

A meta-learning framework for short-term wind power forecasting with SCADA and weather data

Antonis Zakynthinos
HOLISTIC IKE
Athens, Greece
azakynthinos@holisticsa.gr

Elissaios Sarmas
Decision Support Systems Laboratory
National Technical University of Athens
Athens, Greece
esarmas@epu.ntua.gr

Vasilis Michalakopoulos
Decision Support Systems Laboratory
National Technical University of Athens
Athens, Greece
vmichalakopoulos@epu.ntua.gr

Vangelis Marinakis
Decision Support Systems Laboratory
National Technical University of Athens
Athens, Greece
vmarinakis@epu.ntua.gr

Abstract—Forecasting wind power accurately remains a critical challenge in integrating renewable energy into modern power systems. In response to this challenge, this paper presents a hybrid ensemble learning framework for wind power point forecasting, combining multiple Machine Learning regressors through a meta-learning strategy. The approach leverages historical SCADA measurements, weather forecasted data and physically derived theoretical power curves to enrich the feature space. A diverse set of base models, including linear, tree-based, and gradient boosting regressors, are trained on a unified dataset, and their predictions are subsequently used to train a Multi-Layer Perceptron (MLP) meta-learner. This stacked ensemble methodology captures both the individual model strengths and their inter-dependencies, leading to improved generalization. Experimental results, based on a real-world dataset from a Vestas V52 wind turbine in Ireland, demonstrate that the meta-learner consistently outperforms the individual regressors in terms of accuracy, achieving R^2 scores of 0.99. This proves that this study offers an effective solution for improving short-term wind power forecasting.

Index Terms—wind power forecasting, machine learning, meta-learning, wind power, theoretical power curve

I. INTRODUCTION

During the 1980s, wind energy was largely dismissed by the mainstream electric power industry as an experimental or niche technology [1]. Since then, the global capacity for wind energy has increased to 837 GW, contributing to an annual reduction of more than 1.2 billion tonnes of carbon emissions, which is an amount comparable to the total annual emissions of South America according to the Global Wind Energy Council [2]. Although it proves to be an incredible Renewable Energy Source (RES) capable of offering increased power generation at much lower costs and environmental consequences, it requires efficient wind farm site selection [3] and its integration into modern power systems presents inherent unpredictability due to the intermittent nature of wind energy [4].

On the demand side, this uncertainty has caused several problems and therefore energy flexibility is needed [5], [6]. Furthermore, in addition to novelties in demand load forecasting [7], [8], recent developments in demand response have sought to resolve the operational issues arising from the integration of RES utilizing the demand flexibility [9]–[11]. However, the effectiveness of these novel techniques is highly dependent on the conditions and capabilities of the energy supply side and therefore it is crucial to improve the accuracy of wind energy generation forecasting methodologies which can be categorized into physical, statistical, Machine Learning (ML) and hybrid methods [12]. In the ML category, several studies have shown that an effective alternative to conventional algorithms involves the use of ensemble predictors [13]–[16], which combine multiple base models. By processing the output of several weak learners, this approach can improve the accuracy of both classification and regression tasks while also potentially reducing computational overhead [17].

Despite its proven value, prior wind-power research still omits physics-based power-curve features in stacked meta-learners and lacks multi-year, turbine-level tests which are gaps that highlight the contributions of this study. These omissions underscore the contributions of this study, which not only integrates theoretically derived features to better capture the physical behavior of wind turbines but also validates the proposed framework using a real-world, long-term dataset at the turbine level, offering a more comprehensive and scalable approach to wind power forecasting.

The paper is structured as follows: Section II contains a description of the methodology followed for the conducted experiment, as well as a brief overview of each model architecture. The dataset used for this study, details of the experimental process and final results are analyzed in Section III followed by Section IV which contains the concluding remarks of this study.

II. METHODOLOGY

The proposed approach of this study is shown in Figure 1. It represents a hybrid ensemble methodology for Wind Power Forecasting (WPF) by also incorporating meta-learning into a unified framework. The process begins with data acquisition, which includes the forming of the input dataset based on historical weather forecasts and wind power generation data. The collected information serve as essential input features, capturing both historical trends and anticipated atmospheric conditions. Furthermore, theoretical power curves calculated through physical modeling are included as prior information to improve forecast accuracy. These curves link manufacturer specifications with meteorological inputs, allowing the models to remain consistent with the physical principles underlying wind energy production. Subsequently, multiple regressors are trained on the pre-processed final dataset to generate individual predictions, each capturing the underlying data patterns in a different way. These predictions are then integrated into a meta-learning framework as new input features and used to train a Multi-Layer Perceptron (MLP) to optimize the combination of these base outputs. This stacked-ensemble approach yields the final predictions upon which a comprehensive comparative analysis is conducted to evaluate performance across models and assess the benefits of ensemble learning.

