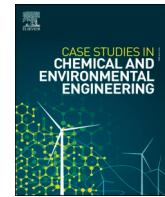




Contents lists available at ScienceDirect

Case Studies in Chemical and Environmental Engineering

journal homepage: www.sciencedirect.com/journal/case-studies-in-chemical-and-environmental-engineering

Case Report

Advanced techniques for wind energy production forecasting: Leveraging multi-layer Perceptron + Bayesian optimization, ensemble learning, and CNN-LSTM models

Seyed Matin Malakouti ^a, Farrokh Karimi ^b, Hamid Abdollahi ^a, Mohammad Bagher Menhaj ^{a,*}, Amir Abolfazl Suratgar ^a, Mohammad Hassan Moradi ^c^a Distributed and Intelligent Optimization Research Laboratory, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran^b Research Fellow, Institute for Research in Fundamental Sciences (IPM), Sharif University of Technology, Iran^c Department of Biomedical Engineering, BioelectricAmirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:

Wind turbines
 Ensemble (gradient boosting and xgboost)
 Multi-layer Perceptron with bayesian optimization
 CNN-LSTM method

ABSTRACT

Due to the shortage of fossil fuels in many countries, power plants that rely on fossil fuels will be phased out in favor of wind turbines as the primary source of energy generation. These fossil fuel plants wreak havoc on the natural world, making humans and other life forms susceptible to illness. The production potential of wind turbines was investigated. Consequently, methods such as XGBOOST, Multi-Layer Perceptron with Bayesian Optimization (MLP + BO), Gradient Boosting Regression Tree (GBDT), Ensemble (gradient boosting and xgboost), and CNN Long Short-Term Memory (CNN-LSTM) have been utilized. A mean square error (MSE) of 7.2 in 45 seconds was achieved using the Ensemble technique, and an MSE of 6.8 in 450 seconds was obtained with the CNN-LSTM method. Wind power is readily available and straightforward to acquire globally, indicating its potential as a reliable and sustainable energy source.

1. Introduction

Nowadays, wind energy has grown in significance as a source of electrical energy and has established itself as a vital component of a sustainable energy portfolio. With the use of wind turbine subsystems, this technique converts wind mechanical energy into electrical energy. It is unique as a potential energy generation method for the future since it minimizes carbon emissions, provides an endless supply, and is kind to the environment. Because wind energy is produced using a clean, renewable resource, it has significant environmental protection benefits. This renewable energy source improves the environmental effects of energy plants, particularly with regard to air quality, and reduces the release of carbon dioxide. As so, it effectively contributes to the fight against climate change (see Fig. 3).

Furthermore, the development of wind farms reduces adverse impacts on ecosystems and stops environmental degradation.

Wind speed fluctuation is a serious obstacle to the efficient production of energy, even if uninterrupted and powerful wind speeds are necessary. We need to forecast wind speed in the short term in order to

tackle this challenge. These forecasts are utilized to maximize energy output, guard against equipment and turbine damage, and maintain a secure working environment. Predictions of wind speed for the near future are created by examining meteorological data. These forecasts are essential for optimizing energy output, maintaining facility safety, and modifying the operating heights and speeds of turbines. For wind energy plants to be sustainable and efficient, these forecasts must be accurate.

Wind energy has grown increasingly significant owing to its reliable, non-polluting, and cost-free nature [1,2]. Therefore, renewable energy sources (RES) have recently gotten much attention. Furthermore, RES has attracted much interest due to its ability to minimize greenhouse gas emissions. Despite the many benefits of wind energy, reliable power projections are complex. Not only are meteorological variables and seasonal effects practical considerations in wind generation, but the wind's intermittent nature makes predicting the turbines' output capacity more challenging [3]. Climatic conditions heavily influence wind generation. Wind energy is essential for developing a country's social and economic infrastructure. Because of this issue, it is critical to anticipate wind power that is both exact and reliable [4].

* Corresponding author.

E-mail addresses: matin3475malakoti@gmail.com (S.M. Malakouti), farrokh.karimi@sharif.edu (F. Karimi), hamid.abdollahi@aut.ac.ir (H. Abdollahi), menhaj@aut.ac.ir (M.B. Menhaj), a-suratgar@aut.ac.ir (A.A. Suratgar), mhmoradi@aut.ac.ir (M.H. Moradi).

The article by Abdelzaher and Awad [5] explores the implementation of new drip irrigation technologies in Upper Egypt as a means to achieve the Sustainable Development Goals (SDGs) for the circular economy and the water-food nexus. The study emphasizes the critical importance of water conservation in Egypt due to rising living standards, population growth, and increased agricultural and industrial activities. Given that the Nile River is Egypt's primary water source, innovative irrigation systems like drip irrigation are essential to optimize water usage and enhance agricultural productivity. Adoption of drip irrigation reduces dependency on fossil fuels by integrating renewable energy sources such as photovoltaic panels. This transition supports the reduction of CO₂ emissions and promotes sustainable agricultural practices.

2. Literature background

Physical theories, conventional mathematical models, machine learning theories, and various combination models are the several types of suggested methods for wind energy forecast [6]. In order to study the changing dynamics of the combined system in the area of wind power, Li et al. produced a mathematical representation of a single system in 2023 that was based upon a conical Concrete-Filled Double Skin Tube wind turbine. The equation that governs of the framework was developed using the Lagrangian methodology, and the technique known as Galerkin was utilized to solve it analytical. Wind tunnel testing of scaled models were used to evaluate the created model, and in-depth parametric discussions ensued. This research produced a trustworthy analytical approach that has been validated [7] and has the ability to forecast the basic frequencies and displacement response with accuracy. In contrast to physical models, statistical frameworks incorporate a Numerical Weather Prediction system and use atmospheric information for predicting wind speed [8,9]. Another useful forecasting tool for wind speed prediction is the Weather Research Forecast approach, which takes into account the resolution of the selected region as well as the complexities of the terrain. In short-term wind speed prediction, it has proven to work well [10]. Machine learning models have been created to improve prediction models as synthetic approaches and statistical models have progressed [11].

It is crucial to remember that no one technique can be used to every dataset, but [12]. As a result, hybrid approaches have been proposed to improve wind speed forecast accuracy by combining classic statistical models with machine learning algorithms [10]. In order to produce superior forecasting models, merging individual prediction models has been the subject of several studies. However, a survey of research on wind speed estimate has shown that these methods lack generally recognized standards [10]. Other models for forecasting, such as hybrid approaches that employ dataset pretreatment techniques, error management strategies, and parameter optimization, also mix several prediction methodologies [12]. With careful analysis, machine learning methods can find patterns and connections in large datasets, which helps them make the best judgments and forecasts [13]. Programs for artificial intelligence perform better and grow more popular as they have access to bigger amounts of data. These methods build forecasting models and identify the relationships between variables by utilizing the structure of regression analysis methods [14]. In the world of machine learning, linear regression is a supervised learning technique that forecasts the value of a parameter based on another factor. It finds the best-fit line for a collection of paired data by using linear regression calculators and the least squares approach [15]. However, by applying the decision tree approach repeatedly and using randomly selected subsets of the dataset for the decision trees, random forest regression increases prediction accuracy [16]. A effective technique based on the eXtreme Gradient Boosting decision tree algorithm is called XGBoost regression [17]. Operations happen fast because eXtreme Gradient Boosting reveals the decision tree using the parallel working approach. One of its most well-known and significant features is its ability to identify the proper

point for data classification into multiple trees by using the dataset's observation points based on their weight values.

Grève et al. [18] developed a variety of machine learning models to assist community members in optimizing their resource scheduling for both generation and consumption, aiming to reduce electricity costs. These models were used to predict local wind power generation for the following day. Besides individual algorithms, an ensemble model was created by combining two neural network algorithms—the Bidirectional Long Short-Term Memory Network (BLSTM) and the feedforward Multilayer Perceptron (MLP)—along with two tree-based methods—Gradient Boosting Decision Tree (GBDT) and Random Forest (RF). The ensemble's overall performance was determined by averaging the outputs of these four algorithms. The researchers found that the ensemble model surpassed the individual algorithms, improving forecasting accuracy by 10 % in terms of Root Mean Squared Error (RMSE), achieving an RMSE of 2327 kW for day-ahead predictions.

Dimitropoulos et al. [19] proposed another method for predicting short-term energy production from a community's solar plant through an infrastructure and monitoring system. They trained four machine learning algorithms to forecast energy production up to 6 hours ahead. The algorithms used were Extreme Gradient Boost (XGBoost), Support Vector Regression (SVR), Long Short-Term Memory Network (LSTM), and Multiple Linear Regression (MLR). Historical consumption data and weather conditions were used to train these forecast algorithms. For evaluation, the dataset was split into 80 % for training and 20 % for testing. XGBoost showed the highest accuracy, achieving an RMSE of 1.834 kWh for 1-h interval predictions, 6 hours ahead. The authors suggested that these results could help schedule energy supply for communities and form the basis for more complex applications needing precise short-term predictions, such as predictive maintenance or energy trading.

Baba et al. [20] aimed to predict the daily power consumption for the upcoming year in a local industrial region. They designed an Artificial Neural Network (ANN) with a (5-15-5-1) architecture, using the Hebbian learning rule to update weights between the input layer and the first hidden layer, while other links were trained using classical back-propagation. The performance of this ANN (5-15-5-1) was compared with two other ANNs with architectures of (5-5-1) and (5-5-5-1), and a variant of the Multiple Model Particle Filter (MMPF) probabilistic method. The developed ANN (5-15-5-1) achieved slightly better results than the other methods, with an average RMSE of 4.921 kWh for daily predictions a year ahead. The authors concluded that increasing the number of hidden layers could improve performance metrics but warned that it might also increase the risk of overfitting or high variance.

