Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering

Basılı/Printed ISSN: 1302-9304. Elektronik/Online ISSN: 2547-958X

Wind Power Generation Prediction Using Machine Learning Algorithms

Makine Öğrenmesi Algoritmalarını Kullanarak Rüzgar Enerjisi Üretimi Tahmini

Özlem Ece Yürek [1], Derya Birant [2*], İsmail Yürek [3]

[1] Dokuz Eylul University, The Graduate School of Natural and Applied Sciences, Izmir, TURKEY
[2] Dokuz Eylul University, Faculty of Engineering, Department of Computer Engineering, Izmir, TURKEY
[3] Semafor Teknoloji, Izmir, TURKEY
Sorumlu Yazar / Corresponding Author *: derya@cs.deu.edu.tr

Geliş Tarihi / Received: 15.04.2020 *Araştırma Makalesi/Research Article*
Kabul Tarihi / Accepted: 21.06.2020 *DOI:10.21205/deufmd.2021236709*
Atıf şekli/ How to cite: YÜREK, Ö.E., BIRANT, D.,YÜREK, İ.(2021). Wind Power Generation Prediction Using Machine Learning Algorithms. DEUFMD, 23(67), 107-119.

Abstract

Renewable energy becomes progressively popular in the world because renewable resources such as solar, geothermal, wind energy are clean, inexhaustible and come from natural sources. Wind energy is one of the most significant resources of renewable energy and it plays a key role in the generation of electricity. Thus, accurate wind power estimation is crucial to deal with the challenges to balance energy trading, planning, scheduling decisions and strategies of wind power generation. This study proposes a prediction model to solve a real-life problem in the renewable energy sector by accurately estimating the amount of wind energy production per hour in the next 24 hours by applying machine learning (ML) techniques using historical wind power generation data and weather forecasting reports. In the proposed approach, first, an unsupervised ML method (i.e., the K-Means clustering algorithm) is applied to group data into meaningful clusters; then, these clusters are accepted as new feature values and added to the dataset to enlarge it; finally, a supervised ML method (i.e., regression) is performed for prediction. This study compares nine supervised learning algorithms: K-Nearest Neighbors, Support Vector Regression, Random Forest, Extra Trees, Gradient Boosting, Ridge Regression, Least Absolute Shrinkage and Selection Operator, Decision Tree, and Convolutional Neural Network. The aim of this study is to investigate the success of different ML algorithms on real-world data of wind turbines and propose a methodology to benchmark various machine learning algorithms to choose the most accurate final model for wind power generation prediction.

Keywords: *Machine learning, Wind power generation prediction, Wind power forecasting, Renewable energy*

Öz

Yenilenebilir enerji dünyada giderek popüler hale gelmektedir, çünkü güneş, jeotermal, rüzgar enerjisi gibi yenilenebilir kaynaklar temiz, tükenmez ve doğal kaynaklardır. Rüzgar enerjisi, yenilenebilir enerjinin en önemli kaynaklarından biridir ve elektrik üretiminde kilit rol oynamaktadır. Bu nedenle, rüzgar enerjisi üretiminin doğru tahmin edilmesi enerji ticareti, planlama, zamanlama kararları ve rüzgar enerjisi üretim stratejilerini dengeleme zorluklarıyla başa çıkmada çok önemlidir. Bu çalışma, tarihsel rüzgar enerjisi üretim verileri ve hava durumu tahmin raporlarını kullanarak yenilenebilir enerji sektöründeki gerçek yaşam sorununu, önümüzdeki 24 saat için saat başına rüzgar enerjisi üretim miktarını makine öğrenmesi (ML) teknikleri ile doğru bir şekilde tahmin edebilmek için bir model önermektedir. Önerilen yaklaşımda; ilk olarak, veri setini anlamlı kümeler halinde gruplamak için denetimsiz bir ML yöntemi (K-Means kümeleme algoritması) uygulanır; daha sonra, bu kümeler yeni öznitelik değerleri olarak kabul edilir ve veri setini büyütmek için eklenir; son olarak, tahmin için denetimli bir ML yöntemi (regresyon) gerçekleştirilir. Bu çalışma dokuz denetimli öğrenme algoritmasını karşılaştırmaktadır: K-En Yakın Komşu, Destek Vektör Regresyonu, Rastgele Orman, Ekstra Ağaçlar, Gradyan Artırma, Ridge Regresyon, En Küçük Mutlak Daralma ve Seçme Operatörü, Karar Ağacı, ve Konvolüsyonel Sinir Ağı. Bu çalışmanın amacı, rüzgar türbinlerinin gerçek dünya verileri üzerindeki farklı ML algoritmalarının başarısını araştırmak ve rüzgar enerjisi üretimi tahmini için en doğru nihai modeli seçmek üzere çeşitli makine öğrenmesi algoritmalarını karşılaştırmak için bir metodoloji önermektir.

Anahtar Kelimeler: *Makine öğrenmesi, Rüzgar enerjisi üretimi tahmini, Rüzgar enerjisi tahmini, Yenilenebilir enerji*

1. Introduction

The world's demand for energy is increasingly growing day by day. Renewable energy has attracted attention globally because resources of renewable energy such as solar, wind, geothermal, biomass energy and hydropower are clean, green, free of costs, low carbon and naturally exist in a wide geographical area. Furthermore, usage of these renewable resources decreases environmental pollution by protecting the ecological environment and they can be recycled in nature.

The wind is one of the most useful and important resources of renewable energy. Wind energy has the opportunity to produce power for every hour and it is a clean and popular way for electricity generating owing to its wide availability. Therefore, wind turbines have a crucial role in the electricity generation portfolio worldwide.

However, wind power forecasting is hard to predict because the wind speed is a weather depended parameter and it is highly unstable, random and volatile. It shows strong randomness in a short period time due to its unstable nature and uncontrollability of the wind flows. Random wind power generation causes an imbalance between power generation and consumption, so people who use this energy are affected by the increase of unstable costs due to low predictability [1]. Therefore, it is obviously seen that accurate prediction of wind power is essential for energy management purposes such as making appropriate generation, distribution, transmission, planning, and scheduling. Besides, accurate wind power prediction improves the usage rate of wind energy [2]. To accomplish this goal, machine learning (ML) is playing a critical role in the energy sector. ML techniques have been widely used to interpret the historical data and then to predict the future so as to increase the prediction performance of wind power generation forecasting.