A. Individual Predictors

The individual regressors used in the initial prediction phase serve as diverse learners, each independently trained on the same dataset to capture varying relationships between input features and the target variable. Their outputs reflect different biases and strengths, which are later leveraged by the meta-learner to produce a more accurate and robust final prediction.

1) *Linear Regressor*: The linear regressor is a fundamental model that assumes a linear relationship between input features and a target variable. Its formula consists of the weighted sum of the input features and a bias as shown in Eq. 1

$$\hat{y} = w^\top x + b \quad (1)$$

where x is the input feature vector, w is the weight vector and b is the bias.

2) *Decision Tree Regressor*: A decision tree regressor is a supervised learning model used to predict continuous target values by learning a hierarchy of decision rules from the input features. Its internal nodes represent those decision rules based on feature values, its branches correspond to the outcomes of these rules, and its leaf nodes provide the final prediction, calculated as the mean of the target values within that specific data partition. In order to create this tree-like structure the algorithm learns to perform splits by minimizing the variance within each region. Once the best split is determined, the data are divided into two subsets, namely the left and right node, and the process is repeated for each subset recursively until a stopping criterion is met as shown in Eq. 2

$$\hat{y} = \frac{1}{R_j} \sum_{x_i \in R_j} y_i \quad (2)$$

where $|R_j|$ is the number of samples in the leaf node R_j , x_i is the feature vector of the i^{th} training sample and y_i the corresponding target value for x_i .

3) *Random Forest Regressor*: A random forest regressor is an ensemble learning method that creates multiple decision trees and the final output is the average of all their predictions. Each tree is trained on a different sample of data and each split considers a random subset of features to reduce similarities between trees. The final prediction is given in Eq. 3

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T f_t(x) \quad (3)$$

where T is the total number of trees, $f_t(x)$ is the prediction of the t^{th} tree for the input feature vector x .

4) *Extra Trees Regressor*: Extra Trees, which stands for Extremely Randomized Trees, is an ensemble ML model that combines multiple decision trees, similarly to the Random Forest, but it generates splits randomly without any optimal threshold searching and it uses the entire training sample for each tree. This increased randomness often leads to faster training and can improve generalization.

5) *Gradient Boosting Regressor*: Boosting is an ML ensemble technique that builds multiple weak learners, typically decision trees, in a sequential manner while each new model is trained to correct the residual errors of the previous ones. At each iteration t , the model is updated to minimize the remaining prediction errors, resulting in a progressively stronger and more accurate overall model, as described by the formula in Eq. (4):

$$f_t(x) = f_{t-1}(x) + \alpha \cdot l_t(x), \quad \alpha \in (0, 1] \quad (4)$$

where $f_t(x)$ is the model at a time step t , $l_t(x)$ is the weak learner trained on the residual error of the model at the previous time step $t - 1$, while α is the learning rate. The gradient boosting regressor extends the standard boosting approach shown in Eq. (4) by training each new weak learner on the pseudo-residual errors of the previous model. These are defined as the negative gradients of a specified differentiable loss function with respect to each sample and are detailed in Eq. (5):

$$r_i^{(t)} = - \left[\frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i} \right]_{\hat{y}_i = f_{t-1}(x_i)} \quad (5)$$

where $r_i^{(t)}$ is the pseudo-residual of the i^{th} sample at the time step t , $L(y_i, \hat{y}_i)$ is the differentiable loss function, y_i is the true target value for the i^{th} instance and $\hat{y}_i$ is the predicted value for the i^{th} instance.