Musbah et al. [21] presented a forecasting model for energy management using either Light Gradient Boosted Machine (LightGBM) or Random Forest (RF) to predict the best combination of energy sources in a Hybrid Energy System (HES). The HES case study included wind, gasoline, and a diesel generator to provide electricity to a remote area. Initially, a historical demand-side dataset was used to calculate five criteria using the TOPSIS method, which included energy efficiency, CO₂ emissions, gasoline and diesel fuel prices, labor, and fuel consumption. The Analytic Hierarchy Process (AHP) and Fuzzy Analytic Hierarchy Process (FAHP) were used to determine the weight values for these criteria. In the subsequent phase, the best combination of energy sources was predicted using the RF and LightGBM algorithms to validate the proposed approach. The results showed that the RF algorithm performed better, achieving an accuracy of 81.81 %, compared to LightGBM's 68.6 %. The study also indicated that the differences between AHP and FAHP were minimal, suggesting that either method could be used to achieve similar results.

Zoji et al. [22] conducted a study to predict a more accurate energy usage profile in an office building for the upcoming hours. They employed a methodology to develop genetic fuzzy rule-based systems using the iterative rule learning approach (GFS.FR.MOGUL). The findings were compared with earlier methods, including two fuzzy-based

S.M. Malakouti et al.

systems and various artificial neural network approaches. The study found that the proposed method could compute more realistic electricity consumption estimates for the upcoming hours, achieving an average MAPE forecast error of 9.54 % for 12-h ahead predictions, and showing a reduced standard deviation compared to previous methods.

Zozi et al. [23] introduced a contextual learning approach for energy forecasting to support decisions made by Building Energy Management Systems (BEMS). BEMS are designed to ensure continuous energy availability, reliability, and access for consumers. Their approach includes a contextual dimension that identifies and groups various observed contexts based on similarities. Several machine learning techniques were used, including SVM, HyFIS, WM, and GFS. FR. MOGUL. Experiments with real data on energy generation, consumption, and contextual information from sensors installed in a building were conducted to validate the approach. SVM achieved a MAPE of 6.89 % for one-hour ahead consumption predictions. The authors stated that the results could provide recommendations related to consumption with smaller error metrics compared to other machine learning methods.

Al-Shanableh [26] explored the potential of using a Fuzzy Inference System (FIS) to predict the energy consumption of residential buildings in northern Cyprus. Factors such as climate zone, floor area, year of construction, number of occupants, and house type were considered to estimate energy consumption per unit floor area. The FIS model was developed using data from 70 questionnaires, resulting in 67 rules, and tested with an additional 15 questionnaires. The FIS model's predictions closely matched the actual values, with an R^2 value of 0.9884 and an RMSE of 6.6115 kWh/m² year for the training set. For the testing set, the model produced less accurate results, with an R^2 value of 0.8851 and an RMSE of 20.3347 kWh/m² year.

To enhance the interpretability and performance of machine learning models, fuzzy logic has been widely utilized. Researchers have provided a seminal approach to generating fuzzy rules by learning from examples, significantly contributing to the field of fuzzy systems and rule-based methods [24].

The methodology for developing genetic fuzzy rule-based systems has seen substantial advancements. Researchers have introduced MOGUL, a comprehensive approach for obtaining genetic fuzzy rule-based systems under the iterative rule learning approach, offering robust solutions for complex decision-making problems [25].

In the domain of neural networks, the book "Neural Network Design" by researchers remains a critical resource. It provides in-depth knowledge and practical insights into designing effective neural network architectures, making it essential for understanding advanced neural network concepts [36].

A comprehensive survey on modern trainable activation functions was conducted by researchers. This survey offers an extensive review of various activation functions, highlighting their significance in training deep neural networks and improving model performance [38].

The concept of regularization in solving ill-posed problems was pioneered by researchers. Their work on the regularization method provides foundational techniques that are crucial for addressing issues of overfitting and ensuring the stability of solutions in various computational problems [39].

Prasad et al. [27] researched energy sharing in Zero Energy Communities (ZEC) to achieve zero net energy usage annually. Unlike previous studies that focused on individual building economic gains, this research prioritized improving the overall energy status of the community. Using the Deep Reinforcement Learning (DRL) algorithm DQN, the expected value of actions in specific states was approximated, considering long-term rewards. The study examined Winter and Summer conditions across three scenarios with different community configurations and scales. The results showed significant improvements compared to a strategy without energy sharing, with a 40 kWh enhancement over 3 days of Winter for three houses and a 60 kWh improvement for four houses over the same period during Summer. These findings highlighted the capability of buildings to learn to

Case Studies in Chemical and Environmental Engineering 10 (2024) 100881

collaborate, developing a policy similar to the optimal one, thus enhancing community energy independence from the supply grid.

3. Research objectives

1. To investigate the production potential of wind turbines using advanced machine learning techniques.
2. To utilize methods such as XGBOOST, Multi-Layer Perceptron with Bayesian Optimization (MLP + BO), Gradient Boosting Regression Tree (GBDT), Ensemble (gradient boosting and xgboost), and CNN Long Short-Term Memory (CNN-LSTM) for accurate power prediction.
3. To compare the performance of these methods in terms of mean square error (MSE) and computational time.
4. To assess the feasibility of replacing fossil fuel power plants with wind turbines as a primary energy source.

This study offers several theoretical contributions to the field of wind energy forecasting and the broader domain of renewable energy management.

The primary theoretical contribution of this research is the innovative integration of multiple advanced machine learning techniques to improve the accuracy of wind energy production forecasts. Traditional methods have often struggled with the non-linear and complex nature of wind patterns. By leveraging XGBOOST, Multi-Layer Perceptron combined with Bayesian Optimization (MLP + BO), Gradient Boosting Regression Tree (GBDT), Ensemble Learning (combining gradient boosting and XGBOOST), and Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) models, this study provides a robust framework for addressing these challenges. This multi-faceted approach showcases the potential of hybrid models in capturing the intricate dynamics of wind energy data, thus contributing to the theoretical understanding of machine learning applications in renewable energy forecasting.

Another significant theoretical contribution is the demonstration of how combining different machine learning models can lead to substantial improvements in predictive accuracy and computational efficiency. The ensemble approach, which aggregates the strengths of individual algorithms, highlights the importance of model diversity in achieving superior forecasting performance. This finding aligns with and extends the existing body of literature that emphasizes the benefits of ensemble methods in various predictive tasks.

This study also contributes to the theoretical discourse on managing the variability and intermittency of wind energy. By improving the accuracy of wind energy forecasts, the research supports more reliable and efficient integration of wind power into energy grids. This addresses a critical challenge in the renewable energy sector, thereby enhancing the theoretical framework for energy management and grid stability.

From a methodological perspective, the use of Bayesian Optimization for hyperparameter tuning in the MLP + BO model is particularly noteworthy. This technique enhances model performance by efficiently navigating the hyperparameter space, offering a valuable contribution to the methodological practices in machine learning applications. Additionally, the integration of CNNs with LSTM networks, which combines spatial feature extraction with temporal sequence learning, provides a novel methodological approach well-suited for time-series prediction tasks.

Finally, the study's focus on improving wind energy forecasting contributes to the broader theoretical understanding of sustainable energy systems. Accurate predictions are crucial for optimizing energy management and reducing dependency on fossil fuels. This aligns with global sustainability goals and supports the transition to more resilient and environmentally friendly energy infrastructures.

In summary, this study's theoretical contributions span multiple dimensions, including the integration of advanced machine learning techniques, methodological innovations, and implications for

sustainable energy systems. By addressing key challenges in wind energy forecasting, this research not only advances the theoretical framework of renewable energy management but also provides practical insights for enhancing the reliability and efficiency of wind power integration.

4. Method

4.1. XGBOOST

XGboost is a recently developed ML method currently being used extensively in an entirely distinct field, and this method assists in avoiding over-fitting and optimizing computing capabilities [28,29]. The XGboost similar estimate is also conducted automatically during the training procedure. XGboost's additive learning processes are detailed. To fix a weak learner's limitations, the first learner is fitted to the whole input dataset, and they fit the second algorithm to the residuals [29]. This algorithm schematic is shown in Fig. 1. This acceptable strategy is repeated many times until the stopping condition is met. The final method forecast is the sum of each learner's forecasts. The formula below defines the general variation in phase forecasting (see Fig. 2).

$$f_i^p = \sum_{k=1}^l f_k(x_i) = f_i^{(p-1)} + f_i(x_i) \quad (1)$$

Where.

$f_p(x_i)$: learner at phase p ,

$f_i^p, f_i^{(p-1)}$: forecasting at phase p and $p-1$

x_i : the input variables.

