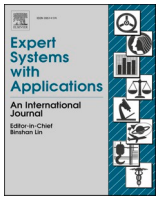
Expert Systems With Applications 237 (2024) 121464

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Data-driven interpretable ensemble learning methods for the prediction of wind turbine power incorporating SHAP analysis

Celal Cakiroglu [a,*], Sercan Demir [b], Mehmet Hakan Ozdemir [c], Batin Latif Aylak [d], Gencay Sariisik [b], Laith Abualigah [e,f,g,h,i,j,k]

[a] Department of Civil Engineering, Turkish-German University, Istanbul 34820, Turkey
[b] Harran University, Department of Industrial Engineering, Sanliurfa 63000, Turkey
[c] Department of Business Administration, Turkish-German University, Istanbul 34820, Turkey
[d] Department of Industrial Engineering, Turkish-German University, Istanbul 34820, Turkey
[e] Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al al-Bayt University, Mafraq 25113, Jordan
[f] Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon
[g] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
[h] MEU Research Unit, Middle East University, Amman 11831, Jordan
[i] Applied science research center, Applied science private university, Amman 11931, Jordan
[j] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia
[k] School of Engineering and Technology, Sunway University Malaysia, Petaling Jaya 27500, Malaysia

ARTICLE INFO

Keywords:
Renewable energy
Wind power
Machine learning
Predictive modeling

ABSTRACT

Wind energy increasingly attracts investment from many countries as a clean and renewable energy source. Since wind energy investment cost is high, the efficiency of a potential wind power plant should be determined using wind power prediction models and wind speed data before installation. Accurate wind power estimation is crucial to set up comprehensive strategies for wind power generation. This study estimated the power produced in a wind turbine using six different regression algorithms based on machine learning using temperature, humidity, pressure, air density, and wind speed data. The proposed estimation model was evaluated on the data received between 2011 and 2020 at station 17,112 in Çanakkale, Turkey. XGBoost, Random Forest, LightGBM, CatBoost, AdaBoost, and M5-Prime algorithms were used to create predictive models. Furthermore, model explanations were presented using the SHAP methodology. Among the regression algorithms evaluated according to the R^2 performance metric, the best performance was obtained from the XGBoost algorithm. Regarding computational speed, the LightGBM model emerged as the most efficient model. The wind speed was shown to be the input feature with the SHAP algorithm's most significant impact on the model predictions.

1. Introduction

Energy is vital for meeting basic human needs (Owusu & Asumadu-Sarkodie, 2016). In addition, the concept of energy from the past to the present is essential for developing the economic and social structure and for modern countries to reach their current state (McCauley et al., 2019). Since the Industrial Revolution, fossil fuels have been the primary energy source, and their use has reached approximately 10,000 million tons of oil equivalent. They provided about 80% of all primary energy worldwide by 2010 (Höök & Tang, 2013). The amount of fossil fuel is expected to meet global energy needs (EIA. International Energy Outlook 2019;, 2019). Fossil fuels are expected to account for 78% of total energy consumption worldwide by 2040 (Klemeš, Varbanov, Ocłoń, & Chin, 2019). However, controlling the rise in carbon emissions has always been a major issue. Although carbon emissions decreased with the COVID-19 pandemic in 2020, it was not enough to reverse the situation. At the same time, climate change has increased the importance of renewable energy sources.

Wind energy has become the fastest-growing renewable energy type (Han et al., 2020; Yang, Shahzadi, & Hussain, 2021). It is the least costly, most reliable, and environmentally friendly energy source. Therefore, wind energy is gradually preferred for daily energy use (Taghizadeh-

* Corresponding author.
E-mail addresses: cakiroglu@tau.edu.tr (C. Cakiroglu), sercanxdemir@gmail.com (S. Demir), hakan.ozdemir@tau.edu.tr (M. Hakan Ozdemir), batin.latif@tau.edu.tr (B. Latif Aylak), gsariisik@gmail.com (G. Sariisik), aligah.2020@gmail.com (L. Abualigah).

https://doi.org/10.1016/j.eswa.2023.121464
Received 3 May 2023; Received in revised form 3 September 2023; Accepted 3 September 2023
Available online 7 September 2023

Hesary & Yoshino, 2020). The use of wind energy, which is developing rapidly worldwide, is also rising in Turkey. While the installed power of wind power plants in Turkey was 1320 MW in 2010 and 1805 MW in 2011, it increased to 7369 MW in 2018 with an average increase of 750–1000 MW every year (ELEKTRİK PİYASASI YILLIK SEKTÖR RAPORU (EPİAŞ) (2019); TÜREB, 2019).

Despite the significant increase in wind capacity every year, it poses challenges. A challenging aspect of wind power forecasting is its reliance on the weather and its inherently unpredictable, random, and volatile nature. The uneven output of wind power leads to an imbalance between power generation and consumption, which impacts the costs of its use. Therefore, precise wind power forecasts are vital to energy management tasks, including generation, distribution, transmission, and planning (Aslam et al., 2021).

As a data-driven approach to solving real-world problems, ML techniques have proved their importance over the past decade (Cheng, Du, & Yao, 2022; Ghoddusi, Creamer, & Rafizadeh, 2019). The prediction of wind power has been improved using numerous ML algorithms, as previously stated. This study presents an explainable ML strategy to forecast wind power accurately. In addition to the accuracy and computational efficiency of different predictive models, the relationships between different input features and the model output are also presented. To the best of the authors' knowledge, this study fills a gap in the literature by applying explainable ensemble machine learning models to predict wind power. Besides, among the many input features used in this study, the most impactful variables on wind turbine power were revealed.

2. Literature review

The prediction of wind power has been studied in the literature in various ways using different analysis methods across different time horizons. Some investigations focused on short-term forecasting, while others investigated 24-hour ahead forecasting (Blazakis, Katsigiannis, & Stavrakakis, 2022; Gupta, Natarajan, & Berlin, 2022; Hu, Li, Zhang, & Fang, 2022; Huang, Li, Wei, & Zhang, 2022; Mahaseth et al., 2022; Nguyen & Phan, 2022; Piotrowski, Baczyński, Kopyt, & Gulczyński, 2022; Zhang & Li, 2022). While early studies have considered persistence and statistical methods for wind power forecasting, most recent studies have relied on machine learning (ML) algorithms. ML algorithms have advantages over generalized models because they can adapt to shifting patterns within datasets and generate models based on input data.

Considering the problems with instability and poor prediction accuracy in short-term wind power prediction, Huang et al. (2022) introduced a prediction model incorporating BiLSTM-CNN-WGAN-GP. Variational mode decomposition (VMD) was used to decompose the initial wind energy data into natural mode sequences. The bidirectional long short-term memory (BiLSTM) network was utilized as the generation model of the Wasserstein generative adversarial network with gradient penalty (WGAN-GP), while the convolutional neural network (CNN) was used as the discrimination model. Using the minimum-maximum game, the BiLSTM and CNN network models enhanced sample generation quality and forecasting accuracy. In their study, Mahaseth et al. (2022) used multiple ML techniques to estimate wind power based on two years of wind data collected from four locations. According to their findings, the K-nearest neighbor (KNN) algorithm was the most successful in predicting wind energy. Zhang and Li (2022) integrated the fast correlation-based feature (FCBF) selection algorithm with the VMD to study the characteristics of wind data and employed least squares support vector machines (LSSVMs) to predict wind power. Experimental results indicate that the designed model is significantly more accurate than a comparable model predicting outcomes.

An innovative hybrid model (NE-KC-GWO-SVM) was proposed by Hu et al. (2022) for predicting wind speed. Neo4j (NE) is used for data preparation, k-means clustering (KC) for data analysis, grey wolf optimizer (GWO) for kernel function optimization, and support vector machines (SVM) to enhance prediction accuracy. The experimental results revealed that, when compared to the other models, the suggested model has the highest accuracy. It is also found that the model predicts with good stability and has a satisfactory level of time complexity. Additionally, the suggested prediction system was more precise than similar predictions, minimizing the variations in power generation and improving the power grid's stability. Gupta et al. (2022) used a five-model ML approach to forecast short-term wind speed. The results show that large-margin distribution machine-based regression (LDMR) predicts outcomes more accurately than other models, but the extreme learning machine (ELM) is more computationally efficient.

Wind energy's efficacy also relies on forecasts for the next 24 h. With even a minor improvement in the quality of these forecasts, a safer and more economical system can be achieved. Blazakis et al. (2022) used three deep learning models to predict medium-term wind speed: multi-head CNNs, multi-channel CNNs, and encoder-decoder long short-term memory (LSTMs). Wind speed forecasting results were better for multi-channel CNN than other deep learning methods examined. Piotrowski et al. (2022) used ensemble, hybrid, and single methods to predict wind power for the next 24 h. ML solutions such as gradient-boosted trees, random forests (RF), multilayer perceptrons, LSTM, KNN regression (KNNR), and SV regression (SVR) were used. A comparison of all methods found that the ensemble method had the lowest normalized mean absolute error (nMAE). It has been proposed by Nguyen and Phan (2022) to construct a hybrid model combining decomposition with deep learning embedded with genetic algorithm (GA) optimization to predict wind speed 24 h ahead. Ensemble empirical mode was utilized to break down historical wind data into intrinsic mode functions (IMFs) after pre-processing. The IMF's were trained and tested using the CNN-Bidirectional LSTM model. The outcome reveals that the proposed strategy outperformed the alternative methods by a wide margin. Chang et al. (2021) discuss the role of global Industry 4.0 technology management in the wind turbine industry's growth, focusing on factors such as legal policies, patents, and companies. Huang, Liu, Wang, Pan, and Chang (2021) propose a cost-aware collaborative task execution (CACTE) scheme using a multi-agent deep deterministic policy gradient (MADDPG) based on cost-aware gradients.

