Min	0.001
	Max	23.559

**Table 2:** Training and Test samples for each season

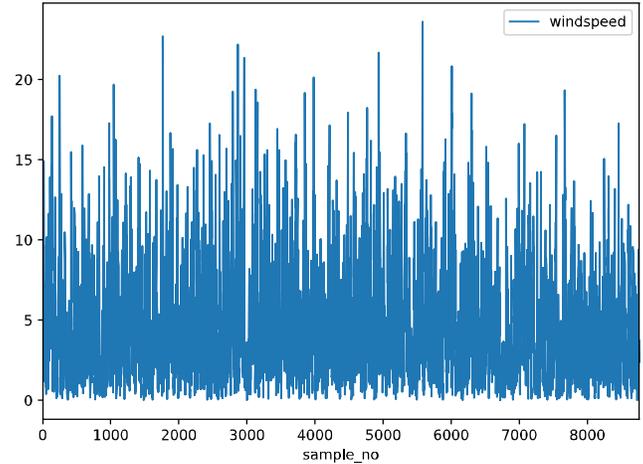
Season	Winter	Spring	Summer	Autumn
Training Data Period	1 Jan - 1 March	16 March - 3 May	16 May - 22 Aug	16 Sept - 27 Nov
Training Samples	1420	1171	2361	1747
Test Data Period	2 March - 15 March	4 May - 15 May	23 Aug - 15 Sept	28 Nov - 15 Dec
TestSamples	355	292	590	436

**5. WIND SPEED DATASET**

The Kingdom of Saudi Arabia (KSA) is basically a desert land with hot summers and cold winters. Climatic conditions determine availability of wind energy at a given site. Wind power systems are characterized by availability of wind speed resources. Yanbu is an industrial city that plays a major role in the economy of Saudi Arabia. Yanbu is a highly potential wind resource location for exploitation of wind energy and for

deployment of wind power systems. The long term wind speed data used in the present study covers the period 1970-1982 [49] and is shown in Figure 1 and the wind speed characteristics are presented in Table-1. This data has been used for training models of the selected algorithms.

The details of period of each the season and information of Training and Test data selected for each season is depicted in Table-2. We selected 75% of hourly data instances as training data and 25% samples as testing data in each season.



**Figure 1:** Long-term (1970-1982) average wind speed for different months of Yanbu, Saudi Arabia.

**Table 3: Optimum Parameters selected for each Algorithm**

Algorithm	Details of Optimal parameters
KNN [43]	Number of nearest neighbors (k)=5 and Nearest neighbor search algorithm with Euclidean Distance.
RF [44]	The maximum depth of the tree = 0 for unlimited, Number of trees in the random forest N=500.
SVR [50][45]	The complexity constant C=1. The Polynomial Kernel was applied and the epsilon parameter in epsilon-insensitive loss function L=0.001.
Holt-Winter [40]	Number of seasons for the seasonal cycle CL=4 , Seasonal and Trend smoothing factor both set to 0.2.
LR	Use M5 [51] method for attribute selection and setting to remove highly correlated input attributes.

**6. RESULTS AND DISCUSSION**

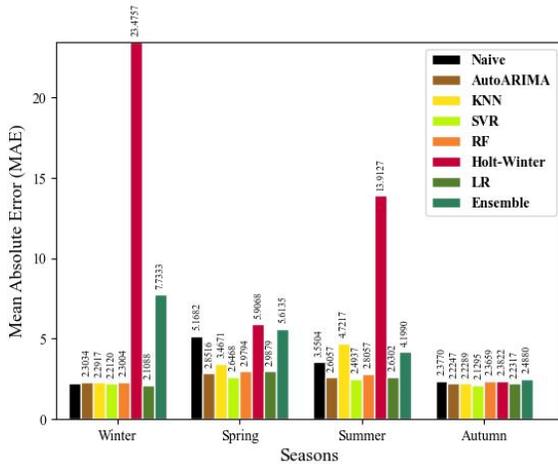
As shown in Table-2 each model is trained with 75% of hourly data instances and evaluated with 25% samples as testing data in each season. Figure2, Figure 3, Figure4 and Figure5 show the Mean Absolute Error(MAE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE) and Symmetric Mean Absolute Percentage Error (sMAPE) respectively, for one-step ahead forecasting after employing each algorithm in four seasons. We used the ‘GridSearchCV’ class from `scikit-learn`[42] in order to do Exhaustive search over specified parameter values for an estimator, to look for the optimum selected parameter in each algorithm (as shown in Table-3). Moreover, Table-4 reports,