A. XGBOOST

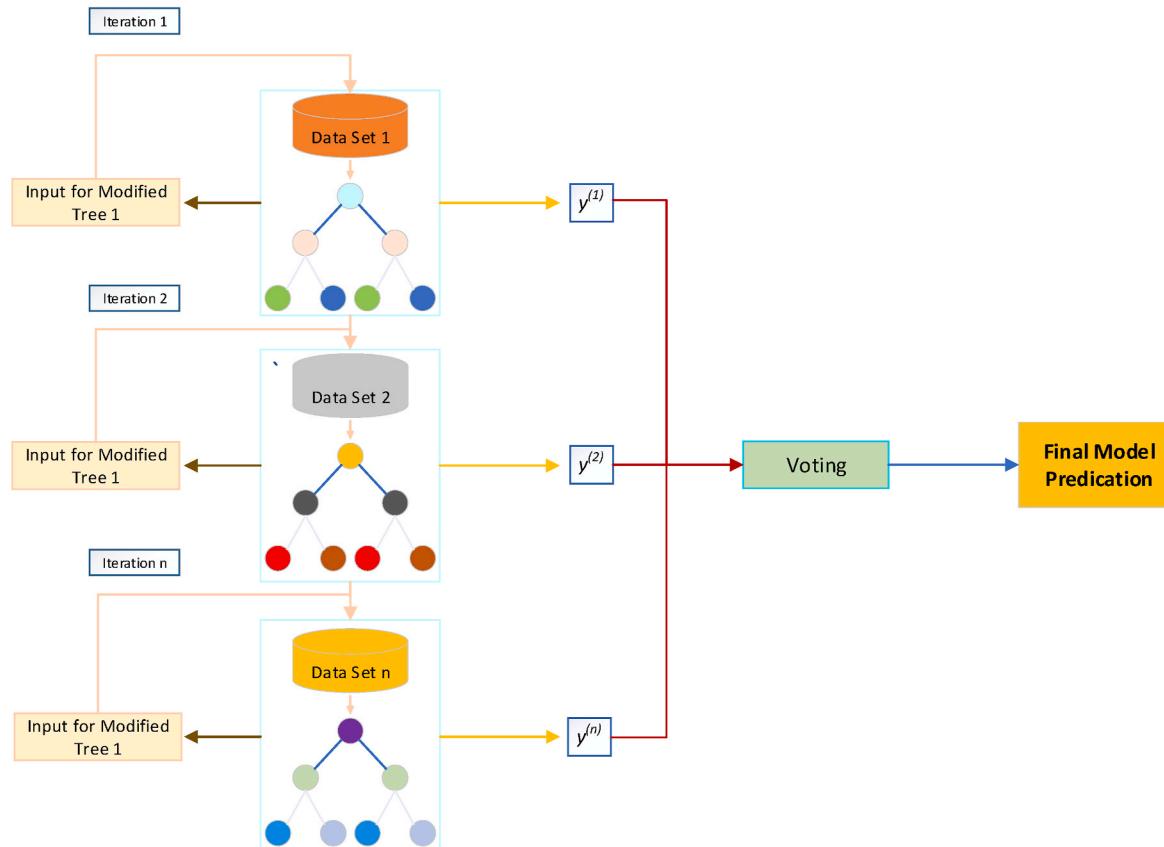


Fig. 1. Xgboost.

To minimize over-fitting difficulties without sacrificing method computing efficiency, the XGBoost develops an analytical technique to evaluate the method's "goodness" according to the original functionality.

$$Obj^{(p)} = \sum_{k=1}^n l(\bar{y}_i, y_i) + \sum_{k=1}^p \sigma(f_i) \quad (2)$$

Where.

l : the loss function

n : the total amount of data collected

σ : the regularization term defined as follows:

$$\sigma(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (3)$$

Where.

ω : vector scores in leaves

γ : lowest possible splitting cost for a leaf node

λ : the regularization factors

4.2. Gradient Boosting Regression Tree (GBDT)

GBDT is a decision tree ensemble learning method that uses the Boosting method for collective learning. The forward distribution technique distinguishes the Gradient Boosting Regression Tree from the Ada boost; however, the weak learner restricts the intelligent application of the Classification and Regression Tree (CART) regression tree approach, and iterative thinking and Adaboost are also dissimilar.

Assuming that $G_{t-1}(x)$ was the best learner in the previous iteration

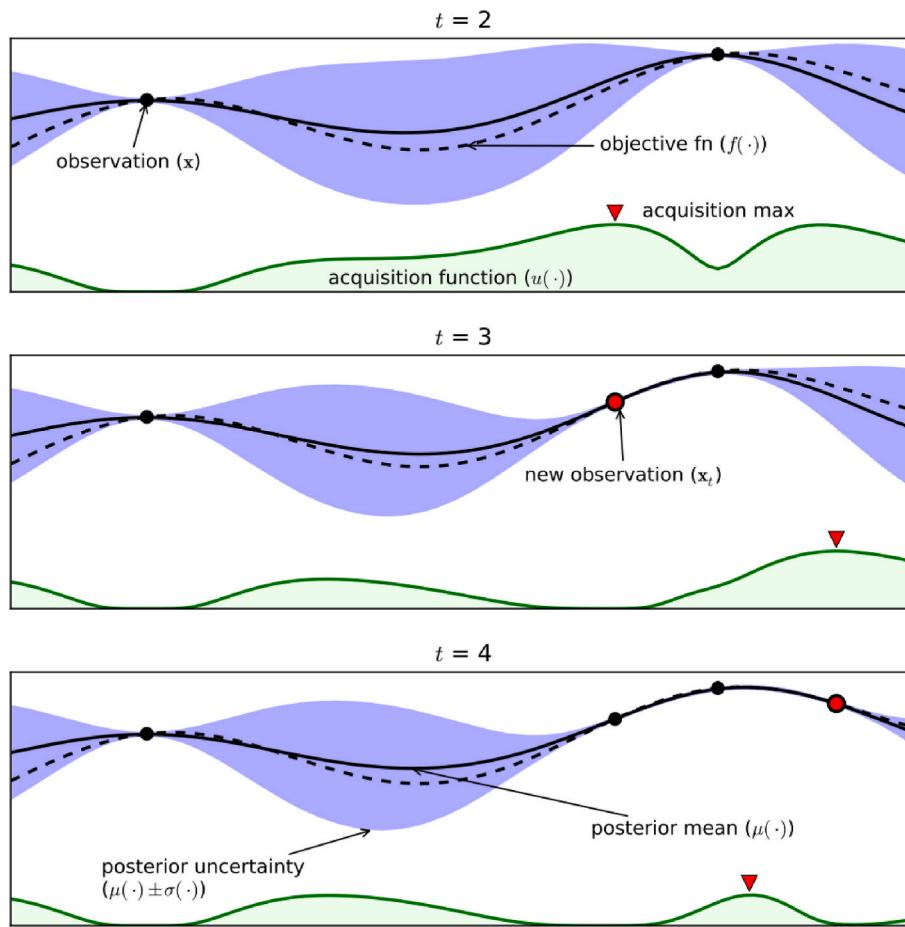


Fig. 2. Illustrating Bayesian optimization: Three iterations.

of the Gradient Boosting Regression Tree, the loss function is $L(y, G_{t-1}(x))$. The goal of this round is to find a weak learner $h_{t-1}(x)$ for the Classification and Regression Tree model so that the loss $L(y, G_{t-1}(x))$ in this iteration is the smallest.

According to Friedman's proposal, a Classification and Regression Tree should be fitted to the loss function by approximating its value with the Full gradient of the function. It can be shown that in the T round, the loss function has a negative gradient for the I sample if:

$$q_{ti} = - \left[\frac{\partial L(y_i, G(x_i))}{\partial, G(x_i)} \right]_{G(x)=G_{t-1}(x)} \quad (4)$$

Using (x_i, q_{it}) ($i = 1, 2, \dots, m$) To get the first regression tree, we may use the Classification and Regression Tree fitting procedure. That part of the tree where the leaves are.

Q_{tj} , $j = 1, 2, \dots, J$.
J: the number of leaf nodes.

The best result number for you and the leaf nodes is c_{tj} for each leaf node sample.

$$c_{tj} = \operatorname{argmin} \sum_{x_i \in Q_{tj}} L(y_i, G_{t-1}(x_i) + c) \quad (5)$$

This allows for the following to be acquired as the fitting function for the present decision tree:

$$h_t(x) = \sum_{j=1}^J c_{tj} I(x \in Q_{tj}) \quad (6)$$

As a result, the following describes the strong learner identified in the most recent round:

$$G_t(x) = G_{t-1}(x) + \sum_{j=1}^J c_{tj} I(x \in Q_{tj}) \quad (7)$$

4.3. Ensemble

ML approaches known as "ensembles" take many individual "base systems" and merge them into a single "best-fit" prediction system. The prediction performance of a regression system may be improved using a variety of ensemble methodologies [30]. The ensemble method is used to outperform the individual basis learners [31] by combining many weak learners.

4.4. CNN-LSTM

The CNN-LSTM model integrates the advantages of both CNN and LSTM when modeling time series to offer local-spatial properties. CNN provides spatial correlation feature vectors of meteorological parameters at different locations, created in time series and used as input data for the LSTM network. This algorithm aims to predict the output of the target wind farm in the following M seconds. The degree to which past values of a given meteorological factor N correlate with those values over a given time series data M. And hence formula 8 is as follows:

$$P(p_t, p_{t+1}, \dots, p_{t+m}, \dots, p_{t+M-1}) = \sum_1^N \times \sum_1^{M-1} (IW(t-m, f_n) \cdot C(IW(t-m, f_n), p_{t+m})) \quad (8)$$