In particular, wind power generation enterprises must notify the Republic of Turkey Energy Market Regulatory Authority (EMRA) at one-hour intervals about the amount of energy they will generate within the next 24 hours each morning. When businesses produce the amount of energy they reported, they can sell the produced electricity at the highest price. If energy production is made below or above the reported amount, the unit prices are reduced by EMRA. For this reason, an accurate estimation of energy to be produced in 24 hours has become very important in terms of profitability. The main objective of this study is to solve a real-life problem in the renewable energy sector by accurately estimating the amount of energy production per hour in the next 24 hours by applying machine learning techniques using historical wind energy production data and weather forecasting reports of the enterprise.

Some enterprises calculate and estimate 24-hour production values based on human interpretation. However, it is a time-consuming process and strongly depends on the personnel responsible for this task. For this reason, in this study, we propose an approach that automatically predicts wind power generation.

This study proposes an approach that consists of two main stages. In the first stage, an unsupervised ML method (K-Means) is applied to group data into meaningful clusters and then, these clusters are accepted as new feature values and added to the dataset. In the second stage, regression analysis is performed as a supervised ML technique using hourly electricity generation and predicted weather values, such as wind speed, cloud cover, wind direction, air temperature, and pressure. In order to determine the most accurate wind power prediction model, we compared various ML algorithms, including K-Nearest Neighbors (KNN), Support Vector Regression (SVR), Random Forest (RF), Gradient Boosting (GB), Extra Trees (ET), Least Absolute Shrinkage and Selection Operator (LASSO), Ridge Regression (RR), Decision Tree (DT), and Convolutional Neural Network (CNN). We compared the performances of algorithms in a case study for the wind energy power prediction by using a real-world wind power generation data that covers a period of two years from 2017 to 2018 of wind turbines located in Turkey.

The paper is organized as follows. Section 2 summarizes the related works in the literature on the subject. In Section 3, the proposed approach with its advantages and methods applied to predict wind power energy production are explained. Section 4 is the section in which the implementation details, the experimental results, and the evaluation of these results are discussed. Finally, Section 5 concludes the paper and gives possible future research directions.

2. Related Work

In recent years, studies about wind power generation forecasting are becoming more and more popular due to the strengths of wind power as a renewable energy resource. While some studies focused on the short-term forecasting [3-6], the others worked on the 24-h ahead prediction [1,7,8]. Nowadays, deep learning algorithms have been widely applied to predict wind power generation. For instance, Zhang et al. [3] applied Long Short-Term Memory (LSTM) algorithm and used the Gaussian Mixture Model (GMM) to understand the characteristics of error distribution for wind turbine power forecasting and stated that LSTM improves the forecasting accuracy. Again Zhang et al. [2] applied LSTM, but this time Auto Encoder (AE) algorithm was utilized to reduce the data dimension before entering into LSTM. Their results showed that the accuracy of the AE-LSTM was nearly equal to the LSTM model, but the training and prediction time of the AE-LSTM was much smaller than the LSTM. In addition, their results were compared with the success of the Support Vector Machine (SVM) model. It was seen that the accuracy of the LSTM model prediction was higher than SVM. Likewise, Cali and Sharma [4] applied LSTM based Recurrent Neural Network (LSTM-RNN) in order to predict 1 to 24 hours ahead of wind power.

Hong et al. [7] presented CNN based prediction model which was cascaded with a three-layered Radial Basis Function Neural Network (RBFNN) for 24 h-ahead wind power generation prediction. In their study, Double Gaussian Function (DGF) was used as its activation function. The characteristics of wind power were extracted by convolution, pooling and kernel operations of CNN. After that, these characteristics were fed into a new RBFNN as inputs. Adaptive Moment Estimation (ADAM) was applied so as to optimize the parameters of both RBFNN and CNN algorithms. Another study conducted by Dolara et al. [1] demonstrated the construction of prediction models for a wind farm power generation with 24 hours' horizon. The main goal of their paper is to construct an accurate wind power prediction model by applying the Feedforward Neural Networks (FFNN) algorithm. Real data was used from a wind farm which is located in Southern Italy. Their obtained results were compared with the predictions provided by a commercial weather service using numerical weather prediction models. In another study, Ma and Zhai [8] presented a new approach based on the hybridization of FFNN, Ant Colony Optimization algorithm (ACO), and Wavelet Transform (WT) to predict for 24 h-ahead wind energy generation. Their study aimed to predict the wind speed in the first stage and then the predicted future speed was entered to the second stage to forecast future power generation.

Sanz et al. [9] concentrated on the prediction task for renewable energy applications which involve feature selection. They introduced a new approach for feature selection based on the

Coral Reefs Optimization algorithm to improve the prediction success of the proposed system. In addition, they applied the Extreme Learning Machine (ELM) algorithm for prediction. In their study, real-world data obtained from a wind farm in Spain was used. Besides, different time-horizons such as hourly and daily were considered so as to improve the ELM prediction performance and it was shown that the hourly time-horizon prediction was better. Their results showed that applying feature selection algorithms improved the performance in renewable energy-related predictions at different time horizons. It was observed that a 20% increase in hourly and daily wind speed prediction was achieved based on the comparison of the systems without the feature selection process with the proposed system.

SVM and SVR algorithms have also been used in this area to predict accurate wind power forecasting [2,5,10,11,12]. Zhang et al. [10] studied the improved SVM wind power prediction based on the Genetic Algorithm (GA). In their work, GA was applied to optimize the parameters and Radial Basis Function (RBF) was chosen as the kernel function in SVM modeling. Their results showed that the proposed GA-SVM model had a better prediction performance than the model constructed with default parameters. Another study conducted by Li et al. [5] combined SVM and improved dragonfly algorithm to predict wind power generation. The improved dragonfly algorithm was applied to optimize the input parameters of SVM. In their study, the real-world dataset obtained from a wind farm in France was used. Their results were compared with the Gaussian process regression and backpropagation neural network and it was stated that the proposed model had better prediction performance.