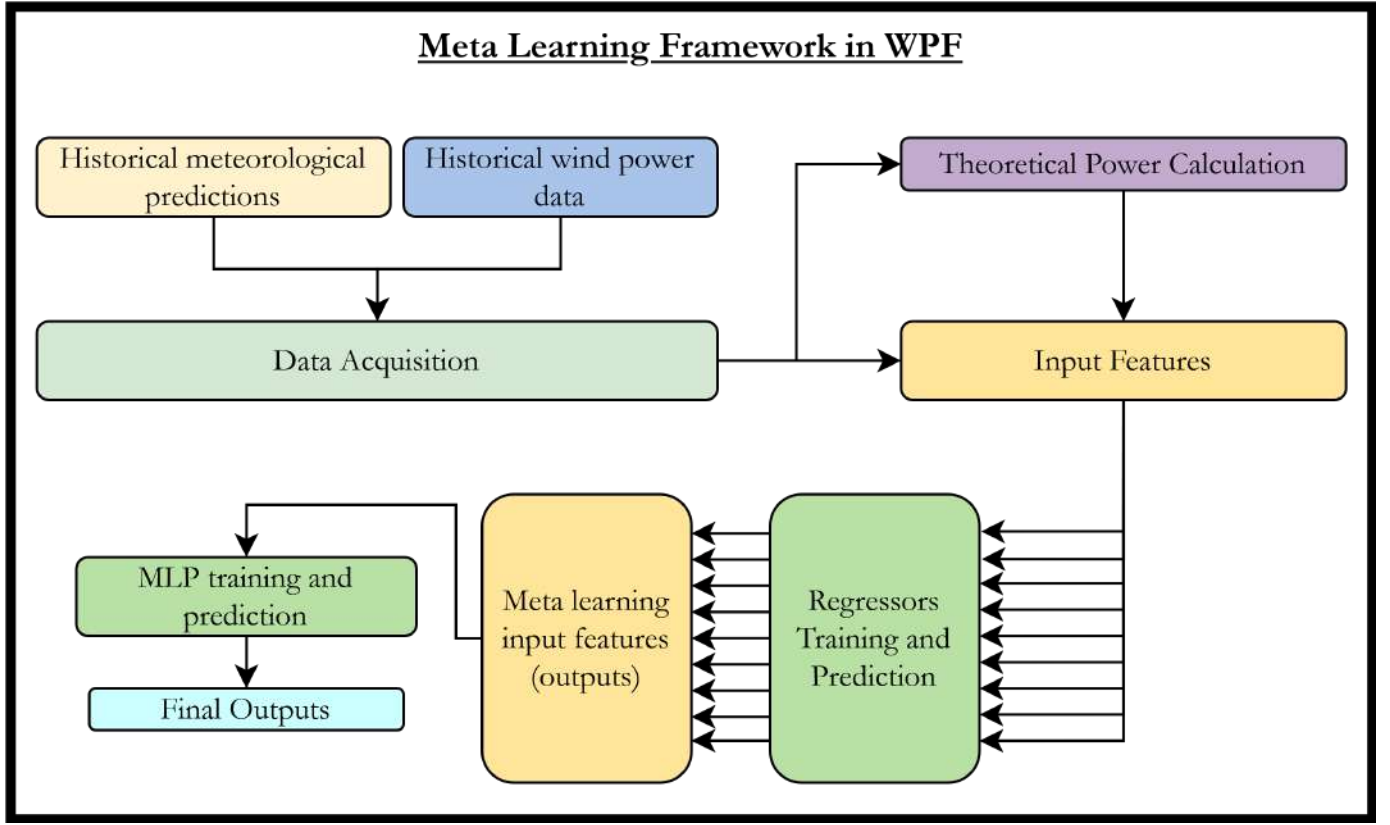


Fig. 1: Proposed methodology workflow

6) *AdaBoost Regressor*: AdaBoost, which stands for Adaptive Boosting, is an algorithm that follows the basic boosting principles defined in Eq. (4) but instead of using residuals for a weak learner's training, it updates sample weights after each learner based on its accuracy. In this way, samples that are difficult to predict are assigned larger weights. At each iteration t , a weak learner is trained to minimize its weighted average error as shown in Eq. 6

$$\epsilon_t = \sum_{i=1}^N w_i^{(t)} \cdot L(y_i, f_t(x_i)) \quad (6)$$

where $w_i^{(t)}$ is the weight of sample i at iteration t and $L(\cdot)$ is the loss function. This error reflects the weak learner's performance and is used to calculate the weighted coefficient for this weak learner as shown in Eq. 7, which in turn is used in Eq. 8 to update the sample weights.

$$\alpha_t = \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \quad (7)$$

$$w_i^{t+1} = w_i^{(t)} \cdot e^{(\alpha_t \cdot L(y_i, f_t(x_i)))} \quad (8)$$

This adaptive mechanism allows AdaBoost to focus on difficult examples and progressively refine its predictions with each iteration.

7) *CatBoost Regressor*: Traditional gradient boosting trains each weak learner on residual errors computed from the same dataset, which can lead to data leakage. Additionally, it requires manual encoding of categorical features to handle them effectively. CatBoost addresses these limitations by implementing ordered boosting and leak-free target-based encoding. It also constructs symmetric trees, where each split at a given depth occurs on the same feature using the same threshold. Rather than using pseudo-residuals as in Eq. (5), CatBoost trains on ordered pseudo-residuals, as defined in Eq. (9), where each residual for a data point is calculated using only the samples that precede it in a given permutation σ .

$$r_i^{(t)} = - \left[\frac{\partial L(y_i, f_{t-1}^{\sigma(i)-1}(x_i))}{\partial \hat{y}_i} \right] \quad (9)$$

where $\sigma(i)$ is the position of the instance i in the permutation and $f_{t-1}^{\sigma(i)-1}(x_i)$ is the previous model's prediction whose training was on the first $\sigma(i) - 1$ instances of the permutation. In addition, regarding categorical encoding, CatBoost replaces each instance of a categorical variable with the mean target value of that category, calculated using only the preceding instances in a specified permutation σ , thereby preventing target leakage. This encoding strategy is formally defined in

Eq. (10).

$$\hat{x}_k^i = \frac{\sum_{x_j \in \mathcal{D}_k} \mathbf{1}_{\{x_j^i = x_k^i\}} \cdot y_j + ap}{\sum_{x_j \in \mathcal{D}_k} \mathbf{1}_{\{x_j^i = x_k^i\}} + \alpha} \quad (10)$$

where $\hat{x}_k^i$ is the encoding of the categorical feature i of instance k , $\mathcal{D}_k = \{x_j : \sigma(j) < \sigma(k)\}$ the set of instances before the instance k , p is the prior target value and α is a regularization constant controlling the influence of the prior target value.