Ti, Deng, and Zhang (2021) proved that artificial neural networks (ANNs) give reasonable wind power estimates compared to analytical models due to their flexibility and capacity to handle nonlinearity in processes. The LSTM and Gaussian mixture model (GMM) algorithms are used by Zhang, Jiang, Chen, Li, Guo, and Cui (2019a) to improve wind power forecasts' accuracy and understand how error distribution affects forecasts. Using deep neural networks as a forecasting method improves forecast accuracy and reduces operational costs, making wind and wave energy more competitive against other renewable sources.

There has also been an attempt to optimize input data to improve prediction accuracy (Li, Zhao, Tseng, & Tan, 2020; Zhang, Wang, Lin, Geng, Lei, & Wang, 2019b). An enhanced SVM built on a GA was investigated by Zhang et al. (2019b) to forecast wind energy. SVM models used radial basis functions (RBFs) as kernel functions, while GAs was used for parameter optimization. The proposed GA-SVM model performed better at making predictions than a model built with some standard parameters. Li et al. (2020) combined the SVM with an enhanced dragonfly algorithm for forecasting wind energy production. SVM input data were optimized using the enhanced dragonfly method. Data gathered from a French wind farm was used in their study. The results showed that the suggested model outperformed backpropagation neural networks and Gaussian process regression in prediction.

Much research has been done on ML strategies designed to improve prediction quality by combining the benefits of multiple models. Demolli, Dokuz, Ecemis, and Gokcek (2019) analyzed historical wind data to predict wind power with ML models, including RF, SVR, KNN, and the minor absolute shrinkage and selection operator (LASSO). They emphasize that ML models may be used on sites other than those where

the models were trained. These models, however, are static and do not account for historical data. It is significant to note that if time-series data have a moderate or temporal solid dependence, considering time-lagged values can increase forecast accuracy.

Using physical and data mining-based methodologies, Yan and Ouyang (2019) introduced a hybrid model for a wind farm that predicts wind power for the next three months. Maroufpoor, Sanikhani, Kisi, Deo, and Yaseen (2019) examined six ML algorithms that could predict wind speed based on meteorological data. Four ML models have been applied to wind power prediction by Buturache and Stancu (2021): artificial neural networks, SVR, regression trees, and RFs. According to the results, the SVR represents the best forecasting method when considering performance and training time using a single metric. Recent work by Deng et al. (2019) employed a bidirectional gated recurrent unit model to predict wind power. It was found that this approach can automatically model the relationship between wind speed, wind direction, and wind power.

Eyecioglu, Hangun, Kayisli, and Yesilbudak (2019) compared the performances of different ML algorithms in predicting wind turbine power generation, namely linear regression, k-nearest neighbor regression, and decision tree algorithms. They found that the k-nearest neighbor regression algorithm yields a lower coefficient of determination values, whereas decision regression algorithms have lower MAE values. Furthermore, they evaluated meteorological parameters such as wind speed, wind direction, barometric pressure, and air temperature according to their importance on the wind power parameter and showed that the most critical parameter was the wind speed. Alkesaiberi, Harrou, and Sun (2022) investigated enhanced ML models' forecasting performances using wind power time series data. They applied Bayesian optimization (BO) to optimally tune hyperparameters of the Gaussian process regression (GPR), SVR with different kernels, and ensemble learning (ES) models (i.e., Boosted trees and Bagged trees).

Zhang, Peng, and Nazir (2022b) investigate the effectiveness of the variational heteroscedastic Gaussian process regression (VHGPR) model in wind speed forecasting. The authors employ Marginalized Variational approximation and complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) to enhance forecasting performance. Zhang et al. (2022a) propose a hybrid deep learning model for wind speed prediction using convolutional neural networks, Bidirectional long short-term memory, improved sine cosine algorithm (ISCA), and time-varying filter-based empirical mode decomposition (TVFEMD).

Optimized GPR and ensemble models turned out to outperform other machine learning models. Moreover, they have incorporated dynamic information into their structure to improve the forecasting performance of these models further. In particular, lagged measurements have been added to capture time evolution in these models' designs. Also, more input variables, such as wind speed and direction, were used to improve wind forecast performance further. The results demonstrate the benefit of considering lagged data and input variables.

A group of researchers used artificial neural network and deep learning (DL) techniques on the estimation of the wind turbine power generation. Sun et al. (2020) introduced a power prediction model based on an artificial neural network that estimates the power generation of wind turbines for given wind speeds, wind directions, and yaw angles. The model considers the wake effect and minimizes the impact of the wake on the entire wind farm. Xu et al. (2021) presented a new concept named as quantile power curve and introduced a quantile loss function-based neural network algorithm. Moreover, the authors introduced a quantile power curve-based index that assesses the wind turbine power generation performance and uncertainty. Zhang, Yan, Infield, Liu, and Lien (2019c) introduced a long short-term memory (LSTM) model based on a deep learning network that predicts wind turbine power. The authors employ the Gaussian mixture model (GMM) to describe the error distribution for short-term wind turbine power prediction.

3. Methodology

3.1. Data preparation

The dataset comprises 3444 data samples from the period 2011–2020. All data was obtained from the General Directorate of Meteorology at station 17,112 for Çanakkale province in Turkey. Each one of these data samples contains measured values of temperature (°C), wind speed (m/s), humidity (%), and air pressure (hPa) on a given day for nine years. The average values of an entire day's measurements have been included in the dataset for each day. Considering the geographical regions of Turkey in terms of wind speeds, the annual average wind speed at 10 m altitude is 3.3 m/s in the Marmara region (Ucar & Balo, 2010), and due to low-speed wind turbines start spontaneously at speeds of 2–4 m/s (Howell, Qin, Edwards, & Durrani, 2010). For these reasons, in our study, a three-bladed vertical wind turbine with a wind speed of 2–4 m/s is considered in wind turbine power calculations.

The power to be obtained from the wind depends on air density (ρ), rotor swept area (A), Betz limit (C_p) and cube of wind speed (V^3) and is calculated using Eq. (1).

$$P_A = \frac{1}{2}\rho A C_p V^3 \tag{1}$$

The rotor swept area is calculated as the product of the maximum width of the rotor blades (D) and the maximum vertical height of the rotor blades (H) using Eq. (2),

$$A = DH \tag{2}$$

Wind turbines can start generating electricity at a certain wind speed and produce between cut-in (2–4 m/s) and cut-out (25–35 m/s) speeds. The Betz limit (C_p) is taken into account when calculating the power generation for wind turbines. With today's technology, it is impossible to convert all the kinetic energy of the air into electrical energy. For this reason, in the calculations, the Betz limit is taken as approximately 59.26 % in order to generate the highest amount of electrical energy from the kinetic energy of the air.

Air density (ρ) in the Equation developed by the International Committee for Weight and Measurement (CIPM) was calculated with the following variables: p_d is the partial dry air pressure (Pa), R_d is the specific gas constant for dry air 287.058 J/(kg K), T is the temperature (K), p_v is the water vapor pressure (Pa), R_v is the specific gas constant for water vapor, 461.495 J/(kg K), M_d is the molar mass of dry air 0.0289654 kg/mol, M_v is the molar mass of water vapor 0.018016 kg/mol, and R is the universal gas constant, 8.314 J/(K mol) (see Eqs. (3) and (4)).

$$\rho = \frac{p_d}{R_d T} + \frac{p_v}{R_v T} \tag{3}$$

$$\rho = \frac{p_d M_d + p_v M_v}{RT} \tag{4}$$

The water vapor pressure (p_v), saturated vapor pressure (p_{sat}), and relative humidity ($\varnothing$) calculations are made with the formulas in Eqs. (5) and(6).

$$p_v = \varnothing p_{sat} \tag{5}$$

$$p_{sat} = 6.1078 \cdot 10^{\frac{7.5T}{T+273.3}} \tag{6}$$

3.2. Machine learning methodologies

This section presents the process of developing six different machine learning predictive models based on a dataset that comprises 3444 data samples. All of this data was collected from measurements made in the Çanakkale region of Turkey from 2011 to 2020. During this period, days with missing data values have been discarded from the dataset.

Corresponding to each data sample, the turbine power has been calculated, which is the output feature in this study. The following sections present the statistical distributions of the input and output features, followed by a general description of the machine learning procedures used in this study.

