Wind Forecast at Medium Voltage Distribution Networks

Herbert Amezquita Ortiz
herbert.amezquita@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Portugal
October 2022

***Abstract*—Due to the intermittent and variable nature of wind, Wind Power Generation Forecast (WPGF) has become an essential task for power system operators, who are looking for a reliable wind penetration into the electric grid. Since there is a need to forecast wind power generation accurately, the main contribution of this thesis is the development, implementation and comparison of WPGF methods to be used by Distribution System Operators (DSOs). The methodology applied comprised five stages, pre-processing, feature selection, forecasting models, post-processing and validation; using historical wind power generation data (measured at secondary substations) of 20 wind farms connected to the Medium Voltage (MV) distribution network of Portugal.**

After comparing the accuracy of eight different models in terms of their Relative Root Mean Square Error (RRMSE), Extreme Gradient Boosting (XGBOOST) appeared as the best-suited forecasting method for wind power generation. The best average RRMSE achieved by the proposed XGBOOST model for 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021) corresponds to 13.48%, outperforming the predictions of the Portuguese DSO by more than 20%.

***Index Terms*—Medium Voltage Distribution Network, Short-Term Forecasting, Wind Power Generation Forecast, Extreme Gradient Boosting (XGBOOST)**

I. Introduction

Nowadays, the world is going through an energy transition process from fossil fuels to renewable energies, that aims to reduce the environmental impact of the energy sector. To increase the penetration rate of Renewable Energy Sources (RES) in power systems, significant incentive schemes and policies have been considered by governments. The European Union (EU) under the 2030 climate and energy framework for the period 2021-2030 is part of the ambitious European Green Deal. The framework commits the EU to reduce greenhouse emissions at least 40% (as compared to 1990 levels), to increase the amount of renewable energy in the energy mix by at least 32% and to improve energy efficiency by at least 32.5% [1]. To achieve those targets, a high penetration of RES like solar, wind, hydropower, geothermal, biomass, biofuels, waves or tidal is necessary.

Out of all the available RES, Solar Photovoltaic (PV) and wind are considered now the most abundant, developed, economically viable and commercially accepted worldwide [2]. Without considering hydropower, wind has the higher installed capacity of the renewables and according to the Global Wind Report 2021, year 2020 was the best year in history for the global wind industry.

Since supply and demand should be equal at all times but wind power generation depends on the availability of wind, that is a weather dependent source, the integration into the existing electricity supply system brings some challenges at the level of secondary substations that need to be addressed by DSOs of power networks. The challenges include system stability and reliability, due to grid congestion or intermittency of supply; system balance, that requires a strong information exchange between the DSO and the Transmission System Operator (TSO) or flexibility services (voltage support and demand-side response) to ensure that the network is stabilized amid the varying energy generation and consumption [3].

Here is where WPGF appears as one of the most efficient ways to overcome some of these problems and to help the power system operators to reduce the risk of unreliable electricity supply. The development of new techniques to improve understanding of wind power generation, through simulation, forecasting, distribution curve fitting, filtering and modeling, allows making better decisions about expansion of the wind sector and better management of the electricity system [4].

Thus, this thesis intends to develop and implement a framework with several forecasting models for wind power generation in wind farms connected to the MV distribution network of Portugal. Specifically Persistence, Auto-Regressive (AR), Auto-Regressive with Exogenous Variable (ARX), Long Short-Term Memory (LSTM) neural network, XGBOOST, Random Forest (RF), Decision Trees (DT) and Support Vector Machine (SVM) models are developed and tested using real data measured at the secondary substations and provided by the Portuguese DSO.

This data covers seven years of information (2015-2021) of power generated by 20 wind farms in Portugal mainland. It also includes the DSO predictions for the years 2020 and 2021, that are used to compared with our models results (through an error metric). Different meteorological parameters that might influence the forecast results like temperature, radiation, wind speed or wind direction are also considered into the models and that weather data comes from the Instituto Português do Mar e da Atmosfera (IPMA). Two years of meteorological data are available for the analysis, specifically 2020 and 2021. The main goal of this work is then to improve the DSO performance by reducing the error as much as possible.