Demolli et al. [11] conducted several experiments to investigate the performances of ML algorithms based on the daily wind speed dataset. xGBoost, KNN, SVR, LASSO, and RF algorithms were applied and finally, it is pointed out that the success of the SVR algorithm was more satisfying than other algorithms for estimating long-term wind power values.

Okumus and Dinler [6] worked on 1h-ahead forecasting of wind power generation by combining Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS) algorithms. The input data was gathered from three different sites that are located in Turkey, Amasra, Bandırma, and Selçuk. Also, Kramer et al. [12] applied SVR and self-organizing maps algorithms to show these algorithms have an important role in monitoring and predicting renewable energy. Marvuglia and Messineo [13] concentrated on detecting the abnormalities of the wind turbines power curve. Three different machine learning algorithms Generalized Mapping Regressor (GMR), FFNN and a General Regression Neural Network (GRNN) were used and compared in order to predict the relation between the wind speed and the generated power in a wind turbine. This relation was represented as a curve that has a logistic function shape. The main goal of their paper was to detect the abnormalities of the wind turbines' power curve.

Wang et al. [14] prepared a detailed review and discussion of renewable energy prediction ways about several deep learning algorithms. They aimed to see its efficiency, strength and application potential. In their paper, current researches, difficulties, and future investigation directions were presented.

3. Material and Method

3.1. The proposed approach

This study proposes an approach that consists of two main stages. In the first stage, a clustering method is applied to divide data into meaningful groups and then, these clusters are accepted as new feature values and added to the dataset. In the second stage, regression analysis is performed for prediction.

Figure 1 shows the general flow of the proposed approach. Firstly, the dataset is generated by putting together weather forecast reports and wind power generation amount for each day within a particular period. After generating the dataset, data preprocessing steps such as data cleaning, data reduction, and data transformation are applied to improve the quality of data before feeding it to machine learning algorithms. After preprocessing steps, a clustering algorithm (i.e. K-Means) is used so as to group data into meaningful clusters. After that, these clusters are added to dataset in the feature generation process. In the next step, various machine learning algorithms are implemented to build the classification models in order to predict wind power generation. After all, the success of each ML algorithm is measured based on different evaluation metrics and their results are presented. Finally, the best model is chosen to assist experts in predicting the generated amount of electricity that will be produced from wind energy. In this way, the power plant will minimize the loss caused by incorrect estimation and will become competitive in energy pricing thanks to accurate wind power prediction.

3.2. The advantages of the proposed approach

Some enterprises calculate and estimate 24-hour production values based on human interpretation. However, it takes a lot of time to perform the estimation process manually and there are difficulties in estimating the amount of energy to be produced in the absence of the personnel responsible for this task. To overcome these problems, this study aims to construct an accurate wind power prediction model by applying machine learning algorithms that will automatically estimate wind power generation.

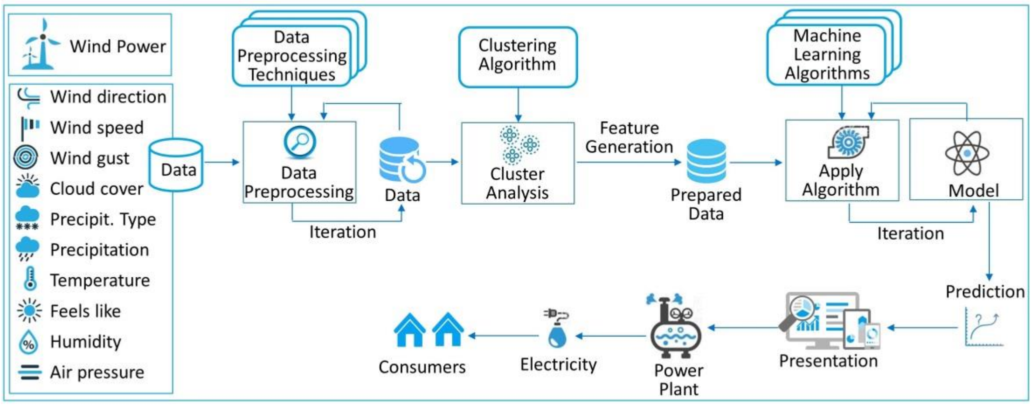


Figure 1. Overview of the proposed approach

This study provides the following advantages to the enterprise.

- A complete machine learning-based prediction will be performed by eliminating human interpretation. In this way, organizational memory will be transferred to digital systems.
- More accurate forecasts will be made through the analysis of large historical data.
- The forecasting process will take place in a shorter time than the current situation. Through, both the time will be saved and the staff responsible for the forecasting work will be free from exhausting and routine work.
- Since the learning process will be repeated in certain periods, the system will be able to make more successful predictions day by day.

3.3. Machine learning algorithms

The supervised ML algorithms applied in this study are briefly described as follows.

K-Nearest Neighbors (KNN) is a popular supervised ML algorithm that is used for both regression and classification. Data points are classified based on defined *k* value as the number of neighbors to be considered. KNN calculates similarities between data points and accepts that similar objects located near each other. Similarities are calculated with a distance measuring technique such as Euclidean, Manhattan or Minkowski distance metric when predicting the values of new data points [15].

Support Vector Regression (SVR) is one of the common versions of the Support Vector Machine algorithm (SVM) which is used to predict a continuous variable. This version of SVM for regression proposed in [16]. SVR tries to fulfill generalized performance and reduce the error between the predicted value and the actual value [17].

Random Forest (RF) is a widely used learning algorithm that is used for both classification and regression problems. RF is an ensemble technique that constructs multiple decision trees at training time by selecting a set of bootstrap samples and random feature subsets from the dataset. The final result is obtained based on the outputs of decision trees. The logic behind the decision making procedure is to output the majority voting of the classes for classification and mean prediction of the individual trees for regression problems [18].

Extra Trees (ET) is named for Extremely Randomized Trees. This algorithm is an ensemble learning technique based on decision trees. It builds multiple trees and splits nodes by randomizing subsets of features. The algorithm is very similar to Random Forest, but in Extra Trees, the same training set is used to construct all trees, and nodes are split on random splits, instead of best splits [19].