Fig. 1 shows the correlation plot of all the features included in the dataset. The upper triangle of the correlation matrix in Fig. 1 shows the Pearson correlation coefficients between the input features. The font size in the upper triangle is proportional to the correlation magnitude. Fig. 1 displays the scale of each variable in a diagonal tile on one of the vertical and horizontal axes. In this diagram, the diagonal displays histograms of the distributions for all the features, while the lower triangle has bivariate scatter plots with regression lines. The diagonal in Fig. 1 shows the variable distributions with histograms. Fig. 1 shows the most excellent correlation between turbine power and wind speed with a Pearson correlation coefficient of 0.87. This is followed by the correlation between air density and air pressure (r = 0.68). The most remarkable inverse correlation is observed between temperature and air density with r = -0.98.

Fig. 2 shows each design feature's upper and lower bounds and the distribution of the features in different ranges. For each design feature, the entire range of values has been split into compartments shown with different colors, and the upper and lower bounds of each compartment are written on the boundaries above the horizontal bars. Each compartment's total number of data samples has been written into the corresponding positions inside the horizontal bars. In Fig. 2, the length of each compartment is proportional to the number of samples belonging to that compartment. The feature ranges have been split between their upper and lower bounds into subintervals according to the number of data samples in each subinterval in order to generate a visual description of their distributions. In case of input features that are not uniformly distributed, Fig. 2 visualizes the intervals in which most data

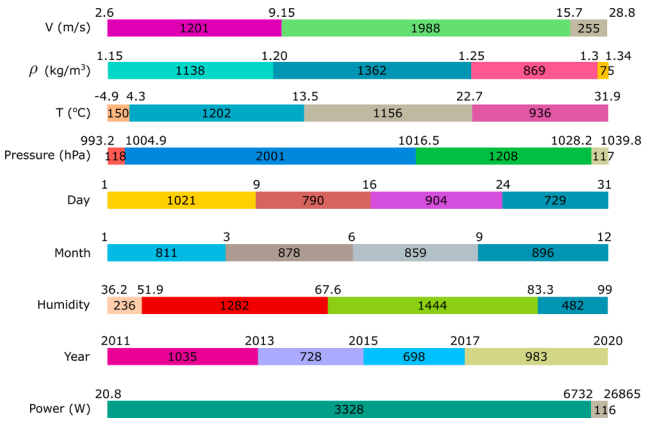


Fig. 2. Feature ranges in the dataset.

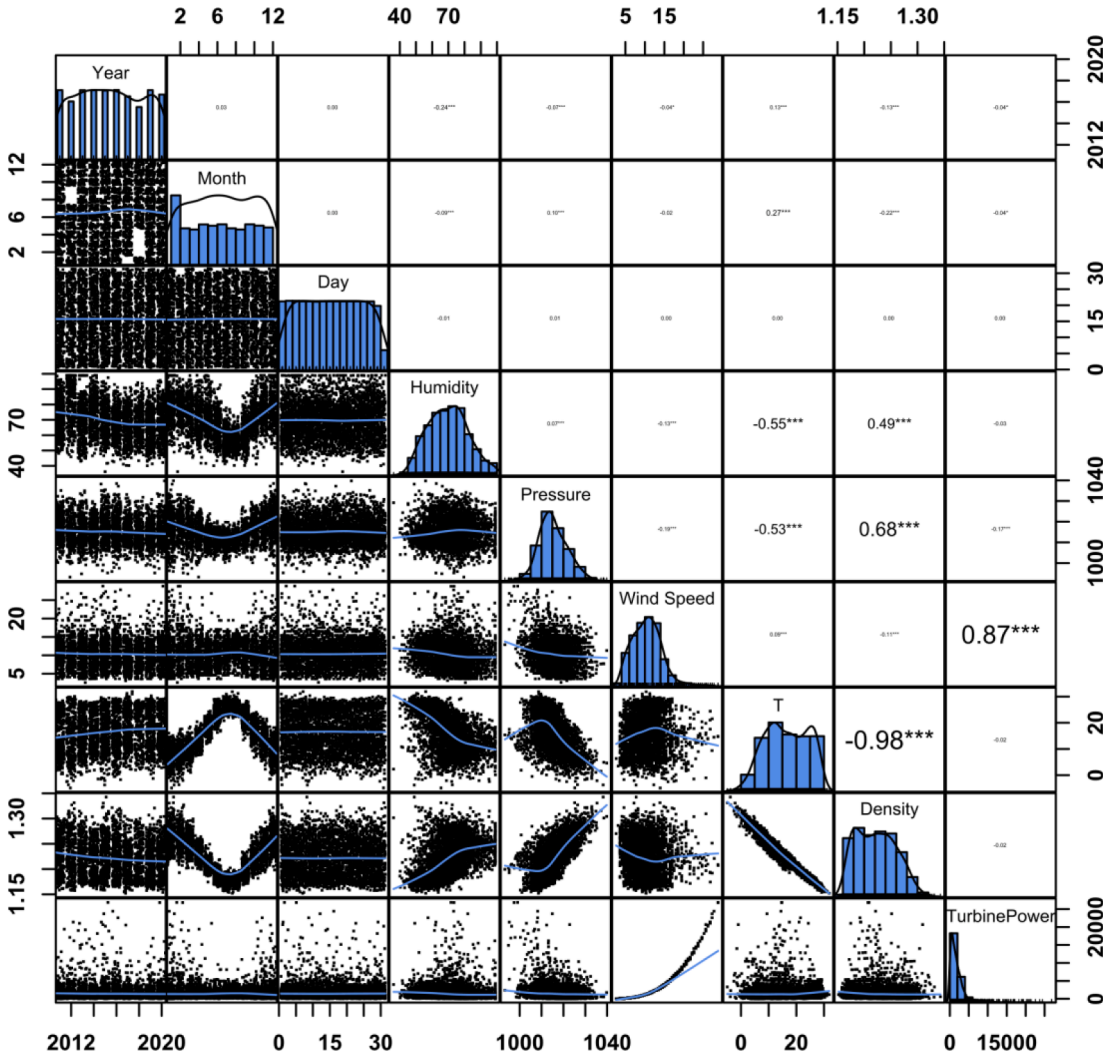


Fig. 1. Correlation plot of the input features.

samples are located. Fig. 2 shows that in a significant portion of the data samples (57.7%), the wind speed (V) ranges between 9.15 m/s and 15.7 m/s, and in 92.6% of the samples, the wind speed is below 15.7 m/s. According to Fig. 2, the temperature (T) stays above 4.3 °C 95.6% of the time; above this level, the temperature has a uniform distribution. A similar uniform distribution can also be observed for the air density values less than 1.3 kg/m^{3,} while in 97.8% of the cases, the air density is below 1.3 kg/m^3. In 79.2% of the days, the humidity was between 51.9% and 83.3%. Fig. 2 confirms the turbine power's skewed distribution, which can also be observed from the histogram in Fig. 1. Fig. 2 shows that in 96.6% of the samples, the turbine power was less than 6732 W.

Using the Shapley Additive exPlanations (SHAP) approach, the feature contributions to the model output have been further examined. In this study, the XGBoost, Random Forest, LightGBM, CatBoost, AdaBoost, and M5-Prime algorithms were used to create the predictive models. The theoretical foundation of these strategies is outlined in the sections that follow.

3.2.1. Machine learning algorithms

The development of strong predictors as an ensemble of weak decision trees is a characteristic of the tree-boosting techniques employed in this study. The eXtreme Gradient Boosting (XGBoost) algorithm belongs to this category and is one of the most often utilized algorithms. The capacity of the XGBoost algorithm to scale to billions of samples while maintaining excellent computing efficiency is one of its distinguishing characteristics. The XGBoost model's output can be described using Eq. (7) where $\widehat{y}_i$ is the predicted value for the data sample with index i, f_k represents a regression tree, K is the number of regression trees and x_i is the feature vector of the sample with index i. In Eq. (8), where T is the number of leaves and w_k is a vector of leaf weights, the regularized objective function $\mathscr{L}(\phi)$ is minimized to produce regression trees. The computation of the ideal leaf weights w_j^* using the loss function L is also shown in Eq. (9) where I_j is the collection of sample indices for the j-th leaf in the Equation (Bakouregui, Mohamed, Yahia, & Benmokrane, 2021; Chen & Guestrin, 2016).

$$\widehat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{7}$$

$$\mathscr{L}(\phi) = \sum_i L(y_i, \widehat{y}_i) + \sum_k \Omega(f_k) = \sum_i L(y_i, \widehat{y}_i) + \sum_k \gamma T + \frac{1}{2}\lambda \|w_k\|^2 \tag{8}$$

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, g_i = \frac{\partial L\left(y_i, \widehat{y}_i^{(t-1)}\right)}{\partial \widehat{y}_i^{(t-1)}}, h_i = \frac{\partial^2 L\left(y_i, \widehat{y}_i^{(t-1)}\right)}{\partial\left(\widehat{y}_i^{(t-1)}\right)^2} \tag{9}$$

The LightGBM and CatBoost algorithms are further variations of the gradient boosting methodology. The distinguishing feature of the LightGBM algorithm is the application of the gradient-based one-side sampling (GOSS), parallel learning, and exclusive feature bundling (EFB) method, which enable the LightGBM algorithm to process large datasets with high accuracy, training speed, and efficiency (Ke et al., 2017). The histogram-based gradient boosting feature of LightGBM improves the efficiency of the models in dealing with large datasets. LightGBM creates histograms for each continuous feature. The discretization of data into histogram bins reduces the number of potential segmentation points that need to be considered. The optimal segmentation points of the features are determined by traversing the histogram bins. Also, using histograms instead of continuous variables has favorably affects the memory needed to process large datasets. On the other hand, the CatBoost algorithm implements the ordered boosting method, increasing the overall model performance for datasets with categorical features. Additionally, CatBoost addresses the problem of prediction shift by utilizing ordered target statistics (Degtyarev & Naser, 2021; Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018).