The remainder of the thesis is organized as follows: Section II presents a literature review related to wind power and wind speed forecasting. Section III explains systematically how the work was done. Section IV shows the forecast results obtained for each method and the comparison in terms of error performance between them and also with the DSO predictions provided. This section also includes the different tests or improvements performed to the final method chosen. Finally, Section V summarizes the main outcomes of the thesis, the limitations encountered in the process and suggests future work related to the topic.

II. Literature Review

Wind power generation forecast have been a topic of interest for many researchers during the recent years, due to importance of integrating RES to the power system and all the implications that it brings. Hence, this section presents a review of regression and Artificial Intelligence (AI) forecasting methods and a general overview of different publications and studies related to WPGF.

A. Wind Forecast Classification

A forecast system is characterized by its time horizon, which is the future time period for which the wind generation will be predicted. Based on [5] wind forecasting can be separated according to the prediction horizon, into the following categories:

- Very-short-term forecasting: Few seconds to 30 minutes ahead.
- Short-term forecasting: 30 minutes to 6 hours ahead. Mainly useful for operational purposes (economic load dispatch planning, load increase/decrease decisions).
- Medium-term forecasting: 6 hours to 1 day ahead. Aim to increase operational security of day ahead electricity markets and corroborate online/offline decisions.
- Long-term forecasting: Multiple days ahead to 1 year or more. Provide information for power system risk assessment and also to identify potential for wind power generation in specific areas, providing valuable data for energy planners [4].

B. Wind Forecast Methods

Based on the analysis of the literature, wind forecast methods can be divided into six overall groups: Persistence method, physical methods, statistical methods, Artificial Neural Networks (ANN) based models, hybrid methods and new models.

Persistence method uses the simple assumption that the value at a certain future time will be the same as it is when the forecast is made. It is based on the assumption of a high correlation between present and future values and produces accurate predictions for very-short term forecasts [6]. As expected, the accuracy of this model degrades rapidly with the increasing prediction lead time [7], so it is normally used as a reference to evaluate the performance of advanced methods.

Physical methods use forecast values from a Numerical Weather Prediction (NWP) model as an input to calculate the wind power generation based on the power curve.

Statistical methods are based on training with measured data (time series). They are easy to model, capable to provide timely prediction [7] and mostly used for short-term forecasting. Several types of time series models may be considered, but the most popular are AR and its variants ARX, ARMA and ARIMA.

ANN can identify the non-linear relationships between input features and output data. ANN are typically composed of nodes (or neurons) that are distributed across different layers, namely input, hidden and output layers. Each node in a layer is linked to the ones in the next by means of a weight parameter that measures the strength of that connection [8]. There are several kinds of ANN but the most common neural networks used for WPGF are: Feed Forward Neural Network (FFNN), Back-Propagation Neural Network (BPNN) and Recurrent Neural Network (RNN), which also includes a more advanced version called LSTM neural network.

Hybrid methods refer to the combination of different forecasting methods with the aim of retaining the merits of each technique and improve the overall accuracy [9]. It includes the combination of physical and statistical methods, the combination of alternative statistical methods or the combination of models for short-term and medium-term forecasting for instance.

The last group corresponds to some novel wind forecast models that have been developed in recent years. Between the most interesting ones, XGBOOST, Adaptative Neural Fuzzy Inference System (ANFIS), RF and SVM models have achieved the most accurate predictions for wind power generation.

Some of the most relevant papers found in the literature about WPGF are:

A study made by *M.Duran et al* [10], that tested an ARX model for wind power prediction using wind speed as exogenous variable. The results for a 24 hours time horizon showed a significant improvement in accuracy, when comparing the mean error of their model with persistence and a traditional AR model. According to [10], when compared with AR the improvement of ARX is about 14% and about 26% when compared with persistence.