Gradient Boosting (GB) is an ensemble method and in this method, the predictors are made sequentially, not independently. Gradient Boosting is one of the boosting algorithms. This algorithm is a ML technique for both classification and regression problems. It generates a strong predictor in the form of an ensemble of weak predictors. Gradient Boosting combines the outputs of several weak learners whose performance is better than random chance in order to produce more successful ones [20].

Ridge Regression (RR) algorithm is a variation of a linear regression which utilizes a shrinkage estimator. Shrinkage is applied to reduce the effects of sampling variation. In the method, a ridge estimator is used as a shrinkage estimator so as to enhance the least-squares estimate when multi-collinearity exists. The algorithm reduces the model complexity and multi-collinearity by shrinking the coefficients. This algorithm implements a regularized form of least-squares regression and avoids over-fitting [21].

Least Absolute Shrinkage and Selection Operator (LASSO) is a machine learning method that uses shrinkage. LASSO performs both regularization and variable selection, so it is generally used when there are a large number of features. The aim is to improve the accuracy of prediction and to reduce the over-fitting. Like Ridge regression, LASSO regression method is also suitable for the models with high levels of multicollinearity. LASSO regression performs L1 regularization and the aim is to get the subset of predictors that decrease the error rate of prediction for a target attribute [22].

Decision Tree (DT) is a commonly used supervised learning method that constructs models in the form of a tree structure. This algorithm builds a predictive model by repeatedly splitting the records at each node to reduce entropy in the resulting branches. The algorithm is easy to understand, visualize and interpret. In addition, it can deal with both binary and multi-class problems [23].

Convolutional Neural Network (CNN) is one of the deep learning methods which consists of a sequence of one or more convolutional layers and ends with some fully-connected layer. Each layer gets the values from the previous layer and converts them to information then passes to the next layer for more processing and generalization. In the first layers, feature extraction related processes are done automatically, hence the constructed model has ability to learn directly from image data, video, text, and sound by eliminating the necessity for manual feature extraction. CNNs are very popular in image and video processing, finding patterns to recognize objects, scenes, faces, and natural language processing [24].

In addition to supervised ML algorithms mentioned above, an unsupervised ML algorithm was applied as a data preprocessing step, before applying any regression algorithm. The K-Means algorithm was used in order to divide the dataset into subgroups where each data point belongs to only one group.

K-Means is a clustering algorithm that partitions the dataset into *k* clusters. In each cluster, a collection of data points grouped together depending on their similarities.

In our approach, after applying the clustering method, each cluster is labeled as a new categorical value, and so the clusters are converted into categorical features. In the next step, one-hot encoding is applied to represent these categorical features as binary vectors. Clustering and one-hot encoding steps are followed to enlarge the dataset in the feature generation process. To sum up, in the proposed approach, the first step is to apply a clustering algorithm in order to group data into meaningful clusters. After that, these clusters are accepted as new features and added to the dataset to enlarge it. In the second step, a regression analysis is performed for prediction.

4. Experimental Studies

In the following subsections, implementation details, dataset description, experimental results and evaluation of these results are explained.

4.1. Implementation details

In this study, the proposed approach was implemented in Python with Scikit-learn and Keras libraries in the Google Colab platform by building different learning models by using regression algorithms in order to predict the power generation of wind turbines. The applied algorithms are KNN, SVR, RF, ET, GB, RR, LASSO, DT, and CNN. The dataset used in this work includes some categorical and numeric values. Since some machine learning algorithms work better with encoded inputs, in this study, the one-hot encoding method was used. *One-hot encoding* is a process of converting categorical variables into a binary vector. In this method, each category value is converted into a new column and for each column 0 or 1 is assigned to this column. In other words, this is a representation of categorical variables as binary vectors.

For each applied algorithms, parameters are tuned to get more accurate prediction results. The applied parameters for each algorithm are listed as follows:

- For KNN, the number of neighbors parameter is set to 5 (n_neighbors=5), weight function used in prediction is uniform (weights='uniform'), algorithm utilized to calculate the nearest neighbors is auto (algorithm='auto'), leaf size is 30 (leaf_size=30), power parameter for the Minkowski metric is 2 (p=2), the distance metric to use for the tree is Minkowski (metric='minkowski'), additional keyword arguments for the metric function is none (metric_params=None).
- For SVR, kernel type is defined as RBF (kernel='rbf'), the degree of the polynomial kernel function is set to its default value 3 (degree=3), independent term in kernel function is kept as its default value 0.0 (coef0=0.0), use of shrinking heuristic is set to its default value true (shrinking=True), the size of the kernel cache in MB is specified as 200 (cache_size=200).
- For Random Forest, the number of trees is configured as 100 (n_estimators=100), the function used to evaluate the quality of a split is set to mean absolute error (criterion='mae'), the maximum depth of the tree is assigned as 30 (max_depth=30), the minimum number of examples required to divide a node is set to 5 (min_samples_split=5), the minimum number of examples required to be at a leaf node is specified as 1 (min_samples_leaf=1), the minimum weighted fraction required to be at a leaf node is 0 (min_weight_fraction_leaf =0.0), the number of features to be tested when determining the best split is configured as sqrt (max_features='sqrt'), maximum leaf nodes is kept as its default value none (max_leaf_nodes=None), minimum impurity decrease is also kept as its default value 0 (min_impurity_decrease=0.0), the use of bootstrap samples when building trees is set to false (bootstrap=False), the use of out-of-bag samples is set to its default value false (oob_score=False).
- For Extra Trees, the number of trees is specified as 100 (n_estimators = 100), the function to evaluate the quality of a split is set to mean squared error (criterion='mse'), the maximum depth of the tree is none (max_depth= None), the minimum number of examples required to divide a node is assigned as 2 (min_samples_split=2), the minimum number of examples required to be at a leaf node is 1 (min_samples_leaf=1), the minimum weighted fraction required to be at a leaf node is 0 (min_weight_fraction_leaf =0.0), the number of features to be tested when determining the best split is set to log2 (max_features='log2'), maximum leaf nodes is kept as its default value none (max_leaf_nodes=None), minimum impurity decrease is kept as its default value 0 (min_impurity_decrease=0.0), the use of bootstrap samples when building trees is false (bootstrap=False), the use of out-of-bag samples is set to its default value false (oob_score=False).
- For Gradient Boosting, loss function to be optimized is kept as its default value least squares regression (loss='ls'), learning rate is set to 0.5 (learning_rate=0.5), the number of