The AdaBoost algorithm is based on consecutively focusing on parts of the training set under-fitted by the predictors of the previous iterations. This is achieved by assigning greater weights (w_i) to under-fitted data samples. The prediction of an Adaboost model can be described in Eq. (10) (Drucker, 1997).

$$\widehat{y}_i = f_t(x_i) = inf\left\{ y \in Y : \sum_{t:\widehat{y}_i^t \le y_i} log\left(\frac{1}{\beta_t}\right) \ge \frac{1}{2}\sum_t log\left(\frac{1}{\beta_t}\right) \right\} \tag{10}$$

$$\beta_t = \frac{\overline{L}}{1-\overline{L}}, \overline{L} = \sum_i L_i p_i, p_i = \frac{w_i}{\sum_i w_i} \tag{11}$$

In Eqs. (10) and (11), β_t denotes the confidence parameter for the predictor with index t and $\overline{L}$ is the average loss for the entire training set. Eqs. (10) and (11) summarize the process of ranking the output of a certain number of predictors for a given sample from largest to smallest and taking the prediction of the predictor that corresponds to the smallest t value, which satisfies the inequality in Eq. (10).

M5-Prime is an extended version of the M5 tree model which uses standard deviation reduction (SDR) as the node splitting criterion, which can be calculated as in Eq. (12) where S denotes the set of samples reaching a node and S_i denotes the subsets after node splitting (Behnood, Behnood, Gharehveran, & Alyamac, 2017).

$$SDR = sd(S) - \sum_i \frac{|S_i|}{|S|} sd(S_i) \tag{12}$$

The leaf values in M5-Prime are approximated by linear regression, and predictions are improved using a smoothing procedure. Further details of the M5-Prime algorithm can be found in (Wang & Witten, 1996).

The random forest models combine several individual decision trees' predictions to obtain a robust model. The single decision trees are trained on randomly chosen subsets of the whole training set using bagging and random feature selection techniques. The prediction of the random forest model is the average of the individual tree predictions, as shown in Eq. (13) where $\widehat{m}_j$ stands for the three decision models with index j and K as the total number of decision trees (Feng, Wang, Mangalathu, Hu, & Wu, 2021).

$$\widehat{m}(\boldsymbol{x}) = \frac{1}{K}\sum_{j=1}^{K} \widehat{m}_j(\boldsymbol{x}) \tag{13}$$

3.2.2. SHAP methodology

The SHAP algorithm is an effective tool for determining the effects of various input variables on model predictions. The SHAP technique uses simplified explanation models that give close results to the original predictive model to explain complex machine learning models. This procedure can be described by Eqs. (14) and (15) (Lundberg and Lee, 2017). In Eqs. (14) and (15), the simplified explanation model and the actual predictive model are denoted by m and f, respectively. The vector of input features is denoted by $\boldsymbol{x}$. This vector is related to $\boldsymbol{x}' \in \{0,1\}^N$ by a mapping function, and N denotes the number of input variables. Here, 1 indicates the inclusion of a feature in the simplified model, and 0 indicates the exclusion of a feature. F and S sets are the set of all features and a subset of features excluding the feature with index i, respectively. The vector of input feature values for the features in the set S is denoted with $\boldsymbol{x}_S$. The impact of a feature with index j is represented with ϕ_j. These values are computed based on the differences in the model output when a feature is included in the model or withheld from the model. A more detailed explanation of the SHAP procedure can be found in (Lundberg and Lee, 2017).

$$m(\boldsymbol{x}') = \phi_0 + \sum_{j=1}^{N} \phi_j x_j' \tag{14}$$

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{j\}}\left(\boldsymbol{x}_{S \cup \{j\}}\right) - f_S(\boldsymbol{x}_S) \right] \tag{15}$$

4. Results and discussion

This section shows the output of the machine learning models introduced in the previous section on the turbine power dataset. Comparisons between actual turbine power values and those estimated by machine learning algorithms are presented. The SHAP technique has been used to assess the machine learning model output and identify the features that most influence model predictions. The coefficient of determination, mean absolute error, and root mean squared error metrics have been used to evaluate the performances of each predictive model.

Fig. 3 compares the calculated and predicted turbine powers using six different predictive models. The model performances are listed in Table 1 using three metrics of accuracy. Fig. 3 shows the model predictions on the training and test sets in different colors and symbols. The entire dataset was split into a training and a test set in a 70% to 30% ratio. A coefficient of determination of $R^2 = 0.9999$ and $R^2 = 0.9994$ could be achieved on the training and test sets, respectively. The input features were date, humidity, pressure, wind speed, temperature, and air density. According to Table 1, the XGBoost algorithm could achieve the best accuracy on the training set concerning all three metrics. On the test set, the XGBoost, Random Forest, and M5-Prime algorithms could achieve equally high accuracy in terms of the R^2 score, whereas the Random Forest model achieved the best accuracy in terms of MAE, and M5-Prime performed best in terms of RMSE. In terms of computational speed, LightGBM was the fastest algorithm.

Fig. 4a compares the predicted and target turbine power levels on randomly selected training and test sets subsets, illustrating the degree

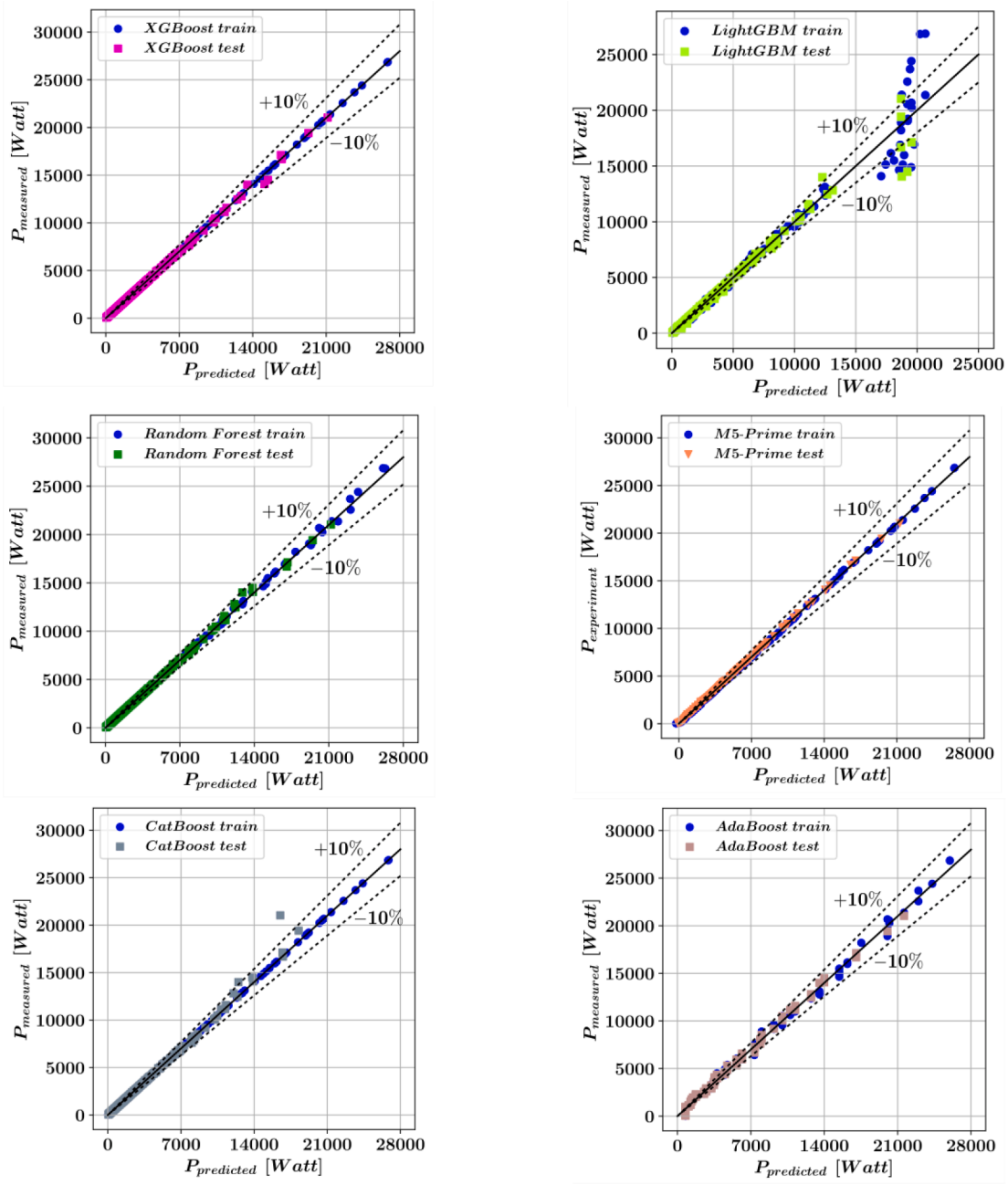


Fig. 3. Prediction of turbine power.