8) *XGBoost Regressor*: XGBoost, which stands for Extreme Gradient Boosting, is built upon the same foundational principles of gradient boosting outlined in Eq. (4) and Eq. (5), and employs a level-wise strategy for splitting tree nodes. Rather than fitting each weak learner to the pseudo-residuals of the previous model, XGBoost introduces a regularized objective function $\mathcal{L}^{(t)}$, as defined in Eq. (11), which guides the selection of the next learner. This objective incorporates both the first- and second-order derivatives of the loss function, enabling more precise optimization, while the regularization term $\Omega(l_t)$ which penalizes the chosen weak learner's complexity to prevent overfitting as shown in Eq. (12).

$$\mathcal{L}^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t)}) + \Omega(l_t) \quad (11)$$

$$L(y_i, \hat{y}_i^{(t)}) \approx L(y_i, \hat{y}_i^{(t-1)}) + g_i l_t(x_i) + \frac{1}{2} h_i l_t^2(x_i)$$

$$\Omega(l_t) = \gamma T + \frac{1}{2} \lambda \sum w_j^2 \quad (12)$$

where g_i and h_i are the gradient and hessian, respectively, of the last model's loss function, T is the number of leaves the weak learner has, λ is the L2 regularization term which helps avoid overfitting, γ is the weak learner complexity penalty and w_j is the weight of leaf j . In this way, the XGBoost model considers both the prediction accuracy and the model complexity for each iteration, offering more accurate and robust predictions.

9) *XGBoost Random Forest Regressor*: The XGBoost Random Forest (XGBRF) model is an extension of the standard XGBoost framework that adopts a bagging-based approach, closely resembling the behavior of traditional Random Forests. Instead of building trees sequentially to correct the errors of previous models, XGBRF constructs trees independently on randomly selected subsets of the training data. Once training is complete, their outputs are averaged to generate the final prediction. Despite this structural shift, XGBRF still benefits from the efficient tree-building algorithm and regularization mechanism of XGBoost, offering a robust and scalable ensemble method for regression tasks.

B. Meta-Learning Procedure

Each individual predictor is trained on the same data and provides individual predictions. Each predictor's results is considered a new input feature and used to form another dataset solely for the training of the meta-learner. The meta-learner

chosen is a MLP, which after its initial training provides the final predictions.

The MLP is considered one of the first neural networks capable of capturing non-linear relationships between input features and the target variable. It is constructed from single artificial neurons that accept the weighted input of other neurons output and process them through an activation function along with a certain bias to calculate its own individual, as can be seen in Fig. 2

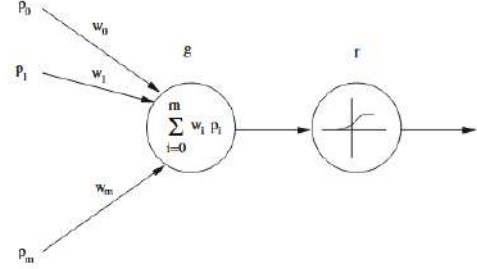


Fig. 2: Single neuron architecture [18]

Its function can be described mathematically as shown in Eq. 13:

$$Output = \text{sign} \left(\sum_{i=1}^n w_i p_i + b \right) \quad (13)$$

where p_i is the i_{th} input which can be either an input feature or the output of a neuron, w_i is the weight of the i_{th} input and b is the bias. Specifically, the MLP includes one or more hidden layers of neurons, as illustrated in Fig. 3, which model the non-linear relationships in the input data before producing the final prediction through a single output neuron.

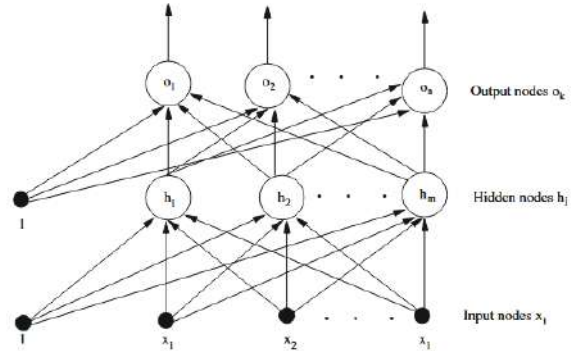


Fig. 3: MLP architecture (classification example) [18]

After an output is provided through training, the error is back-propagated to alter its parameters such as a hidden layer's weights and biases until no further improvement can be applied to the predictions as seen in Eq. 14 which focuses on an individual weight:

$$w = w + \Delta w \quad \text{where} \quad \Delta w = -\eta \frac{\partial w}{\partial L} \quad (14)$$

where Δw is the change in the weighted term, η is the learning rate and L is the error function.