A paper of *J.Catalao* [11] that presents a successful application of ANN in combination with wavelet transform for short-term wind power forecasting in Portugal. The model proposed predicts the value of wind power for 3 hours ahead and it is compared with persistence, ARIMA and other neural network approach. The results of the study confirmed that this model is effective, since the Mean Absolute Percentage Error (MAPE) has an average value of 6.97%, outperforming the other methods analyzed in [11]. Also, the introduction of the wavelet transform enables a reduction of the error when compared with the normal neural network.

A model developed by *M.Mabel et al* [12] to forecast wind power generation of seven wind farms in Muppandal, India. A BPNN is implemented using three input variables: wind speed, relative humidity and generation hours. The model accuracy is evaluated then by comparing the predicted power with the

actual measured values, using two years of training and one year of forecast. The results are satisfactory and in agreement, since the overall percentage error obtained was 4%.

A study from *H.Zheng et al* [13] that proposes a model for short-term wind power generation forecast based on XGBOOST, with weather similarity analysis and feature engineering. Hourly wind power generation is predicted for the week between April 21st and 28th of 2017, using the data from January 1st of 2016 to April 20th of 2017 as training. The results of the proposed model are compared with a BPNN, RF, SVM and a single XGBOOST model. Among all the methods, XGBOOST produced the highest accuracy of prediction, while weather similarity analysis and feature engineering significantly improved the accuracy of the forecasting results when comparing with the single XGBOOST model.

A paper of *Y.Kassa et al* [14] that presents an ANFIS based approach for one day ahead hourly wind power generation forecas. The proposed model is trained with historical wind speed and wind power data of a 2.5 MW rated wind turbine installed in Beijing, using one year of information. The performance of the ANFIS model is therefore evaluated against persistence, a BPNN and a hybrid method and the results demonstrated that ANFIS outperformed all other methods tested, achieving an average MAPE of 6.88%.

A study from *L.Fugon et al* [15] that evaluates three different models for short-term WPGF. The models analyzed are ANN, RF and SVM, while three wind farms in France are considered in the analysis. The data used corresponds to a time series of hourly power production for a 18 months period, specifically from July 2004 to December 2005. For the same period, NWP of Meteo France are used, considering two meteorological variables, wind speed and gust wind direction. The forecast is made once a day for time horizons from 0 to 60 hours ahead (3-hour resolution) and the results revealed that RF outperformed the rest of the models.

In summary, the literature review showed that WPGF is an extend task that depends on the time horizon of the forecast, the resolution and quantity of data used or the meteorological variables considered. There is not a clear method that outperforms all others for WPGF and that is the reason why this thesis develops and compares different methods. The main focus is to find the model that best fits the characteristics of the wind farms analyzed and considering that the sample of 20 wind farms studied represent 10% of the total number of wind farms connected to the MV distribution network of Portugal, the results might be significant for the DSO.

III. Methodology

The methodology proposed in this thesis corresponds to the five stages presented in Figure 1 and described in this section.

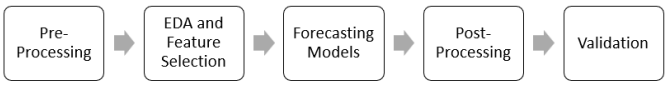

Fig. 1. Methodology stages

A. Pre-Processing

This stage intends to prepare the raw data and make it suitable for the forecasting models by removing the outliers and by dealing with missing data.

In the case of outliers, all the data points that present a value of power generation higher than the installed capacity of the wind farm to which they belonged, are considered outliers and therefore are removed from the datasets. Negative values of power, if they exist, are considered inconsistent data points and are adjusted to zero.

To deal with missing data, several strategies are applied to fill in the gaps. All the strategies are specifically based on two factors, the position (where the data is missing) and the quantity (number of consecutive values that are missing).

In case the missing data is located at the beginning of the dataset, instead of trying to fill the missing values, the algorithm will decrease the length of the training set to the first value that is available but respecting the minimum quantity of data defined. If in the training set 50% or more of the values are missing, then no forecast is done and the training set becomes invalid. On the other side, if the missing data is located at the end of the dataset, a calculation based on the median is used to fill in the missing values.