Table 1
Predictive model performances.

	R^2		MAE		RMSE		Duration [s]
	Train	Test	Train	Test	Train	Test	
XGBoost	0.9999	0.9994	1.517	11.51	2.206	54.45	0.32
LightGBM	0.9834	0.9866	38.94	34.17	323	254	0.10
Random Forest	0.9997	0.9994	5.517	8.561	39.82	53.45	1.26
M5-Prime	0.9995	0.9994	42.23	41.41	53.69	51.63	2.02
CatBoost	0.9999	0.9947	5.129	20.11	7.20	160	4.69
AdaBoost	0.9824	0.9787	266.9	256.9	331.9	320.2	0.23

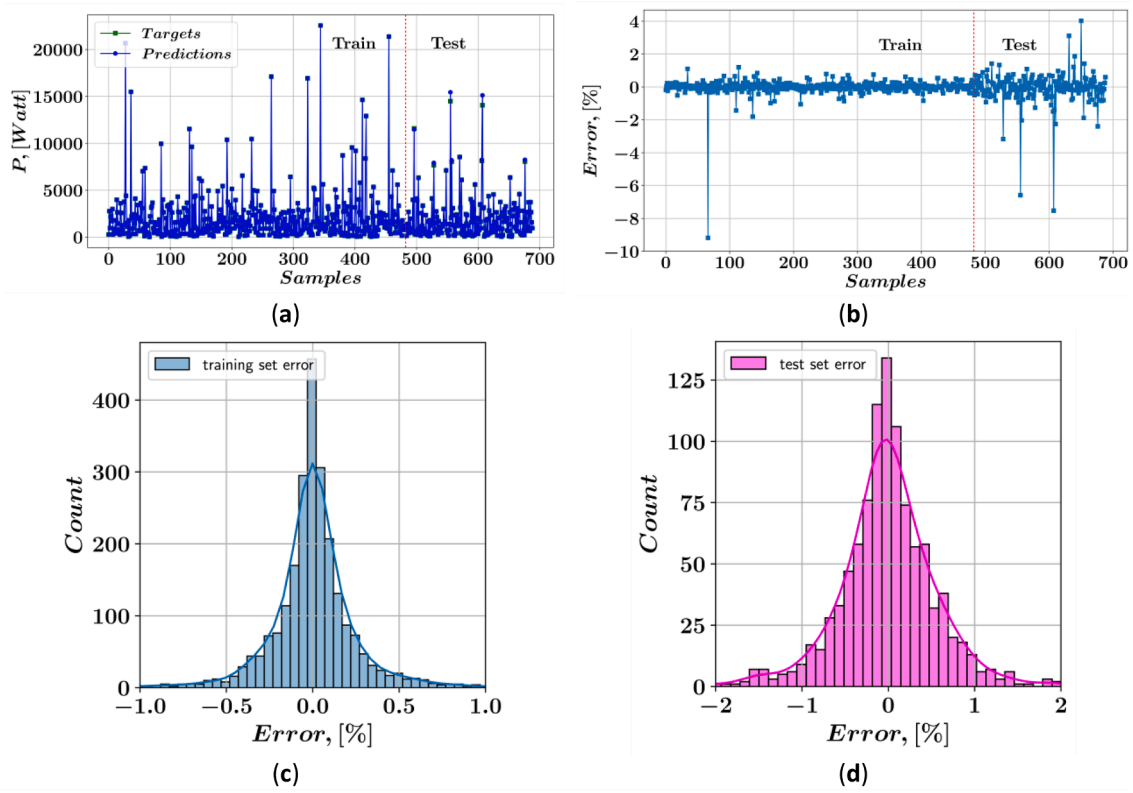


Fig. 4. A) predictions of the turbine power, b) error percentages, c) error distribution of the training set, and d) error distribution of the test set for the XGBoost model.

to which the predicted and target values overlap. Fig. 4b presents the error percentages on the same training and test sets subsets separately. While the percentage errors stay within the ±2% range throughout the training phase except for a single sample, a wider dispersion of the error percentages can be observed for the test phase. Although the error percentage stays in the ±2% range for most of the test phase, occasionally higher percentages up to 8% could be observed. Fig. 4c and d display histogram plots of the error distributions for the entire training

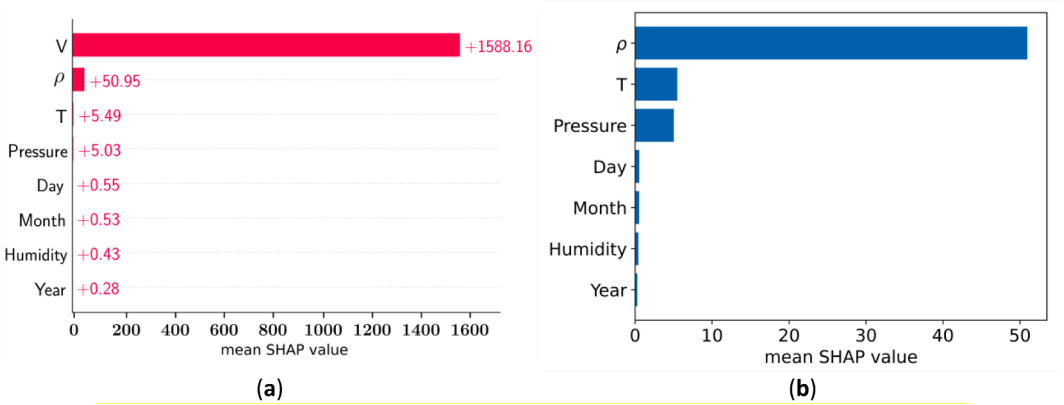


Fig. 5. SHAP feature importances a) including wind speed b) excluding wind speed.

and test sets. According to Fig. 4c and d, the percentage errors have a normal distribution of around 0% for both the training and the test sets, with the test set having a higher standard deviation. In the training set, 98.8% of the samples have an error of less than 1%, whereas, in the test set, 97.6% have less than 2% error.

Fig. 5 ranks the input features according to their impact on the XGBoost model output. The wind speed has the most significant impact on the predictions, followed by the air density. To clarify the differences in impact among the remaining variables, an additional feature importance plot is displayed in Fig. 5b, where the wind speed has been removed. It can be observed that the date parameters and the humidity have negligible impact on the model output. After density, the temperature and air pressure are the two most significant variables, albeit having an order of magnitude less significance than the air density.

Fig. 6 shows the first tree of the XGBoost model. The root of this tree has a wind speed of less than 18.25 m/s. The tree contains 19 internal nodes, all of which are split at different levels of wind speed. Twenty-one leaf nodes contain different values for the turbine power P. The entire XGBoost model of 8 input features consists of 172 trees. Each tree produces a single output value for any given data sample. The sum of all tree predictions determines the prediction of the entire XGBoost model for a given data point.

The ensemble learning models depend on parameters such as maximum tree depth in a model, the total number of trees, and the learning rate. These hyperparameters of the ensemble learning models have been tuned using the GridSearch function available in the Scikit-learn library of the Python programming language. The list of the tuned hyperparameters is given in Table 2. Fig. 7 visualizes the XGBoost model's performance for different learning rate values, the total number of trees, and the maximum tree depth. In Fig. 7, each subplot corresponds to a different value of the maximum tree depth parameter (*max_depth*). Four different levels of the learning rate (L_r) have been represented with different colors in each subplot. It was observed that increasing the learning rate reduces the number of trees necessary for good performance. The horizontal axis shows the number of trees (*n_estimators*) in each XGBoost model. The *n_estimators* parameter has been varied between 20 and 500 with steps of 20. The model performance is measured on the test set, which makes up 30% of the entire dataset, using the R^2 score. It can be observed that increasing the number of trees has a favorable effect on the model predictions. However, increasing the *n_estimators* parameter beyond 200 did not lead to any further improvement in the model performance. Furthermore, among the models with the same number of trees, those with the more significant learning rate performed better at all levels of the maximum tree depth.