III. CASE STUDY

A. Data set

For the purpose of this experiment, a dataset representing the energy behavior of a wind turbine located in Ireland was used. Its specific characteristics are shown in Table I and in order to further enrich this dataset with additional valuable information, weather data corresponding to the location of the turbine were also incorporated.

TABLE I: Vestas V52 wind turbine specifications

Characteristic	Value
Hub height [m]	60
Rotor diameter [m]	52.0
Swept Area [m^2]	2,214.0
Number of blades	3
Cut-in wind speed [m/s]	3.0
Rated wind speed [m/s]	14.0
Cut-out wind speed [m/s]	25.0
Rated power [kW]	850.0

The dataset is a multi-year operational time series compiled by Byrne and MacArtain [19], based on a Vestas V52 wind turbine located at Dundalk Institute of Technology, Ireland (53°59'00.7"N, 6°23'29.0"W). It operates in an urban setting at an altitude of 9 meters, while its rotor is situated at 60 meters with a swept area of 2,124 m^2 . Its maximum power output is 852 kW and Supervisory Control and Data Acquisition (SCADA) systems were used to record data at 10-minute intervals over nearly four years, from 2016 to 2020. Specifically, the active power output, wind speed and wind direction were collected from the initial data.

Moreover, Figure 4a shows a scatter plot between active and theoretical power generation based on wind speeds. A cut-in region for low wind speeds below 3 m/s is evident and shows a power production close to zero as expected, while it increases rapidly after that until a certain saturation period is reached. In theory, this period begins when wind speed becomes higher than 16 m/s but multiple cases of maximum wind power output can be identified for wind speeds even below 13 m/s . The theoretical power curve shows considerable variability, especially for wind speeds close to the saturation period, but it exists between acceptable boundaries accounting for factors such as turbulence, environmental conditions, and mechanical inefficiencies. Active power output shows an expected spread from the theoretical curve, especially for wind speeds above 12 m/s as both overestimation and underestimation can be identified, mainly due to limitative factors such as environmental effects and sensor inaccuracies. Furthermore, Figure 4b reveals that the dominant wind directions that contribute the

most to power generation range from 90°–160° (Southeast) and 200°–300° (Southwest).

The finalized set of input features also includes the theoretical power output of the turbine. To calculate this, historical weather data from the Open-Meteo Application Programming Interface (API) [20] were used. Specifically, exact location coordinates of the turbine were provided to retrieve historical values of air temperature, dew point temperature, and sea-level pressure at an hourly resolution for the corresponding time period. However, a time-resolution alignment was required since the turbine's meter readings were recorded at 10-minute intervals, which led to the duplication of each hourly weather value across six consecutive intervals.

B. Experimental design

Initially, for the correct execution of this experiment, a dataset of appropriate structure and consistency must be constructed to support the training of the models. The information collected from the original dataset includes active power measurements, wind speed, wind direction and the corresponding timestamps. For an additional enhancement of the data and the theoretical power estimation, weather-related inputs such as temperature, dew point, pressure and relative humidity were incorporated. Subsequently, any outliers present in features such as active power, wind speed and wind direction are identified and replaced with the mean of neighboring values to maintain data integrity. Moreover, the feature space was further enhanced with new features derived from existing ones. In particular, the time step feature was decomposed into month, week, day, hour, minute, and season to better capture temporal dependencies and seasonal patterns in the data. Furthermore, the mean wind speed, the mean wind direction, and the wind orientation were calculated, along with the sine and cosine transformations of wind direction, to more effectively model their periodic nature and therefore complex relationship with the target variable.

Following the formation and pre-processing of the final dataset, a splitting procedure occurred that led to the formation of a training dataset with the first 80% of the data and a testing dataset with the final 20% of the data, while a normalization step was applied to scale all variables to the [0,1] range to mitigate the increased influence of features with much higher numerical values during training. Furthermore, the regressors used for the initial training are shown in Table II and all models undergone optimization except the linear regressor. The main hyperparameter tuned was the maximum depth for each base learner these regressors utilize for their final predictions and its value was set to 10 to prevent possible overfitting.

Following the initial training, each regressor's predictions were provided for the previously generated testing dataset. This was further split to a new training and testing dataset with the same ratio, while the individual model's predictions formed a new input feature set for the meta-learner. For this purpose, an MLP with one hidden layer paired with a dropout layer was chosen to train for 50 epochs on this feature set and capture the

TABLE II: Base Regressors Specifications

Regressor	Specifications
Linear Regressor	-
Decision Tree Regressor	Max Depth = 10
Random Forest Regressor	Max Depth = 10
Extra Trees Regressor	Max Depth = 10
Gradient Boosting Regressor	Max Depth = 10
AdaBoost Regressor	Max Depth = 10
CatBoost Regressor	Max Depth = 10
XGBoost Regressor	Max Depth = 10
XGBoost Random Forest Regressor	Max Depth = 10

dependencies between each prediction and the actual value. In addition, hyperparameter optimization was performed through the Optuna Library [21] to obtain the optimal value of its hyperparameters between the ranges shown in Table III