the time taken (in seconds) to train the model for each algorithm with the best optimal parameters with sktime[41] library. It also reports the average time (inseconds) for each investigated algorithm to make a 1-hour ahead prediction.

**Table 4 : Average Training and Prediction Time for test data of Wind Speed Forecasting in each algorithm**

Algorithm	Training (fit) time with optimal parameters (sec)	Prediction time with optimal parameters (sec)
KNN [43]	0.6405	0.3009
RF [44]	2.2490	5.1820
SVR [50][45]	0.7895	0.0825
Holt-Winter [40]	0.4740	0.05058
LR	29.5184	0.0539
Ensemble	1.1358	0.1090

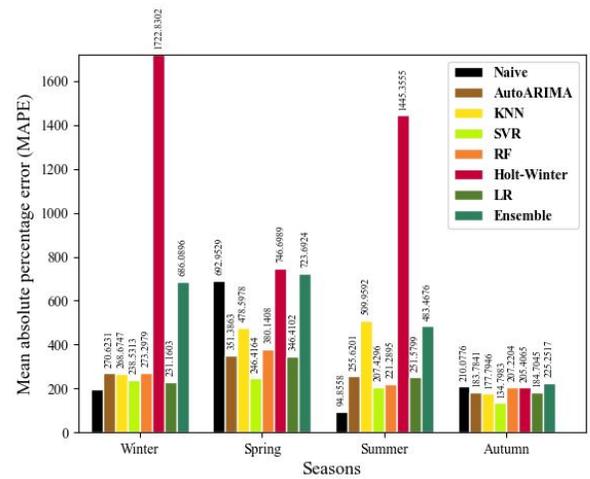
Results indicate that SVR algorithm has the impressive overall performance. Compared to other algorithms, the relative error (SMAPE) of SVR is considerably better. With respect to MAE, SVR is comparable to that of ARIMA and RF.



**Figure 2:** LongMean Absolute Error(MAE) for test data in four seasons for each algorithm.

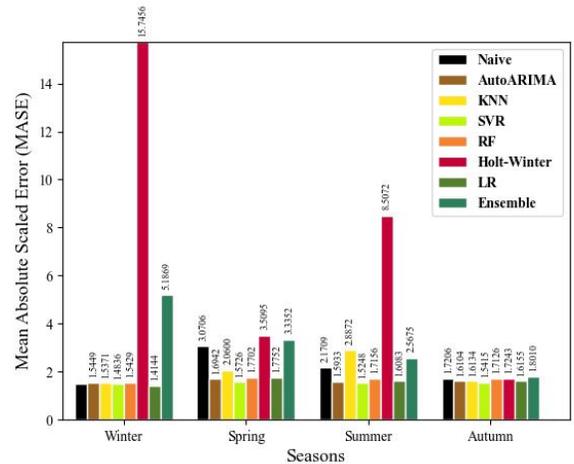
It can be noted from the results that, while all three algorithms RF, KNN, and SVR have comparable absolute errors (MAE), their relative error (SMAPE) is comparatively different.LR algorithm makes use of the same input attributes as other ML algorithms by considering the daily periodicity trend which result in significantly better performance than Holt-Winters algorithm. The Holt-Winters approach shows poor performance according to these two parameters, even-though Holt-Winters algorithm takes into account trend and seasonality to forecast future values.

As shown in the Table-3 it is important to select optimum parameter in each algorithm by trial and error based on the dynamics of wind speed values that will lead to identify local patterns in short-term time series forecasting rather than global patterns, depending on the tuning parameter of specific algorithm.