SHAP summary plots are used to visually represent how different input features contribute to the model output. SHAP summary plots are based on cooperative game theory and the concept of Shapley values. The Shapley values quantify each player's contribution to the outcome of a cooperative game. In the context of machine learning, the input

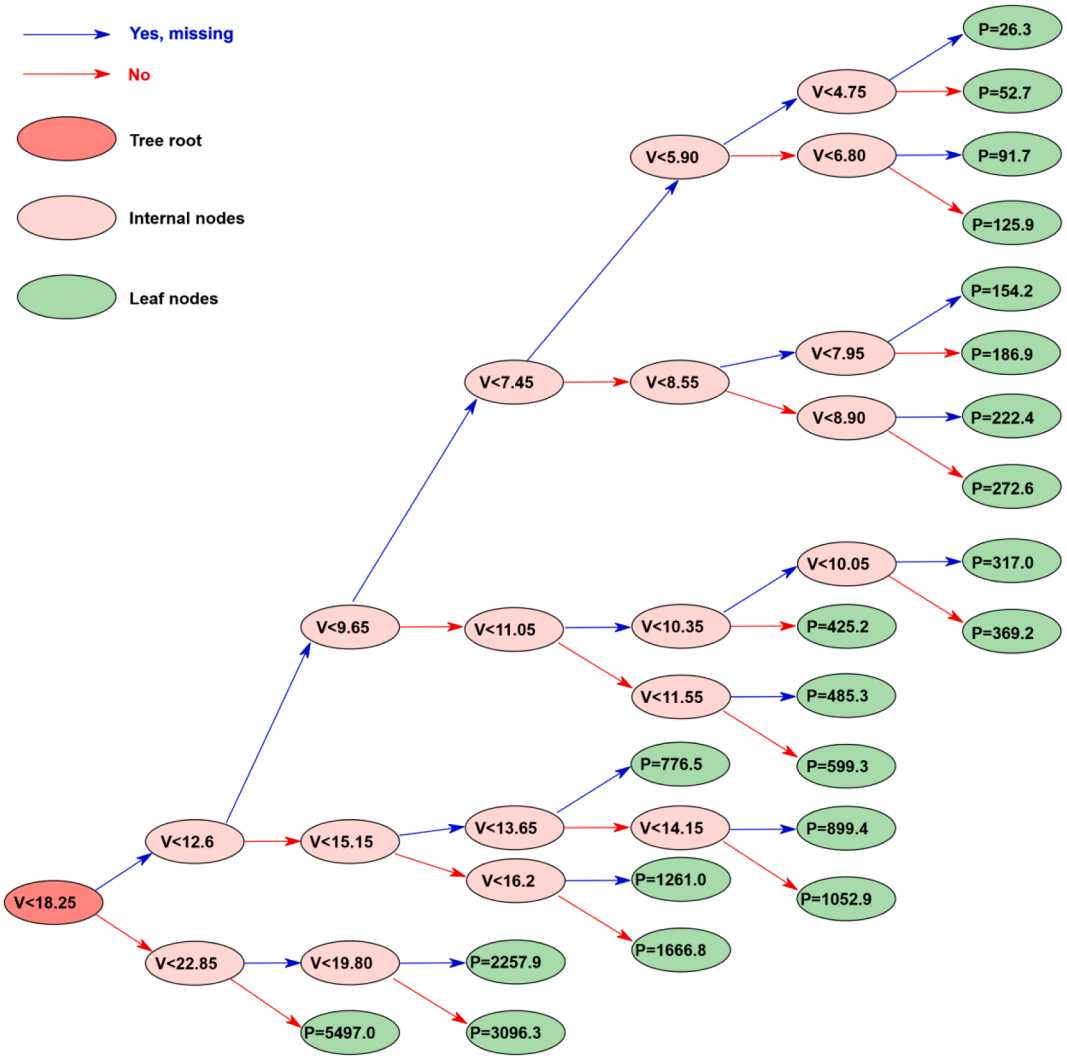


Fig. 6. The first tree of the XGBoost model with eight input features.

Table 2
Hyperparameters of the machine learning models.

Model	Parameter	Grid Search Range	Value
Random Forest	*n_estimators*	[20, 100, 200, 500]	200
	bootstrap	[True, False]	True
	min_samples_split	[2,5,10]	2
	min_samples_leaf	[1,5,10]	1
	max_features	[auto, sqrt, log2]	sqrt
XGBoost	*colsample_bytree*	[0.1,0.3,0.5,1.0]	0.5
	gamma	[0,10,20]	0
	learning_rate	[0.02, 0.05,0.1,0.2]	0.2
	max_depth	[2, 4, 6, 8,12]	2
	min_child_weight	[3,10,20,40,80,400]	3
	reg_alpha	[0,10,20]	0
	reg_lambda	[0,10,20]	10
LightGBM	*n_estimators*	[20, 100, 200, 500]	200
	colsample_bytree	[0.1,0.3,0.5,1.0]	1.0
	boosting_type	[gbdt, rf, dart]	gbdt
	num_leaves	[16, 32, 64]	32
	learning_rate	[0.02, 0.05,0.1,0.2]	0.02
CatBoost	*Iterations*	[500, 1000, 2000]	1000
	leaf_estimation_method	[Newton, Gradient, Exact]	Newton
	max_depth	[2,4,6,8,12]	6
	max_leaves	[16, 32, 64]	64
	learning_rate	[0.02, 0.05,0.1,0.2]	0.2
	bootstrap_type	[Bayesian, Bernoulli, MVS]	MVS
AdaBoost	*n_estimators*	[20, 100,200,500]	20
	learning_rate	[0.02, 0.05,0.1,0.2]	0.02
M5-Prime	*use_smoothing*	[True, False]	False
	use_pruning	[True, False]	True

features represent the players of the cooperative game, and the predictive modeling operation represents the cooperative game. The SHAP values in Fig. 8 quantify the difference in the model prediction when a feature is not included in the dataset for each one of the data points. Each dot in Fig. 8 corresponds to one of the data points in the dataset, and the horizontal position of a dot conveys information about the impact of a feature on the model output for that specific data point. Dots on the positive side of the zero line indicate an increasing contribution of the input feature on the model prediction, whereas dots on the negative side indicate a decreasing effect of the corresponding input feature.

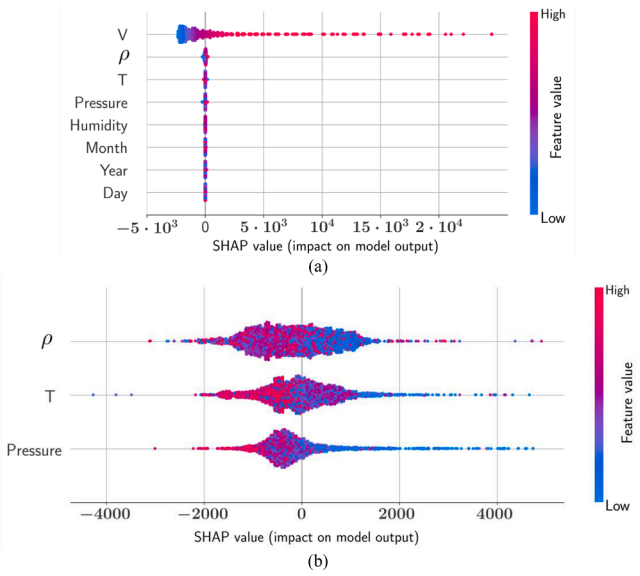


Fig. 8. SHAP summary plot for a) 8 input variables b) 3 input variables.

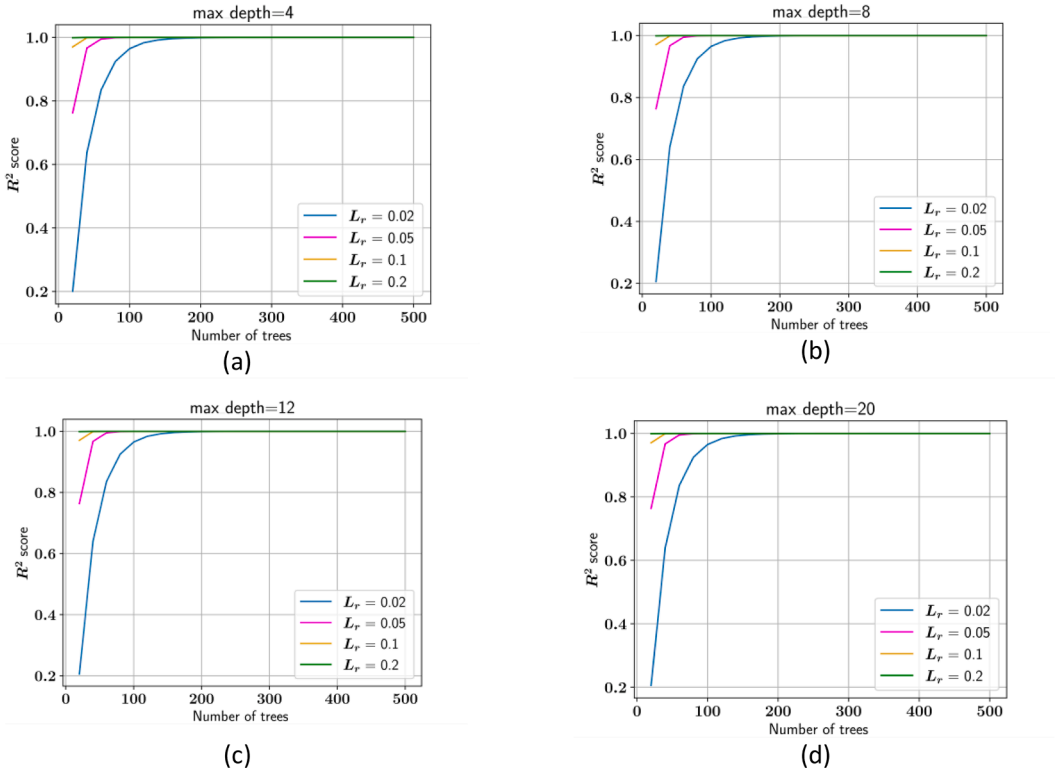


Fig. 7. XGBoost parameter tuning.