boosting stages to perform is specified as 10 (n_estimators=10), the fraction of examples to be utilized for training the base learners is set to its default value 1 (subsample=1.0), the function to measure the quality of a split is mean squared error (criterion='mse'), the minimum number of examples required to divide a node is 2 (min_samples_split=2), the minimum number of examples required to be at a leaf node is 1 (min_samples_leaf=1), the minimum weighted fraction required to be at a leaf node is set to 0 (min_weight_fraction_leaf=0.0), the maximum depth of the individual regression estimators is selected as 10 (max_depth=10), minimum impurity decrease is kept as its default value 0 (min_impurity_decrease=0.0), an estimator item that is utilized to find the initial predictions is set to its default value none (init=None), the number of features to be tested when determining the best split is set to its default value none (max_features=None), the quantile loss function and the alpha-quantile of the huber loss function is 0.9 (alpha=0.9), maximum leaf nodes is set to its default value none (max_leaf_nodes=None), the tolerance for the early stopping is 0.0001 (tol=0.0001).

- For Ridge Regression, the alpha value is 1.0 (alpha=1.0), fit intercept parameter is defined as true so as to calculate the intercept for this model (fit_intercept=True), the normalize parameter is kept as its default value false (normalize=False), the maximum number of iterations for conjugate gradient solver is assigned to its default value none (max_iter=None), the precision of the solution is set to 0.001 (tol=0.001), the solver to be used in the computational routines is configured as singular value decomposition (solver='svd').
- For LASSO, the alpha value is 1.0 (alpha=1.0), fit intercept parameter is set to true so as to calculate the intercept for this model (fit_intercept=True), the normalize parameter is kept as its default value false (normalize=False), use of a precomputed Gram matrix to speed up calculation parameter is set to its default value false (precompute=False), the maximum number of iterations is assigned as 1000 (max_iter=1000), the tolerance for the optimization is 0.0001 (tol=0.0001), the positive parameter is set to false not to force the coefficients to be positive (positive=False), selection parameter is set to cyclic (selection='cyclic').
- For Decision Tree, the function to evaluate the predictive quality of a split is selected as the mean squared error (criterion='mse'), the strategy utilized to select the split at each node is kept as its default value best (splitter="best"), the maximum depth of the decision tree is 10 (max_depth=10), the minimum number of examples required to divide a node is set to its default value 2 (min_samples_split=2), the minimum number of examples required to be at a leaf node is 1 (min_samples_leaf=1), the minimum weighted fraction required to be at a leaf node is 0 (min_weight_fraction_leaf=0.0), the number of features to be tested when determining the best split is kept as its default value none (max_features=None), maximum leaf nodes is set to its default value none (max_leaf_nodes=None), minimum impurity decrease is set to its default value 0 (min_impurity_decrease=0.0).
- For CNN, activation parameter is picked as relu (activation='relu'), loss is assigned as mean squared error (loss='mse'), epoch is specified as 50 (epochs = 50) and mean absolute error and mean squared error are configured as metrics parameter (metrics=['mae', 'mse']).

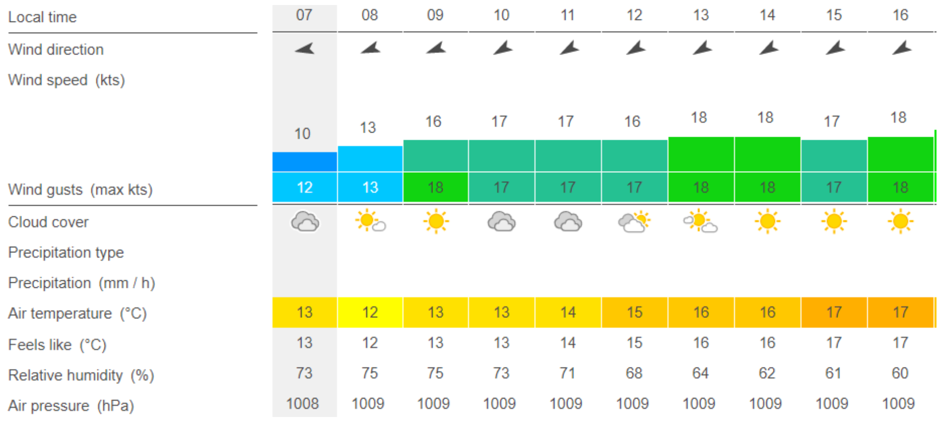
When a predictive model was built by a machine learning algorithm, the dataset was split into training and test sets with 80% and 20% respectively. Finally, results were gathered and the successes of algorithms were discussed based on two evaluation measures: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The obtained results will be explained in the next section.

4.2. Dataset description

In this study, a real-world dataset was used to perform experiments. The dataset contains power generation values of a wind farm located in Turkey and weather condition reports. The data covers a period of two years from 2017 to 2018.

Weather information was obtained from the web site [25], named as wind finder, which presents wind forecasts, wind maps, and live weather reports. Figure 2 shows a sample screenshot from the weather information system. The following weather information was recorded: wind speed (kts), wind direction, wind gust (max kts), cloud cover, precipitation (mm / h), precipitation type, air temperature (°C), the temperature feels like (°C), air pressure (hPa) and relative humidity (%).

Wind direction is measured in degrees and it is represented with arrow icons based on its degree as seen in Figure 2. In the dataset, we categorized the direction values into eight types (north (N), south (S), east (E), west (W), northeast (NE), northwest (NE), southeast (SE), and southwest (SW)) based on their degrees.