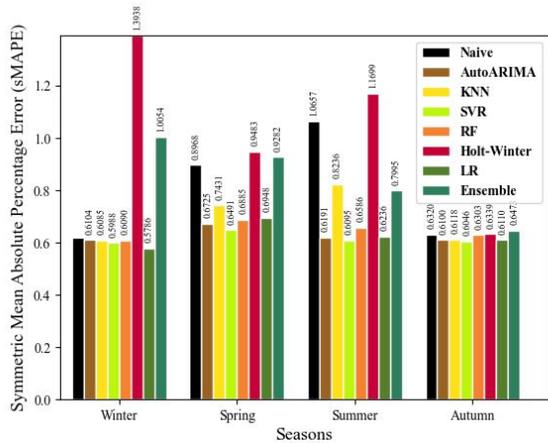
**Figure 3:** Mean Absolute Percentage Error (MAPE) for test data in four seasons for each algorithm.

Since this paper considers only univariate data, the training period is relatively small and overall training time is less than 3 seconds. Also, the samples at the end of the time series in each season are considered for validation of the model. The second column in Table-4 reports the average time (in milli-seconds) taken for 1-hour ahead wind speed forecasting in each algorithm, once they are provided with the query. It should be noted that apart from RF based approach, each prediction model with specific algorithm responds quickly (fractions of second), in less than few seconds limit for short-term wind prediction. Because of the lazy learning-based approach followed in kNN, the training time is short, but the response time increases, as kNN searches the whole training dataset.



**Figure 4:** Mean Absolute Scaled Error (MASE) for test data in four seasons for each algorithm.

The last column in Table-4 reports, the training time of models in each algorithm and it is found that training time increases depending on the complexity of the model and hardware configuration used or training and therefore the training time for LR and RF approach is highest, whereas Holt-Winter and kNN has lowest training time. This paper conducted experiments using sktime[41] library with scikit-learn[42] on an Intel Core i7 CPU, 2.3 GHz Quad with 16MB RAM.



**Figure 5:** Symmetric Mean Absolute Percentage Error (sMAPE) for test data in four seasons for each algorithm.

## 7. CONCLUSIONS AND RECOMMENDATIONS

The outcomes of wind speed forecasts with the different algorithms studied suggest that performance of each algorithm can be different in its output under various error definitions. The wind speed forecasting using different algorithm indicate that the machine learning-based algorithms perform better than the statistical approaches for the performance criteria of MAE and SMAPE. More specifically the forecasting results using approaches of RF, SVR, and kNN showed significantly higher performance than the statistical methods of Holt-Winter and ARIMA algorithms. Out of the evaluated ML-based approaches the performance of SVR and RF is better than kNN. However, KNN based model tends to be simpler because of fewer parameters to be tuned.

## ACKNOWLEDGEMENT

The authors acknowledge the support of Center for Communication & IT Research and Center for Engineering Research, Research Institute, King Fahd University of Petroleum and Minerals for this research work.

## REFERENCES

[1] A. Ucar and F. Balo, "Investigation of wind characteristics and assessment of wind-generation potentiality in Uludağ-Bursa, Turkey," *Appl. Energy*, vol. 86, no. 3, pp. 333–339, 2009.

[2] G. Xydis, C. Koroneos, and M. Loizidou, "Exergy analysis in a wind speed prognostic model as a wind farm sitting selection tool: a case study in Southern Greece," *Appl. Energy*, vol. 86, no. 11, pp. 2411–2420, 2009.

[3] U. Cali and V. Sharma, "Short-term wind power forecasting using long-short term memory based recurrent neural network model and variable selection," *Int. J. Smart Grid Clean Energy*, vol. 8, no. 2, pp. 103–110, 2019.

[4] M. Monfared, H. Rastegar, and H. M. Kojabadi, "A new strategy for wind speed forecasting using artificial intelligent methods," *Renew. energy*, vol. 34, no. 3, pp. 845–848, 2009.

[5] P. Pinson, H. A. Nielsen, H. Madsen, and G.

Kariniotakis, "Skill forecasting from ensemble predictions of wind power," *Appl. Energy*, vol. 86, no. 7–8, pp. 1326–1334, 2009.