TABLE III: MLP hyperparameter optimization

Hyperparameter	Range
Hidden layer size	[32, 64, 128]
Learning rate	[0.0001, 0.001, 0.01]
Dropout rate	[0.2, 0.3, 0.4, 0.5]
Batch size	[32, 64]

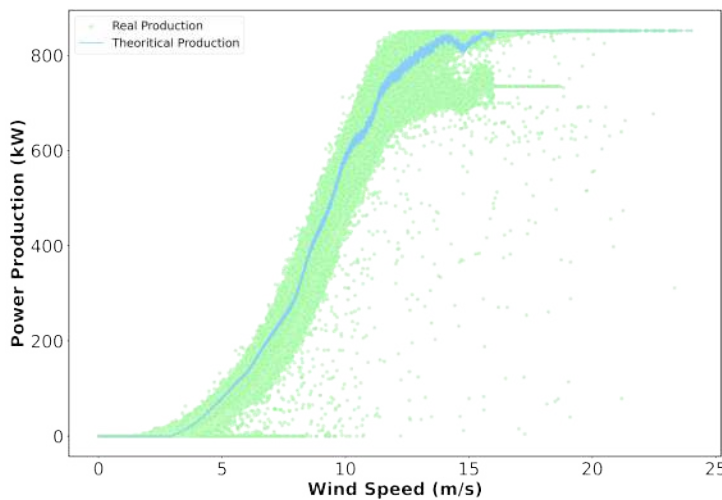
The final model had a hidden layer of size 32, with a learning rate of 0.001, a dropout rate of 0.5 and the following training was performed with a batch size of 32 where the model was used to generate predictions on the unseen

testing dataset. These predicted values were then inversely transformed to restore their original scale, allowing for a direct comparison with the actual measurements. This enabled a thorough performance evaluation using several widely adopted error metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2 score).

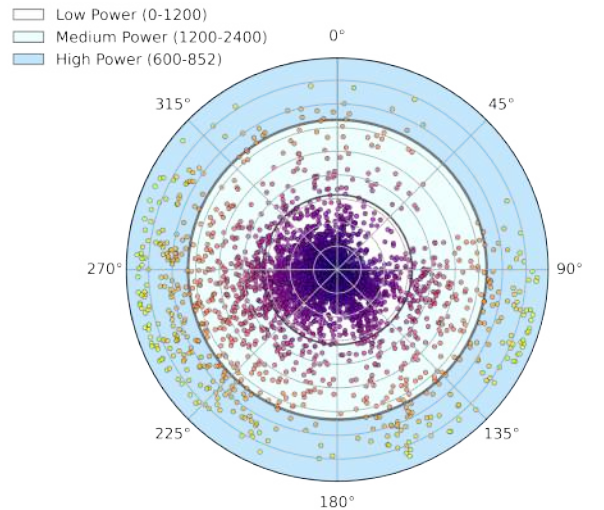
C. Results

In order to illustrate the performance of the ensemble strategy for short-term WPF as proposed in this study, a comprehensive comparison of the prediction results between the MLP meta-learner and each individual regressor is performed by utilizing the aforementioned performance evaluation metrics.

To this end, the respective performance of the various regression models together with the Ensemble MLP meta-learner across the entirety of the final test set is presented in Table IV. Among the evaluated models, the proposed ensemble method achieves the lowest error across all three related key metrics such as RMSE, MAE and MAPE, demonstrating high accuracy and consistency while managing to keep large deviations to a minimum. In addition to minimizing the prediction error, it also manages to attain the highest coefficient of determination with a score of 99%. This exceptionally high R^2 score implies the ability of the model to also accurately capture the underlying structure and temporal dependencies within the data. From the base learners, random forest and linear regressors perform reasonably well with R^2 scores being relatively close to that of the meta-learner, but their elevated RMSE and MAE values suggest increased deviations from the actual observations. On the other end of the spectrum, XGBoost records the highest MAPE among all models, indicating a greater relative error in its predictions, while AdaBoost consistently performs the worst across all other metrics, indicating that it is in fact the least reliable model for this task.