Furthermore, the distance of a data point from the zero line is proportional to the impact of the corresponding input feature on the model prediction in that particular data point. Fig. 8a shows the SHAP summary plot of the XGBoost model trained with eight input features. It can be observed that the wind speed has a significantly more significant impact on the model prediction. Fig. 8a shows that an increase in the predicted wind power values accompanies an increase in wind speed. Fig. 8b was obtained based on the results of the XGBoost algorithm after removing the wind speed and the variables with low impact, according to Fig. 5. Fig. 8b shows the summary plot for SHAP values in the range ±5000, making up 99.5% of all samples. The SHAP summary plot ranks the input features concerning their impact on the predicted turbine power. Each dot in the SHAP summary plot corresponds to one of the samples in the dataset, and its color represents the value a feature takes in a particular sample. As a feature's value increases, the dots' colors change from blue to red. In the summary plot, positive SHAP values indicate an increasing effect and negative SHAP values indicate a decreasing effect of a feature on the predicted turbine power. According to the SHAP summary plot, density has the most significant impact on the model prediction, followed by temperature and pressure. From Fig. 8b, it can be observed that a decrease in the temperature and pressure values is accompanied by an increase in the predicted turbine power since the samples with positive values have predominantly blue colors, whereas the density has a mixed color distribution on both sides of the zero SHAP value.

The feature dependence plots in Fig. 9 have been plotted to understand better each variable's effect on the model output and the dependencies between different input features. In Fig. 9, the horizontal axis shows the values of an input feature, and the vertical axis shows the SHAP values corresponding to this input feature in each data sample. In addition, the color bars to the right side of each subplot show the values of the variable most dependent on the input feature that the horizontal axis represents. According to Fig. 9a, wind speed is the most dependent variable on the air density, and increasing the density values also leads to an increase in the SHAP value for density. For density values less than 1.22 kg/m^{3,} density has a decreasing effect on the model output (negative SHAP value). In contrast, starting from that level, the density has an increasing effect on the model output. For any given density value, it can be observed that the impact of density on the model output increases with wind speed. According to Fig. 9b, as the value of temperature increases, the SHAP value for this variable decreases. At about 15 °C, a transition from positive SHAP values to negative SHAP values can be observed for this variable.

Furthermore, at any given temperature value, increasing the wind speed, which is the most dependent variable on temperature, increases the impact of temperature on the model output. Fig. 9c displays the feature dependence plot of the wind speed, which is the variable with the highest impact on the model output, according to Fig. 5. According to Fig. 9c, as the value of wind speed increases, the SHAP values associated with this variable increase exponentially. The increasing effect of this variable on the model output starts at a wind speed value of about 12 m/s. The air density is the most dependent variable on the wind speed; however, the density value does not significantly affect the SHAP value of wind speed since wind speed has an order of magnitude more significant impact on the model output. Fig. 9d shows that wind speed is the most dependent variable on pressure. The SHAP values associated with pressure are mainly concentrated around the zero SHAP value, and no significant variations in the impact of this variable can be observed except for a small number of data samples.

5. Conclusions

Due to the uncertainties associated with the reliability of wind power, the capacity to predict the available wind power in an area is essential. This work aimed to develop predictive models capable of accurately predicting wind turbine power. To this end, the most efficient ensemble learning algorithms from the literature have been applied in the current study. Based on five variables describing the atmospheric

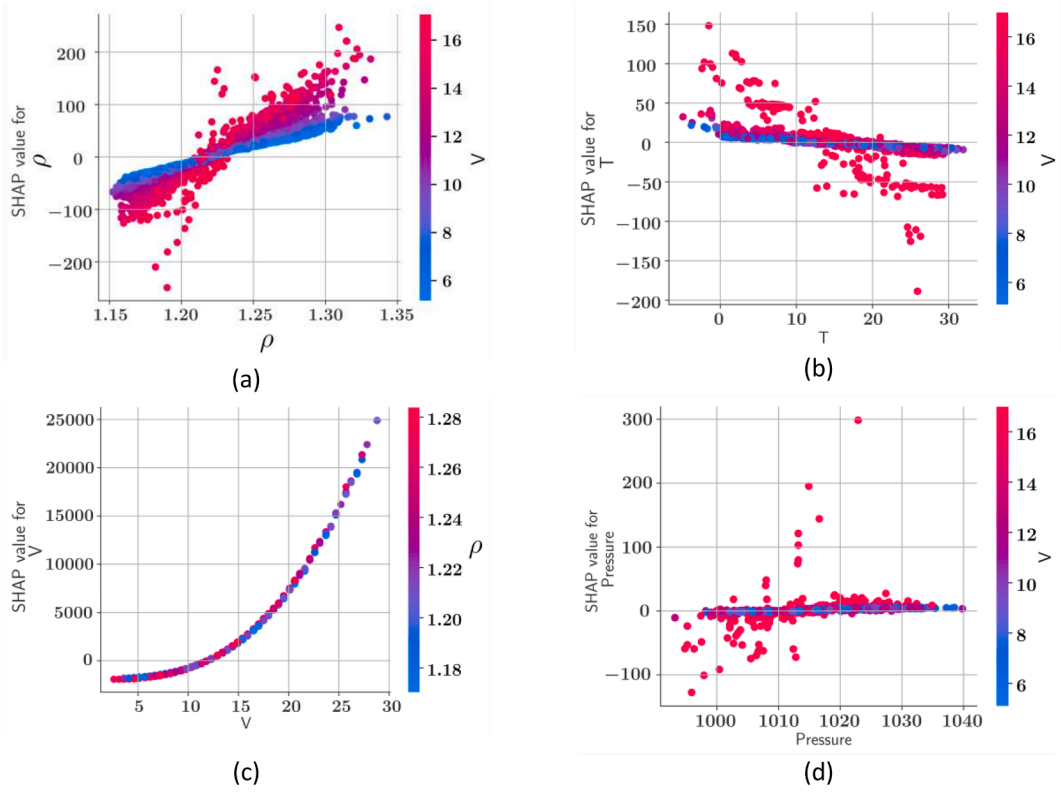


Fig. 9. Feature dependence plots for a) air density, b) temperature, c) wind speed, d) air pressure.

conditions and three variables describing the time of measurements, predictive models were trained on measurements recorded between 2011 and 2020 in the Çanakkale region of Turkey. It was observed that all of the six predictive models were able to predict the wind power with a coefficient of determination greater than 0.95 on the test set, while the XGBoost, CatBoost, Random Forest, and M5-Prime algorithms exceeded the R^2 score of 0.99. To the best of the authors' knowledge, the current study is the first in the area of wind power prediction that also made the predictive models interpretable using the SHAP algorithm. In addition to demonstrating the model performances, the current study also visualizes the impacts of different input features on the model output. The interdependencies of different input variables and their relationships to the wind power are demonstrated using SHAP summary plots and feature dependence plots. The wind speed was shown to have a significantly higher impact on the predictive model outputs, whereas the impact of the air humidity was negligible.

In the future, wind-solar hybrid energy systems can be modeled to increase the amount of energy obtained from wind energy and minimize investment costs. Furthermore, predictive models of the hybrid energy system can be developed by combining wind and solar energy data, taking into account the values of solar radiation and sunshine duration, which are two important parameters.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alkesaiberi, A., Harrou, F., & Sun, Y. (2022). Efficient wind power prediction using machine learning methods: A comparative study. *Energies, 15*(7), 2327.

Aslam, S., Herodotou, H., Mohsin, S. M., Javaid, N., Ashraf, N., & Aslam, S. (2021). A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids. *Renewable and Sustainable Energy Reviews, 144*, Article 110992.

Bakouregui, A.S.; Mohamed, H.M.; Yahia, A.; Benmokrane, B. Explainable extreme gradient boosting tree-based prediction of load-carrying capacity of FRP-RC columns. Engineering Structures, 2021, 245, 112836, https://doi.org/10.1016/j.engstruct.2021.112836.

Behnood, A., Behnood, V., Gharehveran, M. M., & Alyamac, K. E. (2017). Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. *Construction and Building Materials, 142*, 199–207. https://doi.org/10.1016/j.conbuildmat.2017.03.061

Blazakis, K., Katsigiannis, Y., & Stavrakakis, G. (2022). One-Day-Ahead Solar Irradiation and Windspeed Forecasting with Advanced Deep Learning Techniques. *Energies, 15*(12), 4361.