[6] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renew. Sustain. Energy Rev.*, vol. 13, no. 4, pp. 915–920, 2009.

[7] A. Tascikaraoglu and M. Uzunoglu, "A review of combined approaches for prediction of short-term wind speed and power," *Renew. Sustain. Energy Rev.*, vol. 34, pp. 243–254, 2014.

[8] A. Sfetsos, "A comparison of various forecasting techniques applied to mean hourly wind speed time series," *Renew. energy*, vol. 21, no. 1, pp. 23–35, 2000.

[9] K. B. Debnath and M. Mourshed, "Forecasting methods in energy planning models," *Renew. Sustain. Energy Rev.*, vol. 88, pp. 297–325, 2018.

[10] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renew. Sustain. Energy Rev.*, vol. 31, pp. 762–777, 2014.

[11] M. Lange and U. Focken, *Physical approach to short-term wind power prediction*, vol. 208. Springer, 2006.

[12] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econom. Rev.*, vol. 29, no. 5–6, pp. 594–621, 2010.

[13] J. M. Morales, R. Minguez, and A. J. Conejo, "A methodology to generate statistically dependent wind speed scenarios," *Appl. Energy*, vol. 87, no. 3, pp. 843–855, 2010.

[14] J. L. Torres, A. Garcia, M. De Blas, and A. De Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Sol. energy*, vol. 79, no. 1, pp. 65–77, 2005.

[15] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renew. Energy*, vol. 34, no. 5, pp. 1388–1393, 2009.

[16] H. Liu, E. Erdem, and J. Shi, "Comprehensive evaluation of ARMA-GARCH (-M) approaches for modeling the mean and volatility of wind speed," *Appl. Energy*, vol. 88, no. 3, pp. 724–732, 2011.

[17] I. Marović, I. Sušanj, and N. Ožanić, "Development of ANN model for wind speed prediction as a support for early warning system," *Complexity*, vol. 2017, 2017.

[18] L. Lin, J.-T. Eriksson, H. Vihriala, and L. Soderlund, "Predicting wind behavior with neural networks," in *Proceedings the European Wind Energy Conference*, 1996, pp. 655–658.

[19] Y. Ren and P. N. Suganthan, "Empirical mode decomposition-k nearest neighbor models for wind speed forecasting," *J. Power Energy Eng*, vol. 2, no. 04, pp. 176–185, 2014.

[20] Y. Wen, M. Song, and J. Wang, "A combined AR-KNN model for short-term wind speed forecasting," in *2016 IEEE 55th Conference on*