Local time	07	08	09	10	11	12	13	14	15	16
Wind direction	←	↗	↗	↗	↗	↗	↗	↗	↗	↗
Wind speed (kts)	10	13	16	17	17	16	18	18	17	18
Wind gusts (max kts)	12	13	18	17	17	17	18	18	17	18
Cloud cover	☁	⛅	☀	☁	☁	⛅	⛅	☀	☀	☀
Precipitation type										
Precipitation (mm / h)										
Air temperature (°C)	13	12	13	13	14	15	16	16	17	17
Feels like (°C)	13	12	13	13	14	15	16	16	17	17
Relative humidity (%)	73	75	75	73	71	68	64	62	61	60
Air pressure (hPa)	1008	1009	1009	1009	1009	1009	1009	1009	1009	1009

Figure 2. A sample screenshot from the weather information system

Table 1. A sample subset data after applying the feature generation process

Wind Speed	Wind Gust	Air Temperature	Air Temperature Feels	Relative Humidity	Air Pressure	Wind Direction	Cloud Cover	Label
9	11	3	-3	80	1021	N	NC	cluster_label1
9	10	2	-3	83	1021	NE	S	cluster_label3
9	11	2	-4	85	1021	SW	PC	cluster_label2
3	6	14	14	86	1020	SE	C	cluster_label8
5	7	13	13	87	1019	S	SS	cluster_label5
6	8	13	13	90	1020	E	SC	cluster_label4
7	8	12	12	90	1020	W	NC	cluster_label7
11	16	7	2	72	1019	NW	NSC	cluster_label6
11	14	8	3	69	1019	S	NPC	cluster_label6
10	13	8	4	67	1019	E	C	cluster_label5

As shown in Figure 3, cloud cover data is represented with different icons and was fallen into nine categories, including night open (NO), sunny (S), partly cloudy (PC), cloudy (C), slightly sunny (SS), slightly cloudy (SC), night cloudy (NC), night slightly cloudy (NSC), and night partly cloudy (NPC).

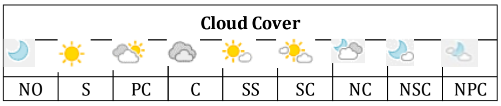


Figure 3. Cloud cover data categories

As shown in Figure 4, precipitation type values were labeled from A to H based on their types.

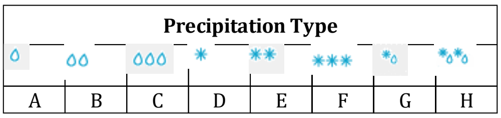


Figure 4. Precipitation type data categories

For the data collection and preparation task, we developed a data generator tool. We created the features of weather forecasting information for the hourly time horizon. Table 1 shows a sample subset of the dataset after the feature generation process. Table 2 shows a sample view of the weather data. A total of 17,256 records were stored in the dataset for the time period of January 1, 2017 to December 31, 2018. In the data preprocessing step, precipitation information was dropped due to many missing values. Since *wind direction* and *cloud cover* are categorical features, the one-hot encoding method was applied to convert them into binary vectors.

As shown in Table 1, after applying the K-Means clustering algorithm, the last column was added to the dataset, which contains the cluster labels, such as 'cluster_label_0', 'cluster_label_1' and so on. The optimal cluster number was investigated and then was set to 9 for the K-Means algorithm, so the dataset was partitioned into 9 non-overlapping groups. All weather forecasting data is merged with the generated electricity amount to create the final version of the dataset. The resulting dataset contains hourly wind power generation values and corresponding weather forecasting information at each hour. The target feature to predict is the "wind power production" column.

4.3. Experimental results

In this study, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used to evaluate the performances of the models for predicting wind power generation. MAE is a model evaluation metric that is used for continuous variables. The prediction error is calculated with the difference between the actual value and the predicted value for an instance. It measures the average magnitude of all absolute errors and calculated by the following formula:

Table 2. A sample view of the weather data

Local time	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h	20h	21h	22h	23h
Wind direction	NE	E	E	E	E	E	E	E	E	E	E	NE	E	E	E	SE	SE	SE	SE	SE	SE	SE	SE	SE
Wind speed (kts)	8	8	8	9	8	8	8	8	7	8	8	7	5	5	4	4	4	5	6	7	8	7	5	5
Wind gust(max kts)	12	12	12	11	11	10	10	9	8	8	8	7	6	5	5	7	7	10	11	12	13	13	10	10
Cloud cover	NO	NO	NSC	NC	NC	NPC	C	SC	C	SS	S	S	S	PC	PC	PC	SS	SS	SC	NC	NO	NO	NO	NC
Precipitation type															A	B	A	A			A	A		
Precipitation (mm / h)															1.6	3.1	1.4	0.4			0.3	0.1		
Air temperature (°C)	11	11	11	11	11	11	11	11	12	12	13	14	15	17	16	16	16	16	16	16	16	16	16	15
Feels like (°C)	11	11	11	11	11	11	11	11	12	12	13	14	15	17	16	16	16	16	16	16	16	16	16	15
Relative humidity (%)	72	73	73	74	74	74	75	74	74	73	70	67	69	69	80	84	87	86	84	84	88	88	90	90
Air pressure (hPa)	1017	1018	1018	1018	1018	1018	1017	1017	1017	1018	1017	1017	1017	1015	1015	1014	1013	1013	1012	1012	1012	1011	1011	1011

$$\text{MAE} = \frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}} |P_i - O_i| \qquad (1)$$

where *n* is the number of samples, P_i is the predicted value, and O_i is the observed value. Unlike MAE, in RMSE, the difference between prediction and actual observation is squared and then averaged. In the end, the square root of the average is calculated. These two evaluation metrics can take values from 0 to infinity (∞). RMSE assigns high values to large errors, because of squaring the errors before averaging them.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}} (P_i - O_i)^2} \qquad (2)$$

In the experimental studies, the real-world data of wind power generation that covers a period of two years from 2017 to 2018 was used to perform experiments. In order to observe the effect of the proposed approach, every model is trained with and without the feature generation process. Table 3 compares the MAE results obtained when the feature generation process was applied and not applied. The results indicate that the proposed approach, which includes the feature generation process, provides an improvement in prediction accuracy for almost all algorithms. For example; when using the decision tree technique, the prediction model built by the proposed approach has a lower error value (0.8260) compared to the model constructed without the feature generation process (0.8329). Applying K-Means clustering and one-hot encoding steps in the feature generation process may not be suitable for only SVR and CNN algorithms, because there is a slight increase in their MAE values. The MAE value of LASSO remains the same. It is observed from Table 3 that among the algorithms performed in this study, Random Forest has superiority in terms of MAE measure. After Random Forest, the most successful algorithms are Extra Trees, Support Vector Regression, and Gradient Boosting algorithms, respectively.