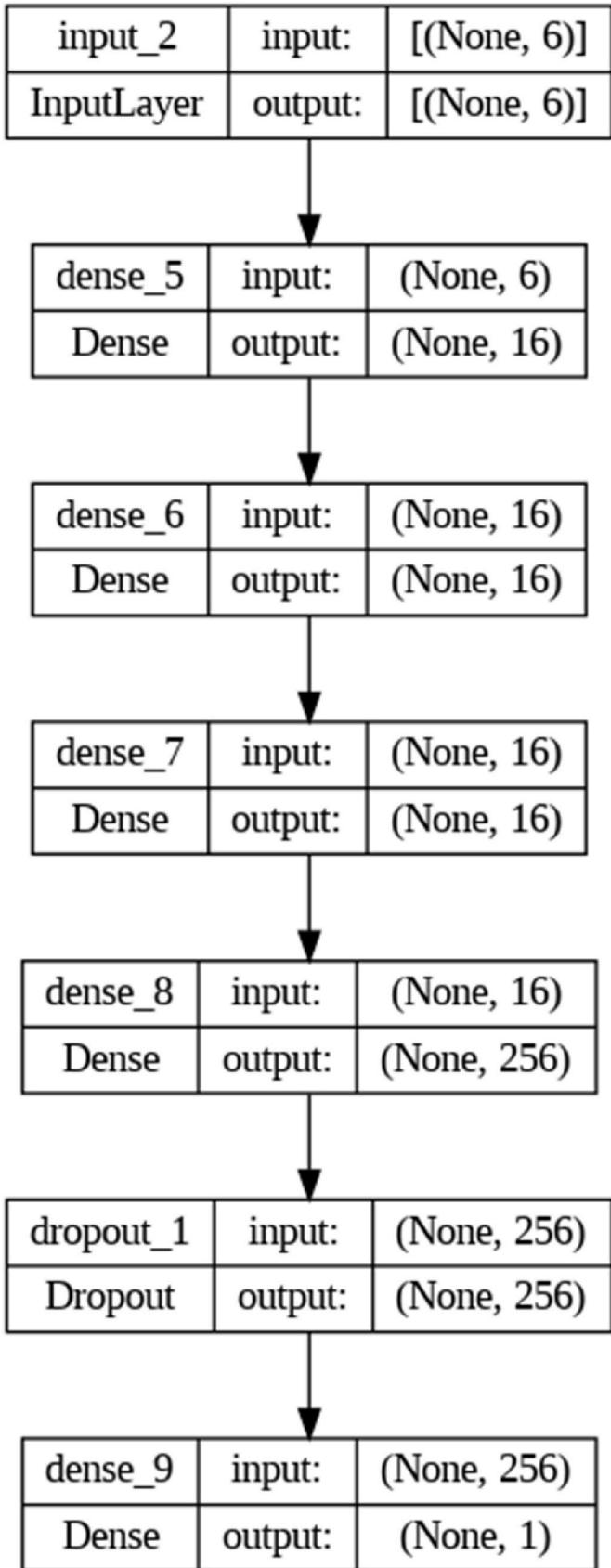


Fig. 3. Illustrating bayesian optimization: Three iterations.

4.5. Neural network hyperparameter tuning with bayesian optimization

The model hyperparameters, optimization parameters, and regularization terms in machine learning algorithms are often subject to meticulous tuning. However, tuning is frequently a "black art" that calls for specialized knowledge, unwritten guidelines, or occasionally brute-force investigation, which is inefficient [32]. Solving an objective function, which is connected to the learning system's hyperparameters, is the primary purpose of the majority of machine learning algorithms. Usually, one assumes that this cost function $f(x)$ is convex, inexpensive to evaluate, or has a recognized mathematical form. However, many learning problems defy these robust presumptions. In most cases, it is costly or difficult to evaluate the objective function, and it is uncertain what the derivatives and convexity features are [33].

Creating automatic methods that maximize (or minimize) a specific learning algorithm's performance for a given task is even more alluring [32]. This section addresses the automated tuning problem in the context of Bayesian optimization, whereby the generalization performance of a neural network is represented as a sample from a Gaussian process. The Gaussian process induces a tractable posterior distribution that facilitates optimum decision-making for the next set of parameters to test by effectively using the data collected by earlier experiments. This has to consider both exploitation (sampling locations likely to give improvement over the present best observation) and exploration (sampling from areas of high uncertainty) [33].

It is named Bayesian because it uses the well-known "Bayes' theorem," which roughly means that the posterior probability of a model, theory, or hypothesis M given evidence, data, or observations E is equal to the likelihood of E given M multiplied by the prior probability of M divided by the marginal likelihood or evidence E .

$$P(M|E) = \frac{P(E|M) \cdot P(M)}{P(E)} \quad (9)$$

This straightforward equation includes the secret to maximizing (or minimizing) the objective function. The prior in Bayesian optimization denotes our perception of the range of potential objective functions. Some conceivable objective functions are more likely than others because, even if the cost function is unknown, it is acceptable to presume that some of its properties have been known before [33].

We are looking to find the hyperparameters of the neural network, x , such that the network's performance index, here mean square error, for testing the dataset is minimal.

$$f(x) = E[e^T e] = E[(t - a)^T(t - a)] \quad (10)$$

$$x^* = \underset{x}{\operatorname{argmin}}\{f(x)\} \quad (11)$$

Where x is the vector of network hyperparameters, t and a are the target and network outputs.

The form of the optimization problem that we are concerned with in an artificial neural network is depreciation, which can be transformed into a maximization problem as follows.

$$f(x) = -g(x) \quad (12)$$

$$x^* = \underset{x}{\operatorname{argmax}}\{g(x)\} \quad (13)$$

In our problem, the objective function ($f(x)$) is unknown and has no close form.

To determine a posterior distribution over the space of functions, Bayesian optimization uses the prior and evidence. Even when the objective function itself is unknown, the Bayesian model provides a sophisticated way for proper priors to characterize the characteristics of the function. Maximizing expected utility—or the least expected risk—is the foundation of optimization. Selecting a utility function and a method

for maximizing its expectation about the objective function's posterior distribution are necessary for determining the following sampling location [33]. Because of this, it is essential that the acquisition functions be cheap to assess or approximate—that is, inexpensive when compared to the cost of analyzing the main objective function, $f(x)$. Optimization of acquisition functions is typically simpler than optimization of the original objective function since they have easily evaluable, analytical forms.

We represent the unknown objective function in Bayesian optimization as a probability distribution. We use this probabilistic model to guide our function exploration and exploitation. Most important is the prior distribution across functions. It stands for our presumptions on the way the objective function behaves. A Gaussian process is the conventional option for the prior (GP). This is how it operates: A Gaussian process is called a set of random variables (functions) indexed by an input space (such as hyperparameters). Every function derived from the GP represents a potential realization of the fundamental process. Gaussian processes (GPs) operate in an infinite-dimensional function space, where each function drawn from the GP represents a possible realization of an underlying stochastic process. The behavior of a GP is determined by its covariance function (kernel), which encodes the similarity between function values at different input positions. Notably, the corresponding function values follow a multivariate normal distribution for any finite set of input points. GPs allow us to calculate the conditional distribution of the function value at an unseen moment, leveraging the observed function values. Moreover, they excel at identifying correlations and uncertainties within the function. The GP is fully discussed in Ref. [34].

Our goal is to determine the objective function's maximum (or minimum, $f(x)$), which is unknown. It is assumed that the function ($f(x)$) originates from a Gaussian process (GP) prior. Our observations consist of pairs $\{x_n, y_n\}$, where the variance of noise in the data is represented by ν and $y_n \sim N(f(x_n), \nu)$. Observed data and the GP prior give us a posterior distribution over functions. The following evaluation point, x_{next} , is chosen under the guidance of the acquisition function $\alpha : X \rightarrow R^+$. The acquisition function, $x_{\text{next}} = \underbrace{\arg\max_x}_{x} \{\alpha(x)\}$, is what we wish to

maximize. The following factors affect the acquisition function: Previous observations $\{x_n, y_n\}$, and hyperparameters for GP (θ). This dependency is represented by the notation $\alpha(x; \{x_n, y_n\}, \theta)$. These functions rely on the predictive variance function $\sigma^2(x; \{x_n, y_n\}, \theta)$ and predictive mean function $\mu(x; \{x_n, y_n\}, \theta)$ of the GP under the Gaussian process prior. Among the favored options are:

Expected Improvement (EI): This approach strikes a balance between sampling doubtful locations for exploration and promising regions for exploitation. It calculates the expected improvement by assessing at a specific point in time.

Upper Confidence Bound (UCB): Takes into account the GP's mean and uncertainty (variance) to balance exploration and exploitation. High uncertainty sampling points are encouraged.

Other variations exist, such as Thompson Sampling and Probability of Improvement (PI). Acquisition functions are fully discussed in Ref. [35].

The process of Bayesian optimization begins with an initial set of function evaluations, which are typically selected randomly. These observations are used to update the GP prior. The next point to be evaluated is chosen based on the acquisition function. It strikes a compromise between exploitation (sampling where the function is probably optimum) and exploration (sampling in ambiguous regions). We update the GP, evaluate the function at the selected point, and recursively go through the procedure again.

A typical run of Bayesian optimization on an issue is depicted in the accompanying figure. There are two points in the optimization. The acquisition function, which models the utility of sampling by considering the mean and variance of the predictions throughout the space, is

maximized at each iteration to decide where to sample next from the objective function. After sampling the objective at the acquisition function's argmax, the Gaussian process is updated and repeated. This stage of Bayesian optimization can also be seen as estimating the objective function using the posterior mean function of a Gaussian process as a surrogate function, sometimes known as a response surface.

This figure showcases the Bayesian optimization process across three iterations. The plots depict the mean and confidence intervals, estimated using a probabilistic objective function model. While the actual objective function remains unknown, its behavior is visualized. Additionally, the acquisition functions are displayed in the lower-shaded plots. High acquisition values occur where the model predicts a favorable objective (exploitation) and where prediction uncertainty is significant (exploration). Notably, the leftmost region is not sampled because of its great uncertainty, accurately reflecting little progress over the highest observation [33].

A collection of parameters known as hyperparameters has values that remain constant on training and testing datasets throughout the learning process. Examples of hyperparameters are activation functions in each layer, batch size, learning rate, number of epochs, hidden layers, number of neurons in each layer, momentum, regularization, dropout rate, etc.

The optimal hyperparameters must be selected throughout the training process to effectively train the model, which can significantly affect it. The parameters should be combined to minimize the loss function and maximize the accuracy or performance of the model. Thus, it is possible to describe the tuning of hyperparameters as an optimization problem [37].

A distinctive aspect of the Bayesian optimization strategy that makes it suitable for tuning the hyperparameters of the neural network is that it seeks to reduce the number of times the objective function is evaluated. Bayesian optimization in this study yields optimum values for every hyperparameter, saving time and enhancing performance.

5. Evaluation processes

5.1. Data preparation

The time series in this article were created using NREL's Renewable Energy Program (RES) software. The usage data includes wind speed in meters per second, wind direction in degrees, air pressure in atmospheres, and temperature in degrees Celsius. Each column's data was captured every hour for a year.