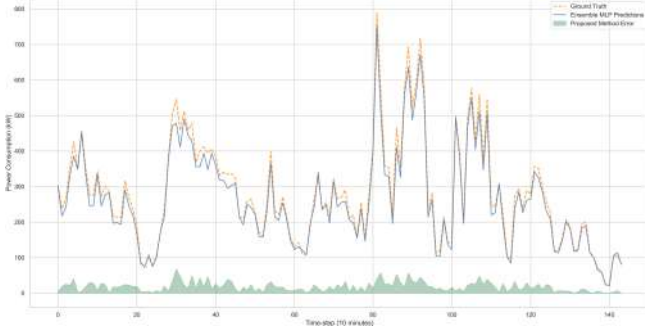
(a) Power production and wind speed scatter plot



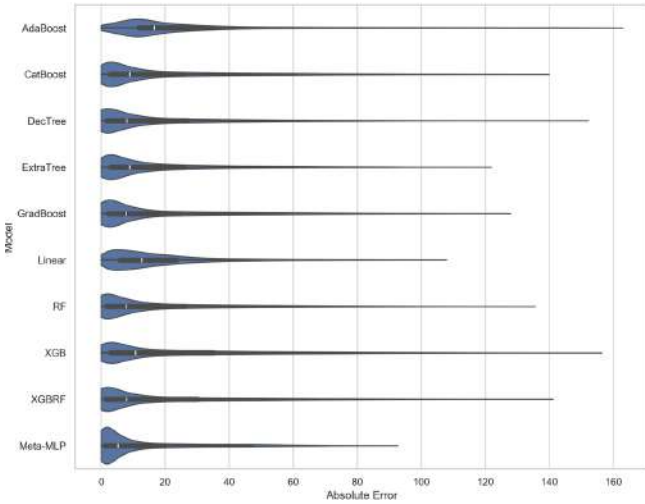
(b) Power production and wind direction polar plot

Fig. 4: Data visualization plots

Moreover, a comparison of the predicted day-ahead power generation values of the meta-learner against the ground truth is presented in Fig. 5a, while a visualization of the respective error is included as well. As can be observed, the aforementioned results are validated, since the prediction line closely tracks the actual values even during peak loads, while the total absolute error is consistently low throughout the sequence, confirming the model’s stability, accuracy and capacity to capture nonlinearity.



(a) Day-ahead MLP Ensemble predictions plot



(b) Cross-model error distribution plots

Fig. 5: Ensemble method forecasting performance visualizations

Lastly, to further understand the distribution of absolute errors across all tested models, several violin plots are presented in Fig. 5b. The Meta-MLP model again exhibits an error distribution characterized by increased concentration

and narrowness which translates to minimal variation and a consistently low prediction error. In contrast, models such as AdaBoost, XGBoost and Decision Trees show much wider and skewed distributions, which indicates higher error variance and more frequent large deviations.

Overall, the results demonstrate the superiority of the ensemble technique through an MLP meta-learner which effectively integrates the strengths of individual models to deliver competitive performance, while also confirming the value of stacking-based ensemble learning in enhancing the predictive performance on such a complex regression task.

IV. CONCLUSIONS & FUTURE WORK

This study introduced a meta-learning-based ensemble framework aiming to enhance the accuracy and reliability of wind power point forecasts. By capturing the non-linear relationships between several heterogeneous regressors predictions and the actual target values using an MLP, the proposed methodology effectively harnesses the individual strengths of each base model. The integration of physically derived theoretical features, rigorous preprocessing and optimized hyperparameter tuning contributed to a robust learning pipeline.

Experimental evaluation confirmed that the meta-learner not only outperformed the individual models in terms of predictive accuracy, but also demonstrated superior generalization capability, achieving near-perfect R^2 scores. These results highlight the potential of ensemble meta-learning approaches to address the inherent uncertainty and complexity of wind power forecasting tasks.

Despite the performance of the model though, it is evidently constrained to a single on-shore turbine and a fixed ten minutes sampling which may mask sub-minute dynamics. As such, future work will aim to: (i) extend the framework to handle multiple turbines with varying configurations and geographical locations, (ii) incorporate higher-resolution heterogeneous temporal data to capture short-term variability, (iii) investigate probabilistic forecasting techniques to provide uncertainty quantification alongside point estimates and (iv) explore adaptive and online learning strategies for continuous model updating in dynamic real-time environments. These steps in future research will support further improving the robustness, scalability, and practicality of the proposed approach for large-scale wind energy integration.

ACKNOWLEDGMENT

The work presented is based on research conducted within the framework of the Horizon Europe European Commission project CRETE VALLEY (Grant Agreement No. 101136139).

TABLE IV: Total metrics results

Metric	AdaBoost	CatBoost	DecisionTrees	ExtraTrees	GradientBoost	Linear	RF	XGB	XGBRF	Ensemble-MLP
RMSE	54.513	32.426	36.566	29.581	30.527	27.403	31.478	38.304	34.022	23.246
MAE	34.425	19.372	19.515	17.157	17.919	18.301	18.486	23.684	20.118	13.841
MAPE	0.378	0.352	0.325	0.355	0.364	0.434	0.299	0.497	0.410	0.234
R^2 Score	93.282	98.062	97.508	98.393	98.275	98.674	98.161	97.235	97.820	99.004

The content of the paper is the sole responsibility of its authors and does not necessarily reflect the views of the EC.