- Decision and Control (CDC)*, 2016, pp. 6342–6346.
- [21] A. Lahouar and J. B. H. Slama, “**Hour-ahead wind power forecast based on random forests**,” *Renew. energy*, vol. 109, pp. 529–541, 2017.
- [22] G.-R. Ji, P. Han, and Y.-J. Zhai, “**Wind speed forecasting based on support vector machine with forecasting error estimation**,” in *2007 International Conference on Machine Learning and Cybernetics*, 2007, vol. 5, pp. 2735–2739.
- [23] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, “**Support vector machines for wind speed prediction**,” *Renew. energy*, vol. 29, no. 6, pp. 939–947, 2004.
- [24] H. G. Beyer, T. Degner, J. Hausmann, M. Hoffmann, and P. Rujan, “**Short term prediction of wind speed and power output of a wind turbine with neural networks**,” in *Proceedings of the EWEC*, 1994, vol. 94, pp. 349–352.
- [25] M. A. Ghorbani, R. Khatibi, M. H. FazeliFard, L. Naghypour, and O. Makarynsky, “**Short-term wind speed predictions with machine learning techniques**,” *Meteorol. Atmos. Phys.*, vol. 128, no. 1, pp. 57–72, 2016.
- [26] G. N. Kariniotakis, G. S. Stavrakakis, and E. F. Nogaret, “**Wind power forecasting using advanced neural networks models**,” *IEEE Trans. Energy Convers.*, vol. 11, no. 4, pp. 762–767, 1996.
- [27] A. More and M. C. Deo, “**Forecasting wind with neural networks**,” *Mar. Struct.*, vol. 16, no. 1, pp. 35–49, 2003.
- [28] S. Wang, N. Zhang, L. Wu, and Y. Wang, “**Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method**,” *Renew. Energy*, vol. 94, pp. 629–636, 2016.
- [29] G. Kariniotakis, E. Nogaret, and G. Stavrakakis, “**A fuzzy logic and a neural network based wind power forecasting model**,” in *EU Wind Energy Conference*, 1996, pp. 596–599.
- [30] X. Wang, G. Sideratos, N. Hatziargyriou, and L. H. Tsoukalas, “**Wind speed forecasting for power system operational planning**,” in *2004 International conference on probabilistic methods applied to power systems*, 2004, pp. 470–474.
- [31] T. G. Barbounis and J. B. Theocharis, “**A locally recurrent fuzzy neural network with application to the wind speed prediction using spatial correlation**,” *Neurocomputing*, vol. 70, no. 7–9, pp. 1525–1542, 2007.
- [32] H. Peng, F. Liu, and X. Yang, “**A hybrid strategy of short term wind power prediction**,” *Renew. Energy*, vol. 50, pp. 590–595, 2013.
- [33] G. R. Kishore, V. Prema, and K. U. Rao, “**Multivariate wind power forecast using artificial neural network**,” in *2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS)*, 2014, pp. 159–163.
- [34] Q. Wang, K. K. Lai, D. Niu, and Q. Zhang, “**A multivariate wind power forecasting model based on ls-svm**,” in *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, 2012, pp. 127–131.
- [35] M. B. Pfeifer, “**A hybrid approach to forecasting wind power using Artificial Neural Networks and Numeric Weather Prediction**,” Wichita State University, 2011.
- [36] N. Hamid and W. C. Wibowo, “**Wind Speed Forecasting Using Multivariate Time-Series Radial Basis Function Neural Network**,” in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2018, pp. 423–428.
- [37] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “**Deep learning for time series classification: a review**,” *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, 2019.
- [38] R. J. Hyndman and A. B. Koehler, “**Another look at measures of forecast accuracy**,” vol. 22, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.
- [39] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [40] C. Chatfield, “**The Holt-winters forecasting procedure**,” *J. R. Stat. Soc. Ser. C (Applied Stat.)*, vol. 27, no. 3, pp. 264–279, 1978.
- [41] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, “**sktime: A Unified Interface for Machine Learning with Time Series**,” no. NeurIPS, 2019, [Online]. Available: <http://arxiv.org/abs/1909.07872>.
- [42] F. Pedregosa *et al.*, “**Scikit-learn: Machine Learning in {P}ython**,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [43] D. Aha and D. Kibler, “**Instance-based learning algorithms**,” *Mach. Learn.*, vol. 6, pp. 37–66, 1991.
- [44] L. Breiman, “**Random Forests**,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [45] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, “**Improvements to the SMO Algorithm for SVM Regression**,” 1999.
- [46] V. N. Vapnik, “**The nature of statistical learning**,” *Theory*, 1995.
- [47] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, “**Improvements to the SMO algorithm for SVM regression**,” *IEEE Trans. neural networks*, vol. 11, no. 5, pp. 1188–1193, 2000.
- [48] L. Breiman and J. H. Friedman, “**Predicting multivariate responses in multiple linear regression**,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.)*, vol. 59, no. 1, pp. 3–54, 1997.
- [49] J. M. Al-Ansari, H. Bakhsh, and K. I. Madni, “**Wind energy atlas for the kingdom of Saudi Arabia**,” *King Fahd Univ. Pet. Miner. Press. Dhahran*, 1986.
- [50] A. J. SMOLA and B. SCHOLKOPF, “**A tutorial on support vector regression**,” *Stat. Comput.*, vol. 14, pp. 199–222, 2004, doi: 10.1023/B:STCO.0000035301.49549.88.
- [51] I. H. Witten and Y. Wang, “**Induction of model trees for predicting continuous classes**,” 1997.