5.2. Evaluation process

Seventy-five percent of the data is used in the 10-fold cross-validation [40–56] to train the model and another portion to test it. The prediction error from Equation One is then estimated using cross-validation as follows:

$$CV(h) = \frac{1}{m} \sum_{i=1}^m L(y_i, h^{-S(i)}(x_i)) \quad (14)$$

S: the number of subgroups

m: the amount of the dataset

L: the loss function

$h^{-S(i)}$: the fitted function

The Grid Search CV used in this research, XGBOOST, and Ensemble, were used to tune hyperparameters in a light gradient boosting machine.

In the placing layer, input features are selected from variables deemed necessary by correlation analysis. There will be 32 channels in the convolution, and the convolutional kernel size will be 1. It is hypothesized that the further one goes into the fitting, the deeper one's

LSTM layer becomes hidden. The LSTM layers are capped at two because adding more layers to the architecture raises the training time. There are 64 neurons in the LSTM's initial cell and another 64 in its second. The AM layer emphasizes the impact on output by learning the feature weight and creating the AM layer's input vector.

In this study, we aim to design a multi-layer neural network model using the "Texas Turbine dataset" to predict the power generated by turbines with minimum error on the validation dataset. The hyperparameters we need to find the optimal value of are as follows.

- The number of hidden layers. The number of hidden layers in an MLP significantly impacts its learning capacity and generalization ability. As the network depth increases, it can capture more intricate and abstract representations of input data. This concept of hierarchical representation allows each hidden layer to learn a set of features that describe the input and propagate that information across subsequent layers. Deep networks, with multiple hidden layers, can learn complex non-linear relationships between input and output data, making them suitable for tasks such as image recognition, natural language processing, and time series prediction. However, choosing the optimal number of hidden layers is not straightforward. Too few hidden layers may limit the network's expressive power, leading to underfitting, while too many layers can result in overfitting. Researchers must strike a balance by considering the complexity of the problem, available data, and computational resources. Researchers can customize MLP architectures to particular tasks by varying the number of hidden layers and nodes, which allows them to balance model complexity and performance. Set {2, 3, 4, 5, 6} contains the number of hidden layers investigated in this study.
- Number of neurons in each hidden layer. Designing multilayer neural networks involves a critical decision: determining the appropriate number of neurons. An excessive number of neurons leads to overfitting on the training data, resulting in poor performance on new, unseen data. A well-generalizing network performs consistently on both training and new data. The network's complexity hinges on its free parameters (weights and biases), which directly relate to the number of neurons. However, adjusting complexity need not involve changing the neuron count. Instead, we can fine-tune the effective number of free parameters while maintaining the actual count. This approach ensures better alignment between network complexity and the data's inherent complexity [6]. For each layer, we explore the hyperparameter domain of the set {8, 16, 32, 64, 128, 256, 512}.
- Activation function in each hidden layer. Activation functions are pivotal in defining the behavior of artificial neurons within neural networks. Their introduction of non-linearity enables networks to effectively capture intricate patterns within data sets. Neural networks can learn complex patterns and adjust to tasks because of their flexibility. Care must be taken, though, as neurons may go dormant and become what researchers call "dead neurons" if an activation function outputs zero for all inputs (as seen in ReLU for negative inputs). Activation function selection ultimately greatly impacts how well the network can represent complex functions [8]. 'Tanh,' 'sigmoid,' and 'ReLU' are this model's study domains for activation functions.
- Regularization. In neural networks, regularization methods are crucial in controlling model complexity. Specifically, we augment the performance index by introducing a penalty term that discourages overly complex networks. This regularization term involves derivatives of the neural network. Incorporating this penalty encourages smoother functions, ultimately enhancing the network's ability to generalize to unseen data [9]. The explored domain for regularization values is [1e-14, 1e-10].
- Initial learning rate and momentum. We can consider adjusting the learning rate during training to enhance convergence speed. The challenge lies in determining when and how much to change the

learning rate based on the surface slope. In neural network training, initialization and momentum are essential components. A particular kind of slowly growing schedule for the momentum parameter and a well-designed random initialization is crucial when utilizing the Adam optimizer (or SGD). Due to these features, it is possible to train deep neural networks (DNNs) on datasets containing long-term dependencies, leading to performance levels previously only possible with more complex optimization techniques. Momentum is a problem for poorly initialized networks because when momentum is missing or not adjusted properly, well-initialized networks falter. The explored hyperparameter domains are [1e-6, 1e-3] for the initial learning rate and [0.4, 0.7] for momentum.

- Dropout rate. In deep neural networks, dropout is a regularization technique frequently employed. A portion of the neurons in each layer are randomly deactivated (dropped out) by dropout during training. This keeps the network from overfitting by encouraging it to rely on a wide range of features rather than being unduly specialized to a few neurons. Enhancing generalization performance on unseen data, dropout effectively approximates training many neural networks with different topologies in parallel. The dropout rate determines the likelihood of neurons dropping out during training. We determine the optimal value for this hyperparameter from the set {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3}. The table below illustrates the optimized hyperparameter values achieved through Bayesian optimization:

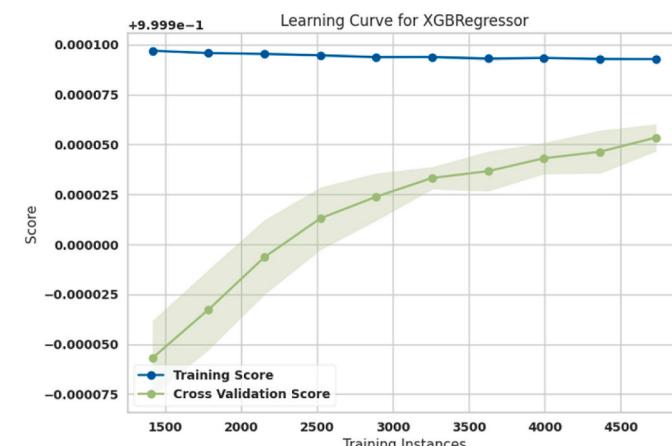
In our study, we employed the Adam optimizer and trained the model using a eight-batch size for 150 epochs. Additionally, we implemented an early stopping technique that monitored the mean square error of the validation dataset, with a patience value set to 20. The model structure is depicted in the following figure.

6. Results

This figure illustrates the training and validation accuracy of the XGBoost algorithm over multiple epochs. The XGBoost model reached approximately 99 % training accuracy and 98.4 % validation accuracy after training with 4750 samples. The learning curve shows how the model's accuracy improves over time, indicating effective learning and generalization capabilities.

[Fig. 5](#) presents the training and validation accuracy of the Gradient Boosting algorithm (see [Fig. 4](#)). The model achieved around 99.994 % training accuracy and 99.992 % validation accuracy after training with the same number of samples. This high level of accuracy demonstrates the model's ability to predict wind power generation accurately (see [Fig. 6](#)).

This figure displays the performance of the Ensemble method, which



[Fig. 4](#). Learning curve for xgboost for predicting power generation.

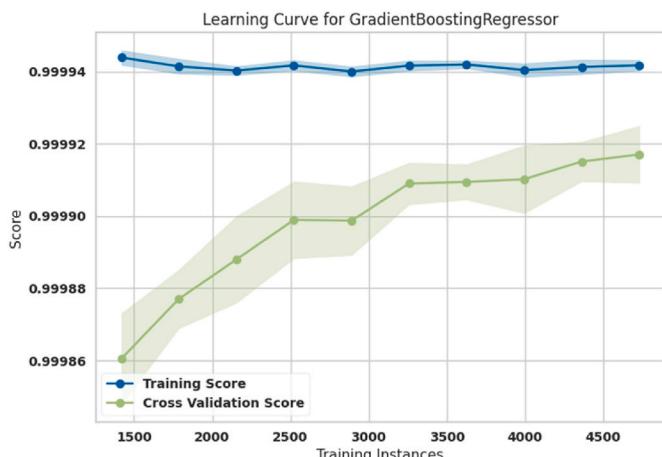


Fig. 5. Learning curve for gradient boosting for predicting power generation.

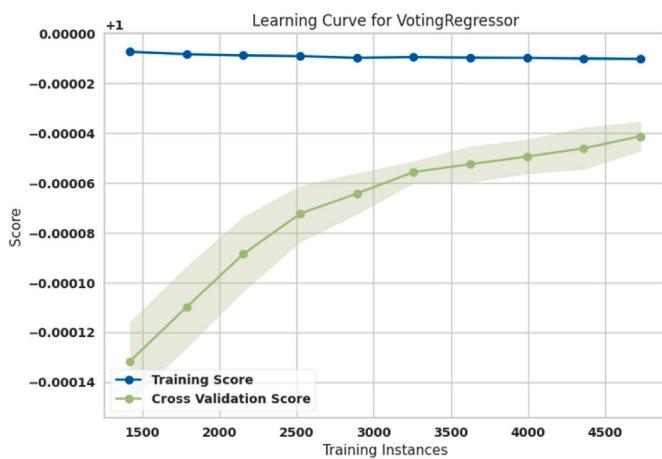


Fig. 6. Learning curve for Ensemble (gradient boosting and xgboost) for predicting power generation.

combines Gradient Boosting and XGBoost. The model showed significant improvement, reaching approximately 99.99 % training accuracy and 99.9996 % validation accuracy. The ensemble approach effectively leverages the strengths of both models to enhance predictive accuracy.

The CNN-LSTM approach was executed using three units of short-term memory and five superficial layers of neurons (Dense). For network training, 80 % of the data was selected, and 200 training steps were repeated while optimizing the optimizer. The optimization function used was Exponential Decay. Fig. 7 illustrates the training loss of the CNN-LSTM model across these 200 training steps. The loss value continuously decreased, reaching an appropriate value of 6.65 after 200 training stages, indicating that the network has been adequately trained and is minimizing error effectively.