Table 3. Comparison of MAE results with and without feature generation process

Algorithm	Without feature generation process (MAE)	With feature generation process (MAE) (Proposed Approach)
KNN	0.7720	**0.7719**
SVR	**0.7283**	0.7289
Random Forest	0.7049	**0.7015**
Extra Trees	0.7290	**0.7269**
Gradient Boosting	0.7742	**0.7671**
Ridge Regression	0.9363	**0.9336**
LASSO	1.0632	**1.0632**
Decision Tree	0.8329	**0.8260**
CNN	**0.9433**	0.9660

Table 4 presents the effect of parameter tuning on the machine learning methods for the proposed approach. It is seen that the performance of the Random Forest, Extra Trees, Gradient Boosting, Ridge Regression, and Decision Tree algorithms improved when the optimized parameters were applied. The best performance improvement was achieved by the Gradient Boosting method.

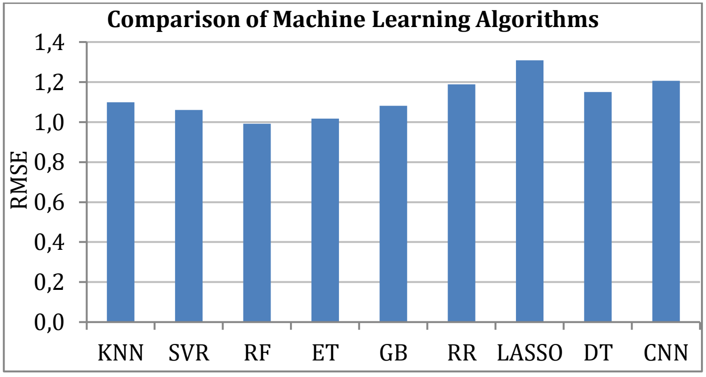


Figure 5. Comparison of machine learning algorithms for wind power generation prediction in terms of RMSE values

Table 4. Comparison of the MAE results with the default and optimized parameters for the proposed approach

Algorithm	MAE (Default Parameters)	MAE(Optimized Parameters)
KNN	0.7719	0.7719
SVR	0.7289	0.7289
Random Forest	0.7449	0.7015
Extra Trees	0.7702	0.7269
Gradient Boosting	1.1158	0.7671
Ridge Regression	0.9578	0.9336
LASSO	1.0632	1.0632
Decision Tree	0.9153	0.8260
CNN	0.9660	0.9660

Figure 5 shows the RMSE results of all applied algorithms on the dataset when the feature generation process is implemented. According to the results, it is clearly seen that Random Forest is the best algorithm with the lowest RMSE value (0.991). Algorithms after Random Forest do not change in the ranking of the most successful ones. They are still Extra Trees, SVR and Gradient Boosting algorithms, respectively. The worst performing algorithm is the LASSO algorithm with 1.3093 RMSE value.

5. Conclusion and Future Work

Renewable energy production prediction is crucial to meet the increasing energy demand. Renewable energy becomes increasingly popular in the world and wind energy is one of the most significant resources of renewable energy. Energy needs can be predicted based on historical data and it is very important for improving energy-saving strategies to plan and manage energy transmission, generation, and distribution. Especially, wind energy has attracted worldwide attention to the generation of electricity. Machine learning techniques are presented to provide increasingly accurate predictions in the energy sector to know the consumption and production amount and habits. This paper has investigated the success of different machine learning algorithms of energy generation prediction. The goal of this study is to solve a real-life problem in the renewable energy sector by accurately estimating the amount of wind energy production per hour in the next 24 hours. For this reason, this study investigates and compares many machine learning algorithms for wind power generation forecasting, including KNN, SVR, RF, ET, GB, RR, LASSO, DT, and CNN. This study proposes an approach that consists of two main stages. In the first stage, a clustering method is applied for feature generation. In the second stage, regression analysis is performed for prediction by using hourly electricity generation and predicted weather values, such as wind speed, cloud cover, wind direction, air temperature, and pressure. The experimental studies demonstrate the efficiency of the proposed approach in wind energy power prediction. The experiments were performed to estimate the amount of energy produced by turbines for a wind power plant located in Turkey. The results show that Random Forest is the best algorithm to be used for prediction with the 0.7015 MAE value. After Random Forest, the most successful algorithms are Extra Trees, SVR and Gradient Boosting algorithms, respectively. The worst performing algorithm is the LASSO algorithm with 1.0632 MAE value. It is concluded from the experimental results that feature generation with the clustering technique can be used to improve the wind power generation prediction accuracy.

As for future work, the weather data can be enriched with information obtained from wind turbines. Because wind turbines are capable of measuring and storing some additional environmental data during generating electricity from wind energy and this information can be useful for the prediction task. The number of features in the dataset can be increased by adding new features, such as turbine failures, maintenance periods, and maintenance activities. In this way, it may be possible to produce better prediction results. In addition, as future work, a new software application can be developed in order to use the approach proposed in this study in the industry. With this application, data collection and prediction processes can be automated.

Acknowledgment

We would firstly like to acknowledge Türker Murat for his support in providing and processing of wind power generation data as well as Bıçakcılar Çandarlı Elektrik Üretim A.Ş. for the sharing of wind power generation data used in the development of this study. We would also like to thank Gülşah Murat for introducing us to this company and for her help in acquiring data used in this work.