When the missing data is not located on the extremes but it is in the middle of the dataset (having available values before and after the gap), four different scenarios are considered:

- If the missing data correspond to one hour (4 data points) or less, the interpolation approach is used. Since only a small number of values are missing, a straight line between both sides gives a good approximation of the missing values.
- From one hour (4 data points) to one day (96 data points) of missing data, an approach based in adjusting the profile of the previous day is used. It considers the time where the missing data is found and also the previous day information for that specific moment, to make a normalization and adapt it to the current day.
- If the missing data goes from one day (96 data points) to one week of 5 days (480 data points), again a median approach is used, but in this case the day of the week and the exact time where the data is missing are also considered. It is relevant to mention at this point that only real values contribute to the median, values created by the missing data algorithm are not taken into account in the median calculation.
- For more than one week (more than 480 data points) of missing values, the gap is not filled because creating artificial values for long periods of time may have a negative effect in the forecast models and consequently in the results. The approach in this case, is to remove the dates that contain the large periods of missing data from the training set, as long as the minimum length defined for the training set is respected.

B. EDA and Feature Selection

Exploratory Data Analysis (EDA) is the process where the user look at and understand the data with statistical and visualization methods. To have an idea about the data contained in the IPMA dataset, Table I presents some descriptive statistics of wind farm 15. The variables T and R stand for temperature and radiation.

TABLE I
DESCRIPTIVE STATISTICS OF WIND FARM 15

	Power (kW)	T (K)	R (W/m^2)	Wind Speed (m/s)	Wind Direction ($^\circ$)
Count	70,176	70,153	70,153	70,153	70,153
Mean	7,834.07	288.58	735.84	7.01	241.51
Std Dev	7,177.94	4.95	729.82	2.80	109.90
Min	0.00	272.94	0.00	0.14	0.02
25th Perc	1,600.00	285.25	86.43	4.93	157.24
50th Perc	5,610.00	288.11	524.35	6.92	284.33
75th Perc	13,102.50	291.45	1,193.59	9.08	338.96
Max	29,705.29	310.58	2,814.88	16.32	359.98

Feature selection consists in determine which features (input variables) will be used in the forecasting models. Only a few variables in the dataset are useful for building the models and the rest of the features are either redundant or irrelevant. If we input the dataset with all these redundant or irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the models [16].

To select the appropriate features, a correlation matrix, which provides the relationship between variables is used. Figure 2 shows the correlation matrix of wind farm 15.

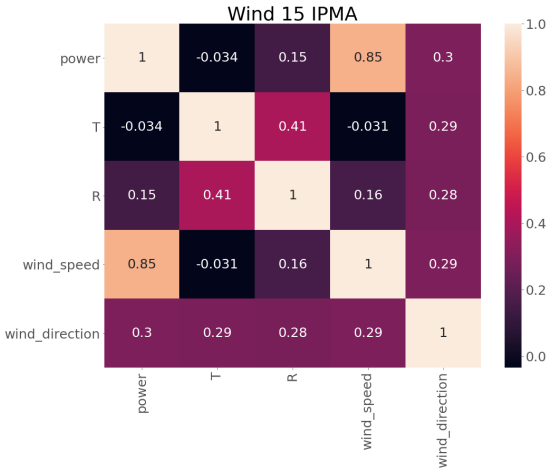

Fig. 2. Correlation matrix of wind farm 15

Based on the correlation matrix, only wind speed and wind direction features are selected to forecast the power generation. Wind speed presents the higher correlation as expected, followed by wind direction. The rest of the variables are discarded because they have either negative or very low correlation.

C. Forecasting Models

Once the outliers are removed, the missing values are filled or handled and the feature selection has been done, the final dataset is divided into the following two subsets:

- Training set, data used by the model to discover and learn patterns between the features and the forecast variable, power.
- Test set, data on which the power predictions are generated. Correspond to unseen data used to evaluate the performance of the model.