The diagrams below (Figs. 8–10) illustrate the MLP + BO's performance during each epoch, evaluated using different metrics on both training and validation datasets.

This figure shows the Mean Absolute Error (MAE) of the MLP + BO method over different epochs for both training and validation datasets. MAE is a measure of the average magnitude of errors in predictions, without considering their direction. A lower MAE indicates better predictive performance, showing the model's precision in predicting wind power generation.

Fig. 9 depicts the Mean Absolute Percentage Error (MAPE) for the MLP + BO method. MAPE measures the accuracy of a forecasting method as a percentage, providing insight into the relative prediction error. Lower MAPE values indicate higher accuracy and better model

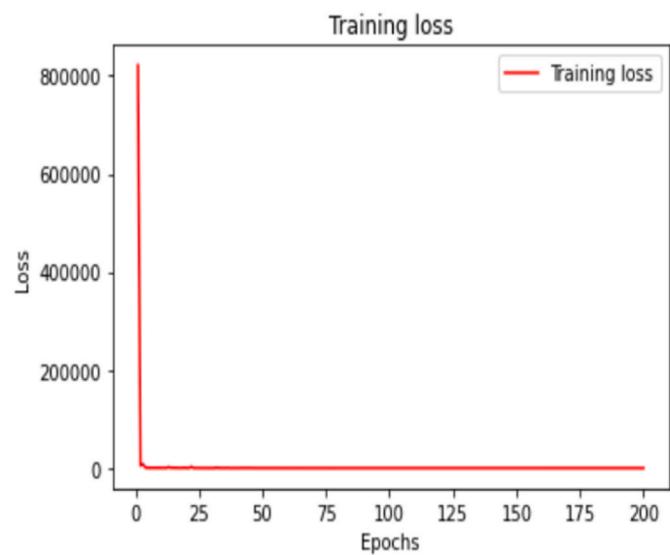


Fig. 7. Training loss for CNN- LSTM method for predicting power generation.

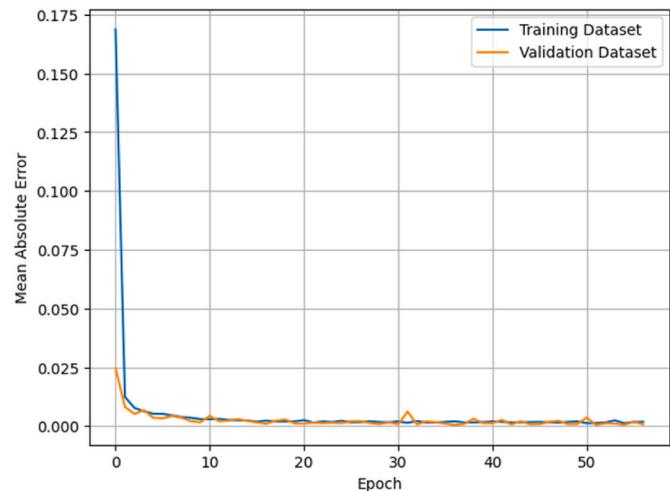


Fig. 8. Mean Absolute Error for MLP + BO method for predicting power generation.

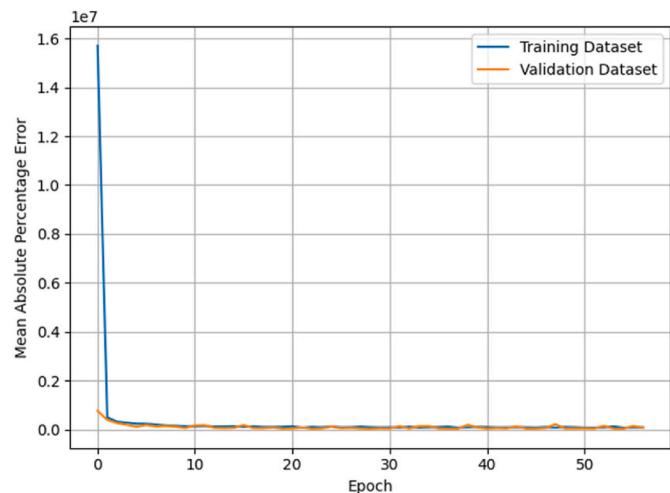


Fig. 9. Mean Absolute Percentage Error for MLP + BO method for predicting power generation.

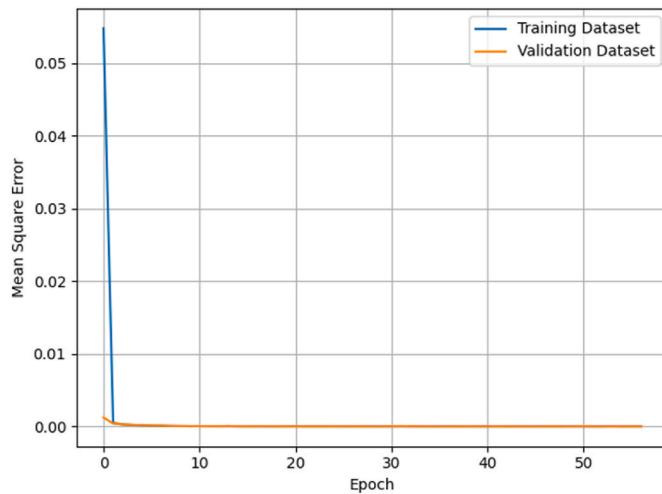


Fig. 10. Mean Square Error for MLP + BO method for predicting power generation.

performance.

This figure shows the Mean Square Error (MSE) for the MLP + BO method. MSE is a common metric for regression models, measuring the average of the squares of the errors. It is particularly sensitive to large errors, making it useful for understanding the model's performance. Lower MSE values indicate better accuracy and predictive performance.

Fig. 11 displays a comparison between the actual system power generated and the model-predicted power generated, both measured in kilowatts (kW). The data table contains 1752 rows, each representing a specific time interval with corresponding actual and predicted power values.

For many entries, the predicted values are very close to the actual values, indicating high accuracy of the model. For example:

At time interval 0, the actual power generated is 1013.613037 kW, while the predicted power is 1016.6300 kW.

At time interval 1, the actual power generated is 298.986389 kW, and the predicted power is 298.9770 kW.

Some entries show minor deviations between the actual and predicted values, which is expected in real-world predictive modeling. For example:

At time interval 1747, the actual power generated is 722.360046 kW, while the predicted power is 716.0110 kW.

There are a few instances where the deviations are more pronounced, which may indicate areas where the model could be improved. For example:

At time interval 1748, the actual power generated is 2798.306641 kW, while the predicted power is 2805.5300 kW.

The purpose of Fig. 11 is to visually validate the model's performance by comparing the actual and predicted power generation values. The close alignment of the two sets of values demonstrates the model's effectiveness and accuracy in predicting wind power generation.

Equations (15)–(18) were used to determine how well the methods worked and choose the best one based on evaluation indices like the coefficient of MSE and the RMSE (see Table 1).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (o_i - p_i)^2}{n}} \quad (15)$$

$$MSE = \frac{\sum_{i=1}^n (o_i - p_i)^2}{n} \quad (16)$$

$$MAE = \frac{\sum_{i=1}^n |o_i - p_i|}{n} \quad (17)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{o_i - p_i}{o_i} \right| \quad (18)$$

Where.

n : number of test data.

o_i : actual data.

p_i : predicted data.

Table 2 shows the root mean square error and mean square error findings for the testing dataset values for wind power forecasts.

In articles [42,46], the data analyzed in this article were used, and in Ref. [42], the best performance was the CNN-LSTM algorithm with MSE = 8.2 in predicting production power, and in Ref. [46], the best performance was the Random Forest algorithm with MSE = 21.38. It had productive power in forecasting, while in this article, the performance of the proposed Ensemble (gradient boosting and xgboost) and CNN-LSTM methods have been improved.

The study investigated the use of advanced machine learning techniques for wind power prediction, utilizing methods such as XGBoost, Multi-Layer Perceptron with Bayesian Optimization (MLP + BO), Gradient Boosting Regression Tree (GBDT), Ensemble Learning

	The Real System Power Generated (kW)	The Model Predicted System Power Generated (kW)
0	1013.613037	1016.6300
1	298.986389	298.9770
2	583.303223	582.7210
3	1157.467529	1155.0100
4	53.611443	54.2094
...
1747	722.360046	716.0110
1748	2798.306641	2805.5300
1749	203.886520	202.9040
1750	187.802185	188.7340
1751	2391.146240	2386.0700
1752 rows × 2 columns		

Fig. 11. Actual system power generated in kilowatts (kW) with the predictions from our designed model using Bayesian optimization (BO).

S.M. Malakouti et al.

Case Studies in Chemical and Environmental Engineering 10 (2024) 100881

Table 1
Optimized hyperparameter for Bayesian optimization.

Hyperparameter	Optimize value (BO)
Number of Hidden Layers	4
Neurons of 1st Hidden Layer	16
Neurons of 2nd Hidden Layer	16
Neurons of 3rd Hidden Layer	16
Neurons of the 4th Hidden Layer	256
Activation Function	ReLU
Regularization	4.66e-11
Initial Learning Rate	2.85e-4
Momentum	0.6346
Drop-out Rate	0

Table 2
The RMSE, MAE, MAPE, and MSE measures evaluated model performance.

method	MAE (kw)	MSE (kw)	RMSE (kw)	Time (sec)
CNN-LSTM	1.55	6.8	2.607	450
MLP + BO	1.47	5.43	2.33	200
ENSEMBLE	1.65	7.2	2.683	45
XGBOOST	1.82	12.22	3.495	8.78
GBR	2.12	14.75	3.84	6.120

(combining gradient boosting and XGBoost), and CNN Long Short-Term Memory (CNN-LSTM). The primary findings indicate that these methods significantly enhance the accuracy of wind power forecasts, with Ensemble Learning and CNN-LSTM showing notable performance improvements.