References

[1] Dolara, A., Gandelli, A., Grimaccia, F., Leva, S., Mussetta, M. 2017. Weather-based Machine Learning Technique for Day-Ahead Wind Power Forecasting. IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA), 5-8 November, San Diego, CA, USA, 206-209. DOI: 10.1109/ICRERA.2017.8191267

[2] Zhang, J., Jiang, X., Chen, X., Li, X., Guo, D., Cui, L. 2019. Wind Power Generation Prediction Based on LSTM. 4th International Conference on Mathematics and Artificial Intelligence, 12-15 April, Chengdu, China, 85-89. DOI: 10.1145/3325730.3325735

[3] Zhang, J., Yan, J., Infield, D., Yongqian, L., Lien, F. 2019. Short-term Forecasting and Uncertainty Analysis of Wind Turbine Power Based on Long Short-term Memory Network and Gaussian Mixture Model, Applied Energy, Volume. 241, p. 229-244. DOI: 10.1016/j.apenergy.2019.03.044

[4] Cali, U., Sharma, V. 2019. Short-term Wind Power Forecasting Using Long-short Term Memory Based Recurrent Neural Network Model and Variable Selection, International Journal of Smart Grid and Clean Energy, Volume. 8, p. 103-110. DOI: 10.12720/sgce.8.2.103-110

[5] Li, L., Zhao, X., Tseng, M., Tan, R. 2019. Short-term Wind Power Forecasting Based on Support Vector Machine with Improved Dragonfly Algorithm, Journal of Cleaner Production, Volume. 242: 118447. DOI: 10.1016/j.jclepro.2019.118447

[6] Okumuş, İ., Dinler, A. 2016. Current Status of Wind Energy Forecasting and a Hybrid Method for Hourly Predictions, Energy Conversion and Management, Volume. 123, p. 362-371. DOI: 10.1016/j.enconman.2016.06.053

[7] Hong, Y., Rioflorido, C.L.P. 2019. A Hybrid Deep Learning-based Neural Network for 24-h Ahead Wind Power Forecasting, Applied Energy, Volume. 250, p. 530-539. DOI: 10.1016/j.apenergy.2019.05.044

[8] Ma, Y., Zhai, M. 2019. A Dual-Step Integrated Machine Learning Model for 24h-Ahead Wind Energy Generation Prediction Based on Actual Measurement Data and Environmental Factors, Applied Sciences, Volume. 9, p. 2125. DOI: 10.3390/app9102125

[9] Salcedo-Sanz, S., Cornejo-Bueno, L., Prieto, L., Paredes, D., García-Herrera, R. 2018. Feature Selection in Machine Learning Prediction Systems for Renewable Energy Applications, Renewable and Sustainable Energy Reviews, Volume. 90, p. 728-741. DOI: 10.1016/j.rser.2018.04.008

[10] Zhang, L., Wang, K., Lin, W., Geng, T., Lei, Z., Wang, Z. 2019. Wind Power Prediction Based On Improved Genetic Algorithm and Support Vector Machine, IOP Conference Series Earth and Environmental Science, Volume. 252:032052. DOI: 10.1088/1755-1315/252/3/032052

[11] Demolli, H., Dokuz, A.S., Ecemiş, A., Gokcek, M. 2019. Wind Power Forecasting Based on Daily Wind Speed Data Using Machine Learning Algorithms, Energy Conversion and Management, Volume. 198: 111823. DOI: 10.1016/j.enconman.2019.111823

[12] Kramer, O., Gieseke, F., Satzger, B. 2012. Wind Energy Prediction and Monitoring with Neural Computation, Neurocomputing, Volume. 109, p. 84-93. DOI: 10.1016/j.neucom.2012.07.029

[13] Marvuglia, A., Messineo, A. 2012. Monitoring of Wind Farms' Power Curves Using Machine Learning Techniques, Applied Energy, Volume. 98, p. 574-583. DOI: 10.1016/j.apenergy.2012.04.037

[14] Wang, H., Lei, Z., Zhang, X., Zhou, B., Peng, J. 2019. A Review of Deep Learning for Renewable Energy Forecasting, Energy Conversion and Management, Volume. 198:111799. DOI: 10.1016/j.enconman.2019.111799

[15] Zhang, Y., Cao, G., Wang, B., Li, X. 2019. A Novel Ensemble Method for K-Nearest Neighbor, Pattern Recognition, Volume. 85, p. 13-25. DOI: 10.1016/j.patcog.2018.08.003

[16] Vapnik, V., Golowich, S.E., Smola, A. 1997. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. 9th International Conference on Neural Information Processing Systems, 2-5 December, Denver, CO, USA, 281-287.

[17] Basak, D., Pal S., Patranabis, D.C. 2007. Support Vector Regression, Neural Information Processing Letters and Reviews, Volume. 11(10), p. 203-224.

[18] Breiman, L. 2001. Random Forests, Machine Learning, Volume. 45, p. 5-32. DOI: 10.1023/A:1010950718922

[19] Geurts, P., Ernst, D., Wehenkel, L. 2006. Extremely Randomized Trees, Machine Learning, Volume. 63, p. 3-42. DOI: 10.1007/s10994-006-6226-1

[20] Zhang, C., Zhang, Y., Shi, X., Almpanidis, G., Fan, G., Shen, X. 2019. On Incremental Learning for Gradient Boosting Decision Trees, Neural Processing Letters, Volume. 50, Issue. 1, p. 957-987. DOI: 10.1007/s11063-019-09999-3

[21] Ohishi, M., Yanagihara, H., Fujikoshi, Y. 2020. A Fast Algorithm for Optimizing Ridge Parameters in a Generalized Ridge Regression by Minimizing a Model Selection Criterion, Journal of Statistical Planning and Inference, Volume. 204, p. 187-205. DOI: 10.1016/j.jspi.2019.04.010

[22] Kim, Y., Hao, J., Mallavarapu, T., Park, J., Kang, M. 2019. Hi-LASSO: High-Dimensional LASSO, IEEE Access, Volume. 7, p. 44562-44573. DOI: 10.1109/ACCESS.2019.2909071

[23] Trabelsi, A., Elouedi, Z., Lefevre, E. 2019. Decision Tree Classifiers for Evidential Attribute Values and Class Labels, Fuzzy Sets and Systems, Volume. 366, p. 46-62. DOI: 10.1016/j.fss.2018.11.006

[24] Patel, S. 2020. A Comprehensive Analysis of Convolutional Neural Network Models, International Journal of Advanced Science and Technology, Volume. 29, Issue. 4, p. 771-777.

[25] Windfinder. https://www.windfinder.com (Access Date: 09.03.2020