REFERENCES

- [1] U. D. of Energy. (2023) How wind energy became integral to the modern grid. [Online]. Available: <https://www.energy.gov/eere/wind/articles/how-wind-energy-became-integral-modern-grid>
- [2] F. J. Lee, "Gwec global wind report 2021," Global Wind Energy Council, Tech. Rep., 2021. [Online]. Available: <https://yenader.org/wp-content/uploads/2020/10/GWEC-I-Global-Wind-Report-2021.pdf>
- [3] I. C. Dimitriou, E. Sarma, G. P. Trachanas, V. Marinakis, and H. Doukas, "Multi-criteria gis-based offshore wind farm site selection: Case study in greece," *Renewable and Sustainable Energy Reviews*, vol. 207, p. 114962, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364032124006889>
- [4] F. Ullah, X. Zhang, M. Khan, M. S. Mastoi, H. M. Munir, A. Flah, and Y. Said, "A comprehensive review of wind power integration and energy storage technologies for modern grid frequency regulation," *Heliyon*, vol. 10, no. 9, p. e30466, 2024. [Online]. Available: <https://doi.org/10.1016/j.heliyon.2024.e30466>
- [5] I. Papias, V. Michalakopoulos, E. Sarma, V. Marinakis, G. Antonese, T. Cioara, and I. Anghel, "A data-driven framework for estimating residential energy flexibility for aggregated demand-side management," *Available at SSRN 5147627*, 2025.
- [6] I. Papias, V. Michalakopoulos, E. Sarma, and V. Marinakis, "Aggregated flexibility dynamic forecasting of residential energy prosumers for dr programs," in *2024 15th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2024, pp. 1–9.
- [7] A. Gabriel, T. Cioara, I. Anghel, I. Papias, V. Michalakopoulos, and E. Sarma, "Hybrid transformer model with liquid neural networks and learnable encodings for buildings' energy forecasting," *Energy and AI*, p. 100489, 2025.
- [8] E. Sarma, S. Strompolas, V. Marinakis, F. Santori, M. A. Bucarelli, and H. Doukas, "An incremental learning framework for photovoltaic production and load forecasting in energy microgrids," *Electronics*, vol. 11, no. 23, p. 3962, 2022.
- [9] V. Michalakopoulos, E. Sarma, I. Papias, P. Skaloumpakas, V. Marinakis, and H. Doukas, "A machine learning-based framework for clustering residential electricity load profiles to enhance demand response programs," *Applied Energy*, vol. 361, p. 122943, 2024.
- [10] L. Todorean, T. Cioara, I. Anghel, E. Sarma, V. Michalakopoulos, and V. Marinakis, "Demand response optimization for smart grid integrated buildings: Review of technology enablers landscape and innovation challenges," *Energy and Buildings*, p. 115067, 2024.
- [11] V. Michalakopoulos, E. Sarma, V. Daropoulos, G. Kazdaridis, S. Keranidis, V. Marinakis, and D. Askounis, "A multi-dimensional unsupervised machine learning framework for clustering residential heat load profiles," *arXiv preprint arXiv:2411.16701*, 2024.
- [12] E. Arslan Tuncar, Şafak Sağlam, and B. Oral, "A review of short-term wind power generation forecasting methods in recent technological trends," *Energy Reports*, vol. 12, pp. 197–209, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352484724003603>
- [13] E. Sarma, E. Spiliotis, E. Stamatopoulos, V. Marinakis, and H. Doukas, "Short-term photovoltaic power forecasting using meta-learning and numerical weather prediction independent long short-term memory models," *Renewable Energy*, vol. 216, p. 118997, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960148123009035>
- [14] E. Sarma, E. Spiliotis, V. Marinakis, T. Koutselis, and H. Doukas, "A meta-learning classification model for supporting decisions on energy efficiency investments," *Energy and Buildings*, vol. 258, p. 111836, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S037877882200007X>
- [15] E. Sarma, E. Spiliotis, N. Dimitropoulos, V. Marinakis, and H. Doukas, "Estimating the energy savings of energy efficiency actions with ensemble machine learning models," *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/4/2749>
- [16] E. Sarma, V. Marinakis, and H. Doukas, *Ensemble Machine Learning Models for Estimating Energy Savings from Efficiency Measures in Buildings*. Cham: Springer Nature Switzerland, 2025, pp. 177–197.
- [17] J. Heinermann and O. Kramer, "Machine learning ensembles for wind power prediction," *Renewable Energy*, vol. 89, pp. 671–679, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960148115304894>
- [18] A. Meyer-Baese and V. Schmid, "Chapter 7 - foundations of neural networks," in *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, second edition ed., A. Meyer-Baese and V. Schmid, Eds. Oxford: Academic Press, 2014, pp. 197–243. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124095458000078>
- [19] R. Byrne and P. MacArtain, "Multi-annual time series operational data for a vestas v52 wind turbine located in a coastal peri-urban environment in ireland," *Data in Brief*, vol. 48, p. 109227, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352340923003463>
- [20] P. Zippenfenig, "Open-meteo.com weather api," 2023. [Online]. Available: <https://open-meteo.com/>
- [21] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2623–2631. [Online]. Available: <https://doi.org/10.1145/3292500.3330701>