The training set is normally larger than the test set because the idea is to feed the model with as much data as possible, to learn meaningful patterns and then apply the things learned to create predictions on unseen data.

Eight different forecasting models are implemented to predict the power generation of the 20 wind farms, starting from persistence (to have a benchmark), passing trough regressive models, a neural network and some newer models.

Specifically, the following forecasting models are tested:

1) Persistence

Persistence forecast corresponds to the power measured at the same time instant from the previous day (96 time intervals before the desired forecast time instant). It can be formulated as:

$$\hat{X}(t) = X(t - 96) \tag{1}$$

Where $\hat{X}(t)$ is the wind power forecast value at certain instant of time and $X(t-96)$ is the wind power value measured 96 time intervals before.

2) Auto-Regressive (AR)

AR(p) model relates p past observations to the current value X_t as [17]:

$$X_t = \mu + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{2}$$

Where μ is the mean value, φ_i is a coefficient which reflects each past observation X_{t-i} influence on current value and ε_t is the actual stochastic perturbation.

3) Auto-Regressive with Exogenous Variable (ARX)

ARX model is an auto-regressive model with exogenous inputs that can described as [18]:

$$X_t = \mu + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{n_x} \eta_i b_{t-i} + \varepsilon_t \tag{3}$$

Where η_i is the exogenous coefficient and n_x is the order of the exogenous inputs.

4) Long Short-Term Memory (LSTM) Neural Network

LSTM is one of many types of RNN. Since RNN cannot store long time memory, LSTM proved to be very useful in forecasting with long time data based on 'memory line'. In a LSTM the memorization is performed trough gates and every node consists of a set of cells responsible of storing passed data streams. To develop the LSTM model in Python, the library tf.keras.layers.LSTM was used.

5) Decision Trees (DT)

Present a tree-like structure, made up of different nodes. The root node is the start of the decision tree, which is usually the whole dataset. Leaf nodes are the endpoint of a branch, or the final output of a series of decisions. The features of the data are internal nodes and the outcome is the leaf node [19]. To develop the DT model in Python, the library sklearn.tree.DecisionTreeRegressor was used.

6) Random Forest (RF)

Combines several decision trees and uses the majority voting of the individual trees to find the overall class. It consists in three steps: randomly selecting training data when making trees, choosing some subsets of features when splitting nodes and employing only a subset of all features for splitting each node in each simple decision tree. To develop the RF model in Python, the library sklearn.ensemble.RandomForestRegressor was used.

7) Extreme Gradient Boosting (XGBOOST)

The process of additive learning in XGBOOST as explained by *N.Dhieb et al* [20] is presented below. First, consider a data set D expressed as follows:

$$D = \{(x_i, y_i),\ where\ x_i \in \mathbb{R}^m\ and\ y_i \in \mathbb{R}\} \tag{4}$$

$$|D| = n \tag{5}$$

Where m is the dimension of the features x_i, y_i is the response of the sample i and n is the number of samples. The vertical bars in Equation 5 denotes the cardinality of the set.

Then, the predicted value of the entry i and denoted as $\hat{y_i}$, is defined as:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i),\ where\ f_k \in F \tag{6}$$

Where f_k indicates an independent tree in the space of regression trees F and $f_k(x_i)$ refers to the predicted score given by the i-th sample and k-th tree.

The objective function of the XGBOOST, denoted by ζ, is given as follows:

$$\zeta = \sum_{i=1}^{n} \ell(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{7}$$

By minimizing the objective function ζ, the regression tree model functions f_k can be learned. The training loss function $\ell(y_i, \hat{y_i})$ evaluates the difference between the prediction $\hat{y_i}$ and the actual value y_i. Herein, the term Ω is used to avoid the overfitting problem by penalizing the model complexity as follows:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2 \tag{8}$$

Where γ and λ are regularization parameters, T and w are respectively the numbers of leaves and the scores on each leaf.