The primary finding of the study is that the hybrid approaches combining different machine learning models provide superior forecasting accuracy compared to individual models. Specifically, the Ensemble Learning method achieved an RMSE of 2.683 kW in 45 seconds, while the CNN-LSTM method achieved an RMSE of 2.607 kW in 450 seconds (Table 2). These results demonstrate the effectiveness of integrating multiple algorithms to handle the non-linear and complex nature of wind patterns.

The improved performance of the ensemble and hybrid models can be attributed to their ability to leverage the strengths of different algorithms. Ensemble Learning, for instance, combines the predictions of multiple models to reduce variance and bias, leading to more robust and accurate predictions. The CNN-LSTM model, on the other hand, benefits from the convolutional neural network's ability to extract spatial features and the LSTM's ability to capture temporal dependencies. This combination allows for better handling of the intricate dynamics in wind speed and power generation data.

The findings of this study are consistent with previous research. For example, Wang et al. [3] demonstrated that combining Bayesian model averaging with ensemble learning improved wind power forecasting accuracy. Similarly, Liu et al. [21] showed that convolutional neural networks could significantly enhance wind turbine power forecasts by capturing complex patterns in meteorological data. However, some studies have noted limitations with individual models, such as the Random Forest method, which showed an MSE of 21.38 in other contexts. These comparisons highlight the relative advantages of the hybrid approaches used in this study.

By comparing our findings with other studies, we observe both consistencies and discrepancies. For instance, Dimitropoulos et al. [19] reported an RMSE of 1.834 kWh for 1-h interval predictions using XGBoost, which is slightly more accurate than our results. However, their study focused on solar energy prediction, which may present different challenges compared to wind energy. The hybrid models developed in this study offer significant improvements in wind power forecasting, contributing to the broader theoretical framework of renewable energy management. These methods not only enhance predictive accuracy but also support more reliable integration of wind

energy into power grids, addressing the critical issue of variability and intermittency in renewable energy sources. Future research could further explore the integration of other advanced techniques and real-time data to continuously improve forecasting models.

7. Conclusions

This research contributes to the theoretical framework of renewable energy management by integrating multiple advanced machine learning techniques to improve wind energy forecasting. The hybrid models combining XGBOOST, MLP + BO, GBDT, Ensemble Learning, and CNN-LSTM offer a robust approach to capturing the non-linear and complex nature of wind patterns. This theoretical advancement demonstrates the potential of ensemble methods in achieving superior forecasting performance, extending the body of literature on machine learning applications in renewable energy.

Furthermore, the use of Bayesian Optimization for hyperparameter tuning provides a methodological contribution by efficiently navigating the hyperparameter space to enhance model performance. The integration of CNNs with LSTM networks, combining spatial feature extraction with temporal sequence learning, represents a novel approach well-suited for time-series prediction tasks. These methodological innovations contribute to the theoretical understanding of optimizing machine learning models for complex predictive tasks.

The practical implications of this research are significant for the renewable energy sector. Improved accuracy in wind energy forecasts supports more reliable and efficient integration of wind power into energy grids, addressing a critical challenge in managing the variability and intermittency of wind energy. This can lead to better energy management strategies, reduced dependency on fossil fuels, and enhanced grid stability, supporting the transition to sustainable energy systems.

The research highlights the importance of model diversity and the effectiveness of ensemble techniques in improving predictive accuracy. The ensemble approach, which aggregates the strengths of individual algorithms, consistently outperformed single models, emphasizing the value of integrating multiple methods to capture complex data patterns.

The use of Bayesian Optimization for hyperparameter tuning significantly enhanced model performance, demonstrating that efficient hyperparameter optimization is crucial for maximizing the potential of machine learning models in predictive tasks.

The integration of CNNs with LSTM networks showed that combining spatial feature extraction with temporal sequence learning can effectively address the challenges of time-series prediction. This approach can be applied to other domains requiring accurate time-series forecasts.