Buturache, A.N. and Stancu, S., 2021. Wind energy prediction using machine learning.

Chang, V., Chen, Y., Zhang, Z. J., Xu, Q. A., Baudier, P., & Liu, B. S. (2021). The market challenge of wind turbine industry-renewable energy in PR China and Germany. *Technological Forecasting and Social Change, 166*, Article 120631.

Chen, T., & Guestrin, C. (2016). August). Xgboost: A scalable tree boosting system. In *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Cheng, B., Du, J., & Yao, Y. (2022). Machine learning methods to assist structure design and optimization of Dual Darrieus Wind Turbines. *Energy, 244*, Article 122643.

Degtyarev, V. V., & Naser, M. Z. (2021). Boosting machines for predicting shear strength of CFS channels with staggered web perforations. *Structures (Vol., 34*, 3391–3403. https://doi.org/10.1016/j.istruc.2021.09.060

Demolli, H., Dokuz, A. S., Ecemis, A., & Gokcek, M. (2019). Wind power forecasting based on daily wind speed data using machine learning algorithms. *Energy Conversion and Management, 198*, Article 111823.

Deng, Y., Jia, H., Li, P., Tong, X., Qiu, X., & Li, F. (2019). In *June. A deep learning methodology based on bidirectional gated recurrent unit for wind power prediction* (pp. 591–595). IEEE.

Drucker, H. (1997, July). Improving regressors using boosting techniques. In *ICML'97: Proceedings of the Fourteenth International Conference on Machine Learning* (Vol. 97, pp. 107-115).

EIA. International Energy Outlook 2019; U.S. Energy Information Administration: Washington, DC, USA, 2019.

ELEKTRİK PİYASASI YILLIK SEKTÖR RAPORU (EPİAŞ), 2019. Available online: https://www.epdk.gov.tr/Detay/Icerik/3-0-24-3/elektrikyillik-sektor-raporu.

Eyecioglu, O., Hangun, B., Kayisli, K., & Yesilbudak, M. (2019). In *November). Performance comparison of different machine learning algorithms on the prediction of wind turbine power generation* (pp. 922–926). IEEE.

Feng, D. C., Wang, W. J., Mangalathu, S., Hu, G., & Wu, T. (2021). Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements. *Engineering Structures, 235*, Article 111979. https://doi.org/10.1016/j.engstruct.2021.111979

Ghoddusi, H., Creamer, G. G., & Rafizadeh, N. (2019). Machine learning in energy economics and finance: A review. *Energy Economics, 81*, 709–727.

Gupta, D., Natarajan, N., & Berlin, M. (2022). Short-term wind speed prediction using hybrid machine learning techniques. *Environmental Science and Pollution Research, 29*(34), 50909–50927.

Han, S., Qiao, Y., Yan, P., Yan, J., Liu, Y., & Li, L.i. (2020). Wind turbine power curve modeling based on interval extreme probability density for the integration of renewable energies and electric vehicles. *Renewable Energy, 157*, 190–203.

Höök, M., & Tang, X. (2013). Depletion of fossil fuels and anthropogenic climate change—A review. *Energy Policy, 52*, 797–809.

Howell, R., Qin, N., Edwards, J., & Durrani, N. (2010). Wind tunnel and numerical study of a small vertical axis wind turbine. *Renewable Energy, 35*(2), 412–422.

Hu, H., Li, Y., Zhang, X., & Fang, M. (2022). A novel hybrid model for short-term prediction of wind speed. *Pattern Recognition, 127*, Article 108623.

Huang, L., Li, L., Wei, X., & Zhang, D. (2022). Short-term prediction of wind power based on BiLSTM–CNN–WGAN-GP. *Soft Computing*, 1–15.

Huang, B., Liu, X., Wang, S., Pan, L., & Chang, V. (2021). Multi-agent reinforcement learning for cost-aware collaborative task execution in energy-harvesting D2D networks. *Computer Networks, 195*, Article 108176.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems, 30*.

Klemeš, J. J., Varbanov, P. S., Ocłoń, P., & Chin, H. H. (2019). Towards efficient and clean process integration: Utilisation of renewable resources and energy-saving technologies. *Energies, 12*(21), 4092.

Li, L. L., Zhao, X., Tseng, M. L., & Tan, R. R. (2020). Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *Journal of Cleaner Production, 242*, Article 118447.

Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.

Mahaseth, R., Kumar, N., Aggarwal, A., Tayal, A., Kumar, A., & Gupta, R. (2022). Short term wind power forecasting using k-nearest neighbour (KNN). *Journal of Information and Optimization Sciences, 43*(1), 251–259.

Maroufpoor, S., Sanikhani, H., Kisi, O., Deo, R. C., & Yaseen, Z. M. (2019). Long-term modelling of wind speeds using six different heuristic artificial intelligence approaches. *International Journal of Climatology, 39*(8), 3543–3557.

McCauley, D., Ramasar, V., Heffron, R. J., Sovacool, B. K., Mebratu, D., & Mundaca, L. (2019). Energy justice in the transition to low carbon energy systems: Exploring key themes in interdisciplinary research. *Applied Energy, 233*, 916–921.

Nguyen, T. H. T., & Phan, Q. B. (2022). Hourly day ahead wind speed forecasting based on a hybrid model of EEMD, CNN-Bi-LSTM embedded with GA optimization. *Energy Reports, 8*, 53–60.

Owusu, P. A., & Asumadu-Sarkodie, S. (2016). A review of renewable energy sources, sustainability issues and climate change mitigation. *Cogent Engineering, 3*(1), 1167990.

Piotrowski, P., Baczyński, D., Kopyt, M., & Gulczyński, T. (2022). Advanced Ensemble Methods Using Machine Learning and Deep Learning for One-Day-Ahead Forecasts of Electric Energy Production in Wind Farms. *Energies, 15*(4), 1252.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in neural information processing systems, 31*.

Sun, H., Qiu, C., Lu, L., Gao, X., Chen, J., & Yang, H. (2020). Wind turbine power modelling and optimization using artificial neural network with wind field experimental data. *Applied Energy, 280*, Article 115880.

Taghizadeh-Hesary, F., & Yoshino, N. (2020). Sustainable solutions for green financing and investment in renewable energy projects. *Energies, 13*(4), 788.

Ti, Z., Deng, X. W., & Zhang, M. (2021). Artificial Neural Networks based wake model for power prediction of wind farm. *Renewable Energy, 172*, 618–631.

TÜREB., 2019. Türkiye Rüzgâr Enerjisi İstatistik Raporu. Türkiye Rüzgâr Enerjisi Birliği. Green Office Kızılırmak Mah. 1443. Cad. No: 22/16 Çukurambar/Ankara.

Ucar, A., & Balo, F. (2010). Assessment of wind power potential for turbine installation in coastal areas of Turkey. *Renewable and Sustainable Energy Reviews, 14*(7), 1901–1912.

Wang, Y., & Witten, I. H. (1996). *Induction of model trees for predicting continuous classes*. Hamilton, New Zealand: University of Waikato, Department of Computer Science (Working paper 96/23).

Xu, K., Yan, J., Zhang, H., Zhang, H., Han, S., & Liu, Y. (2021). Quantile based probabilistic wind turbine power curve model. *Applied Energy, 296*, Article 116913.

Yan, J., & Ouyang, T. (2019). Advanced wind power prediction based on data-driven error correction. *Energy conversion and management, 180*, 302–311.

Yang, H., Shahzadi, I., & Hussain, M. (2021). USA carbon neutrality target: Evaluating the role of environmentally adjusted multifactor productivity growth in limiting carbon emissions. *Journal of Environmental Management, 298*, Article 113385. https://doi.org/10.1016/j. jenvman.2021.113385

Zhang, J., Jiang, X., Chen, X., Li, X., Guo, D. and Cui, L., (2019a), April. Wind power generation prediction based on LSTM. In Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence (pp. 85-89).

Zhang L., Wang K., Lin W., Geng T., Lei Z., Wang Z., (2019b), April. Wind power prediction based on improved genetic algorithm and support vector machine. In IOP Conference Series: Earth and Environmental Science (Vol. 252, No. 3, p. 032052). IOP Publishing.

Zhang, Y., & Li, R. (2022). Short term wind energy prediction model based on data decomposition and optimized LSSVM. *Sustainable Energy Technologies and Assessments, 52*, Article 102025.

Zhang, C., Ma, H., Hua, L., Sun, W., Nazir, M. S., & Peng, T. (2022a). An evolutionary deep learning model based on TVFEMD, improved sine cosine algorithm, CNN and BiLSTM for wind speed prediction. *Energy, 254*, Article 124250.

Zhang, C., Peng, T., & Nazir, M. S. (2022b). A novel hybrid approach based on variational heteroscedastic Gaussian process regression for multi-step ahead wind speed forecasting. *International Journal of Electrical Power & Energy Systems, 136*, Article 107717.

Zhang, J., Yan, J., Infield, D., Liu, Y., & Lien, F. S. (2019c). Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model. *Applied Energy, 241*, 229–244.