A second degree Taylor series can be used to approximate the objective function. Let's define $I_j = \{i|q(x_i) = j\}$ an instance set of leaf j with $q(x)$ a fixed structure. The optimal weights w_j^* of leaf j and the corresponding optimal value can be obtained by the following equations:

$$w_j^* = -\frac{g_j}{h_j + \lambda} \tag{9}$$

$$\zeta^* = \frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i\ \in\ I_j}\ g_i)^2}{(\sum_{i\ \in\ I_j}\ h_i + \lambda)} + \lambda T \tag{10}$$

Where g_i and h_i are the first and the second gradient orders of the loss function ζ. The loss function ζ can be used as a quality score of the tree structure q. The smaller the score is, the better the model is.

As it is not possible to enumerate all the tree structures, a greedy algorithm can solve the problem by starting from a single leaf and iteratively add branches to the tree. Let's say that I_R and I_L are the instance sets of right and left nodes after split. Assuming $I = I_R \cup I_L$, the loss reduction after the split is given as:

$$\zeta_{split} = \frac{1}{2}\left[\frac{(\sum_{i\ \in\ I_L}\ g_i)^2}{\sum_{i\ \in\ I_L}\ h_i + \lambda} + \frac{(\sum_{i\ \in\ I_R}\ g_i)^2}{\sum_{i\ \in\ I_R}\ h_i + \lambda} - \frac{(\sum_{i\ \in\ I}\ g_i)^2}{\sum_{i\ \in\ I}\ h_i + \lambda}\right] - \gamma \tag{11}$$

This formula is usually used in practice for evaluating the split candidates. The XGBOOST model use many simple trees and score leaf nodes during splitting. The first three terms of the equation represent respectively the score of the left, right and original leaf. In addition, the term γ is the regularization on the additional leaf and it will be used in the training process.

To develop the XGBOOST model in Python, the library xgboost.XGBRegressor was used.

8) Support Vector Machine (SVM)

SVM regression trains the model using symmetrical loss function, which penalizes for both high and low misestimates. The aim is to find a hyperplane that differentiates the data points plotted in multi-dimensional space, where each dimension represents the different features used. To develop the SVM model in Python, the library sklearn.svm.SVR was used.

D. Post-Processing

Its main purpose is to check the generated power predictions and adjust the values out of range if they exist. To do that, the algorithm checks two conditions:

- $Power\ predictions >= 0$. The predicted power cannot be negative. In case there are negative values, they are adjusted to 0.
- $Power\ predictions <= Installed\ capacity$. The predicted power cannot be higher than the installed capacity of the wind farm. In this case the maximum forecast value is limited to the installed capacity.

E. Validation

The error metric defined to evaluate the performance of the forecasting models is based on the Root Mean Square Error (RMSE) but with a small difference: in this case the error is normalized by dividing by the installed capacity of the wind farm.

Thus, it is called RRMSE and is calculated as:

$$RRMSE\ (\%) = \frac{RMSE}{P_{installed}} \times 100 \tag{12}$$

$$RRMSE\ (\%) = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2}}{P_{installed}} \times 100 \tag{13}$$

Where N is the total number of samples, $\hat{P}_i$ is the forecast value, P_i is the measured value and $P_{installed}$ is the installed capacity of the wind farm.

The algorithm calculates the daily RRMSE between predictions and real values for the test period defined and then the average of this daily error is reported (as a percentage), to have an idea of the accuracy of the forecast made. This RRMSE metric is used as comparison point in all the results presented.

IV. Results and Discussion

This section presents the results obtained for the forecasting models developed, the comparison of the RRMSE between them and with the DSO results. It also presents the different tests and the tuning performed to the best-suited model in order to improve the results.

A. WPGF Models

Table II presents the RRMSE for Persistence, AR and ARX models; while Table III presents the RRMSE for LSTM, DT, RF, XGBOOST and SVM models. The training and test periods defined in all the simulations were: JUN-NOV of 2021 for training and DEC of 2021 forecast. The meteorological parameters wind speed and wind direction were used as features in all the models that use exogenous variables.