CRediT authorship contribution statement

Seyed Matin Malakouti: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

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

- [1] B. Kosovic, et al., A comprehensive wind power forecasting system integrating artificial intelligence and numerical weather prediction, *Energies* 13 (6) (2020) 1372, <https://doi.org/10.3390/en13061372> (MDPI) (Directory of Open Access Journals – DOAJ).
- [2] T. Liu, Z. Huang, L. Tian, Y. Zhu, H. Wang, S. Feng, Enhancing wind turbine power forecast via convolutional neural network, *Electronics* 10 (3) (2021) 261, <https://doi.org/10.3390/electronics10030261> (MDPI) (MDPI).
- [3] G. Wang, R. Jia, J. Liu, H. Zhang, A hybrid wind power forecasting approach based on Bayesian model averaging and ensemble learning, *Renew. Energy* 145 (2020) 2426–2434, <https://doi.org/10.1016/j.renene.2019.08.053> (SpringerLink).
- [4] S. Hanifi, X. Liu, Z. Lin, S. Lotfian, A critical review of wind power forecasting methods—past, present and future, *Energies* 13 (15) (2020) 3764, <https://doi.org/10.3390/en13153764> (Directory of Open Access Journals – DOAJ) (MDPI).
- [5] M.A. Abdelzaher, M.M. Awad, Sustainable development goals for the circular economy and the water-food nexus: Full implementation of new drip irrigation technologies in upper Egypt, *Sustainability* 14 (21) (2022) 13883, <https://doi.org/10.3390/su142113883>.
- [6] A. Tascikaraoglu, M. Uzunoglu, A review of combined approaches for prediction of short-term wind speed and power, *Renew. Sustain. Energy Rev.* 34 (2014) 243–254, <https://doi.org/10.1016/j.rser.2014.03.033> (SpringerLink).
- [7] D. Li, Z. Zhang, X. Zhou, Z. Zhang, X. Yang, Cross-wind dynamic response of concrete-filled double-skin wind turbine towers: theoretical modelling and experimental investigation, *J. Vib. Control* (2023) 1–13, <https://doi.org/10.1177/10775463231157791>. RSC Publishing (American Institute of Physics).
- [8] F. Cassola, M. Burlando, Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output, *Appl. Energy* 99 (2012) 154–166, <https://doi.org/10.1016/j.apenergy.2012.03.054> (Sci-Hub).
- [9] H. Liu, J. Shi, E. Erdem, Prediction of wind speed time series using modified Taylor Kriging method, *Energy* 35 (2010) 4870–4879, <https://doi.org/10.1016/j.energy.2010.09.012> (Sci-Hub).
- [10] C. González-Minguez, F. Muñoz-Gutiérrez, Wind prediction using Weather Research Forecasting model (WRF): a case study in Peru, *Energy Convers. Manag.* 81 (2014) 363–373, <https://doi.org/10.1016/j.enconman.2014.02.024>. Sci-Hub.
- [11] H. Esen, F. Ozgen, M. Esen, A. Sengur, Modelling of a new solar air heater through least-squares support vector machines, *Expert Syst. Appl.* 36 (2009) 10673–10682, <https://doi.org/10.1016/j.eswa.2009.02.045>.
- [12] C. Ren, N. An, J. Wang, L. Li, B. Hu, D. Shang, Optimal parameters selection for BP neural network based on particle swarm optimization: a case study of wind speed forecasting, *Knowl.-Based Syst.* 56 (2014) 226–239, <https://doi.org/10.1016/j.knosys.2013.11.015>.
- [13] G. Wang, T. Fearn, T. Wang, K.L. Choy, Machine-learning approach for predicting the discharging capacities of doped lithium nickel-cobalt-manganese cathode materials in Li-ion batteries, *ACS Cent. Sci.* 7 (2021) 1551–1560, <https://doi.org/10.1021/acscentsci.1c00611>.
- [14] J.V. Tu, Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, *J. Clin. Epidemiol.* 49 (1996) 1225–1231, [https://doi.org/10.1016/S0895-4356\(96\)00002-9](https://doi.org/10.1016/S0895-4356(96)00002-9).
- [15] M.K. Tran, S. Panchal, V. Chauhan, N. Brahmabhatt, A. Mevawalla, R. Fraser, M. Fowler, Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery, *Int. J. Energy Res.* 46 (2022) 786–794, <https://doi.org/10.1002/er.7202>.
- [16] A. Sekulić, M. Kilibarda, G.B.M. Heuvelink, M. Nikolić, B. Bajat, Random Forest spatial interpolation, *Rem. Sens.* 12 (2020) 1687, <https://doi.org/10.3390/rs12101687>.
- [17] T. Chen, T. He, M. Benesty, *R Package, Version 0.71-2, XGBoost: eXtreme Gradient Boosting*, R Core Team, Vienna, Austria, 2018, pp. 1–3.
- [18] Z.D. Grève, J. Bottieau, D. Vangulick, A. Wautier, P.D. Dapoz, A. Arrigo, J. F. Toubeau, F. Vallée, Machine learning techniques for improving self-consumption in renewable energy communities, *Energies* 13 (2020) 4892, <https://doi.org/10.3390/en13184892> (MDPI) (MDPI).
- [19] N. Dimitropoulos, N. Sofias, P. Kapsalis, Z. Mylona, V. Marinakis, N. Primo, H. Doukas, Forecasting of short-term PV production in energy communities through Machine Learning and Deep Learning algorithms, in: Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 2021, pp. 1–6, <https://doi.org/10.1109/IISA52424.2021.9555544> (Reviewerly) (OUCI).
- [20] A. Baba, Advanced AI-based techniques to predict daily energy consumption: a case study, *Expert Syst. Appl.* 184 (2021) 115508, <https://doi.org/10.1016/j.eswa.2021.115508>.
- [21] H. Musbah, G. Ali, H.H. Aly, T.A. Little, Energy management using multi-criteria decision making and machine learning classification algorithms for intelligent system, *Elec. Power Syst. Res.* 203 (2022) 107645, <https://doi.org/10.1016/j.epres.2021.107645> (X-MOL).
- [22] A. Jozi, T. Pinto, I. Praça, F. Silva, B. Teixeira, Z. Vale, Energy consumption forecasting using genetic fuzzy rule-based systems based on MOGUL learning methodology, in: Proceedings of the 2017 IEEE Manchester PowerTech, 18–22 June 2017, pp. 1–5, <https://doi.org/10.1109/PTC.2017.7981219>. Manchester, UK.
- [23] A. Jozi, T. Pinto, Z. Vale, Contextual learning for energy forecasting in buildings, *Int. J. Electr. Power Energy Syst.* 136 (2022) 107707, <https://doi.org/10.1016/j.ijepes.2021.107707>.
- [24] L.X. Wang, J. Mendel, Generating fuzzy rules by learning from examples, *IEEE Trans. Syst. Man Cybern.* 22 (1992) 1414–1427, <https://doi.org/10.1109/21.199466>.
- [25] O. Cordón, M.J. del Jesus, F. Herrera, M. Lozano, MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach, *Int. J. Intell. Syst.* 14 (1999) 1123–1153, [https://doi.org/10.1002/\(SICI\)1098-111X\(199911\)14:11<1123::AID-INT4>3.0.CO;2-6](https://doi.org/10.1002/(SICI)1098-111X(199911)14:11<1123::AID-INT4>3.0.CO;2-6).
- [26] F. Al-Shanableh, A. Evcil, Prediction of energy consumption of residential buildings in northern Cyprus using fuzzy interference system, *Energy Build.* 256 (2022) 111555, <https://doi.org/10.1016/j.enbuild.2021.111555>.
- [27] A. Prasad, I. Dusparic, Multi-agent deep reinforcement learning for zero energy communities, in: Proceedings of the 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 29 September–2 October 2019, pp. 1–5, <https://doi.org/10.1109/ISGT-Europe.2019.8905628>. Bucharest, Romania.
- [28] J. Fan, X. Wang, L. Wu, H. Zhou, F. Zhang, X. Yu, X. Lu, Y. Xiang, Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: a case study in China, *Energy Convers. Manag.* 164 (2018) 102–111, <https://doi.org/10.1016/j.enconman.2018.02.204>.
- [29] I. Ullah, K. Liu, T. Yamamoto, R.E. Al Mamlook, A. Jamal, A comparative performance of machine learning algorithm to predict electric vehicles energy consumption: a path towards sustainability *Energy & Environment* (2021) 0958305X211044998, <https://doi.org/10.1177/0958305X211044998>.
- [30] L. Rokach, Ensemble-based classifiers, *Artif. Intell. Rev.* 33 (1) (2010) 1–39, <https://doi.org/10.1007/s10462-009-9124-7> (Sci-Hub). (SpringerLink).
- [31] A. Zameer, J. Arshad, A. Khan, M.A.Z. Raja, Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks, *Energy Convers. Manag.* 134 (2017) 361–372, <https://doi.org/10.1016/j.enconman.2016.12.032> (Sci-Hub).
- [32] Jasper Snoek, Hugo Larochelle, Ryan P. Adams, Practical bayesian optimization of machine learning algorithms, *Adv. Neural Inf. Process. Syst.* 25 (2012), <https://doi.org/10.5555/2999325.2999464> (NeurIPS).
- [33] Eric Brochu, Vlad M. Cora, Nando De Freitas, *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*, arXiv preprint arXiv:1012.2599 (2010) arXiv: 1012.2599.
- [34] Carl E. Rasmussen, Christopher Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [35] James Wilson, Frank Hutter, Marc Deisenroth, Maximizing acquisition functions for Bayesian optimization, *Adv. Neural Inf. Process. Syst.* 31 (2018), <https://doi.org/10.5555/3326943.3327126> (ar5iv) (Researcher) (NeurIPS Proceedings).
- [36] Neural Network Design" by Martin T. Hagan, Howard B. Demuth, and Mark H. Beale.
- [37] A.H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization, *Evolving Systems* 12 (2021) 217–223, <https://doi.org/10.1007/s12530-020-09345-2>.
- [38] Andrea Apicella, Francesco Donnarumma, Francesco Isgòr, Roberto Prevete, A survey on modern trainable activation functions, *Neural Network*. 138 (2021) 14–32, <https://doi.org/10.1016/j.neunet.2021.01.026>.
- [39] A.N. Tikhonov, The solution of ill-posed problems and the regularization method, *Dokl. Acad. Nauk USSR* 151 (3) (1963) 501–504.
- [40] I. Rabichi, C. Sekkouri, F.E. Yaacoubi, K. Ennaciri, Z. Izghri, T. Bouzid, L. El Fels, A. Baqaoui, A. Yaacoubi, Experimental and theoretical investigation of olive mill solid waste biochar for vanillin acid adsorption using DFT/B3LYP analysis, *Water, Air, Soil Pollut.* 235 (6) (2024 Jun) 369, <https://doi.org/10.1007/s11270-024-07183-5>.
- [41] K. Meftah, S. Meftah, H. Lamkhanter, T. Bouzid, Y. Rezzak, S. Touil, A. Abid, Extraction and optimization of *Austrocylindropuntia subulata* powder as a novel green coagulant, *Desalination Water Treat.* 318 (2024 Apr 1) 100339, <https://doi.org/10.5004/dwt.2024.100339>.
- [42] A. Naboulsi, T. Bouzid, A. Grich, A. Regti, M. El Himri, M. El Haddad, Understanding the column and batch adsorption mechanism of pesticide 2, 4, 5-T utilizing alginate-biomass hydrogel capsule: a computational and economic investigation, *Int. J. Biol. Macromol.* 8 (2024 Jul) 133762, <https://doi.org/10.1016/j.ijbiomac.2024.133762> (ORCID).
- [43] S.M. Malakouti, A.R. Ghiasi, A.A. Ghavifekr, P. Emami, Predicting wind power generation using machine learning and CNN-LSTM approaches, *Wind Eng.* 46 (6) (2022 Dec) 1853–1869, <https://doi.org/10.1177/0309524X221113013>.
- [44] S.M. Malakouti, Estimating the output power and wind speed with ML methods: a case study in Texas, *Case Studies in Chemical and Environmental Engineering* 7 (2023 Jun 1) 100324, <https://doi.org/10.1016/j.cscee.2023.100324>.
- [45] S.M. Malakouti, Heart disease classification based on ECG using machine learning models, *Biomed. Signal Process Control* 84 (2023 Jul 1) 104796, <https://doi.org/10.1016/j.bspc.2023.104796>.
- [46] S.M. Malakouti, Utilizing time series data from 1961 to 2019 recorded around the world and machine learning to create a Global Temperature Change Prediction Model, *Case Studies in Chemical and Environmental Engineering* 7 (2023 Jun 1) 100312, <https://doi.org/10.1016/j.cscee.2023.100312>.
- [47] S.M. Malakouti, Improving the prediction of wind speed and power production of SCADA system with ensemble method and 10-fold cross-validation, *Case Studies in Chemical and Environmental Engineering* 8 (2023 Dec 1) 100351, <https://doi.org/10.1016/j.cscee.2023.100351>.
- [48] S.M. Malakouti, M.B. Menhaj, A.A. Suratgar, The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction, *Cleaner Engineering and Technology* 15 (2023 Aug 1) 100664, <https://doi.org/10.1016/j.clet.2023.100664> (OUCL).
- [49] S.M. Malakouti, A.R. Ghiasi, A.A. Ghavifekr, AERO2022-flying danger reduction for quadcopters by using machine learning to estimate current, voltage, and flight

S.M. Malakouti et al.

- area, e-Prime-Advances in Electrical Engineering, Electronics and Energy 2 (2022 Jan 1) 100084, <https://doi.org/10.1016/j.prime.2022.100084>. (OUCI).
- [50] S.M. Malakouti, Use machine learning algorithms to predict turbine power generation to replace renewable energy with fossil fuels, Energy Explor. Exploit. 41 (2) (2023 Mar) 836–857, <https://doi.org/10.1177/01445987221138135>.
- [51] S.M. Malakouti, A.R. Ghiasi, Evaluation of the application of computational model machine learning methods to simulate wind speed in predicting the production capacity of the Swiss basel wind farm, in: In2022 26th International Electrical Power Distribution Conference (EPDC), IEEE, 2022 May 11, pp. 31–36, <https://doi.org/10.1109/EPDC56235.2022.9817304>.
- [52] S.M. Malakouti, Discriminate primary gammas (signal) from the images of hadronic showers by cosmic rays in the upper atmosphere (background) with machine learning, Phys. Scripta 98 (4) (2023 Mar 16) 045506, <https://doi.org/10.1088/1402-4896/ac1b2>.
- [53] S.M. Malakouti, Babysitting hyperparameter optimization and 10-fold-cross-validation to enhance the performance of ML methods in predicting wind speed and energy generation, Intelligent Systems with Applications 19 (2023 Sep 1) 200248, <https://doi.org/10.1016/j.iswa.2023.200248>.
- [54] S.M. Malakouti, M.B. Menhaj, A.A.M.L. Suratgar, Early breast cancer diagnosis, Curr. Probl. Cancer: Case Reports 13 (2024 Mar 1) 100278, <https://doi.org/10.1016/j.cpcr.2024.100278>.
- [55] S.M. Malakouti, Prediction of wind speed and power with LightGBM and grid search: case study based on Scada system in Turkey, International Journal of Energy Production and Management 8 (2023) 35–40, <https://doi.org/10.18280/ijepm.080105> (IETTA) (Electronic Scientific Archive URFU).
- [56] S.M. Malakouti, Cancer risk assessment based on family history and smoking habits, Sys. Rev. Pharm. 14 (6) (2023 May 1), <https://doi.org/10.31838/srp.2023.6.02>.