TABLE II
RRMSE for Persistence, AR and ARX: 6 months training, 1 month forecast

Wind Farm	Persistence (%)	AR (%)	ARX (%)	DSO (%)
1	24.594	16.886	15.384	13.482
2	37.717	35.105	19.248	16.187
3	14.895	12.396	12.515	41.013
4	35.714	32.035	28.131	19.850
5	30.814	26.713	20.364	14.874
6	32.897	27.102	21.185	18.929
7	35.403	28.947	19.673	50.877
8	38.103	32.230	27.287	21.536
9	30.248	26.306	20.168	14.372
10	34.136	30.781	20.690	45.416
11	31.939	29.759	23.380	29.586
12	24.222	19.232	17.073	15.593
13	33.787	29.940	18.394	17.470
14	38.087	30.688	25.111	21.550
15	26.191	20.947	13.660	11.954
16	37.940	26.028	36.241	21.983
17	34.902	23.803	29.912	19.541
18	34.712	28.478	33.975	26.619
19	31.060	19.225	24.680	18.242
20	29.404	21.114	26.565	18.998
Average	31.838	27.522	21.046	22.904

TABLE III
RRMSE for LSTM, DT, RF, XGBOOST and SVM: 6 months training, 1 month forecast

Wind Farm	LSTM (%)	DT (%)	RF (%)	XGBOOST (%)	SVM (%)	DSO (%)
1	23.209	19.543	12.371	12.451	19.492	13.482
2	24.413	24.596	19.952	17.637	31.561	16.187
3	8.288	15.671	11.257	10.549	10.399	41.013
4	29.888	28.721	28.698	22.649	47.410	19.850
5	22.727	21.783	14.873	13.617	29.755	14.874
6	22.555	25.147	23.205	19.273	35.377	18.929
7	29.426	32.046	21.851	21.101	20.949	50.877
8	25.833	25.484	25.398	24.427	39.240	21.536
9	26.900	23.291	17.296	17.472	24.699	14.372
10	26.700	22.602	21.286	16.374	28.953	45.416
11	21.877	25.785	22.783	21.412	34.463	29.586
12	19.562	23.656	17.673	17.509	23.907	15.593
13	21.859	17.593	18.195	12.947	26.973	17.470
14	26.925	29.963	26.919	28.127	36.026	21.550
15	22.833	17.059	12.828	11.651	15.297	11.954
16	29.310	25.150	22.071	20.628	37.585	21.983
17	27.285	26.611	25.206	21.862	45.642	19.541
18	24.323	31.979	28.080	26.764	29.098	26.619
19	21.575	27.072	18.103	17.219	24.802	18.242
20	27.717	23.865	19.042	18.595	34.802	18.998
Average	24.160	24.381	20.354	18.613	29.822	22.904

From the results obtained in Tables II and III just three methods, ARX (21.046%), RF (20.354%) and XGBOOST (18.613%) outperformed the DSO results (22.904%). Since XGBOOST has the lower RRMSE, it is chosen as the method to be focus on and to be improved, in order to reduce the percentage of error even more.

B. XGBOOST Adjusting Training and Test Periods

The first test consists on adjusting the training and test periods, to compare the RRMSE of the XGBOOST model under different time horizons. Based on the two years of IPMA data available (2020 and 2021), the following eight combinations of training and test periods are defined:

- *Combination 1:* 6 months training (JAN-JUN of 2021) and 6 months forecast (JUL-DEC of 2021).
- *Combination 2:* 7 months training (JAN-JUL of 2021) and 5 months forecast (AUG-DEC of 2021).
- *Combination 3:* 8 months training (JAN-AUG of 2021) and 4 months forecast (SEP-DEC of 2021).
- *Combination 4:* 9 months training (JAN-SEP of 2021) and 3 months forecast (OCT-DEC of 2021).
- *Combination 5:* 10 months training (JAN-OCT of 2021) and 2 months forecast (NOV-DEC of 2021).
- *Combination 6:* 11 months training (JAN-NOV of 2021) and 1 month forecast (DEC of 2021).
- *Combination 7:* 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021).
- *Combination 8:* 1 year training (JAN-DEC of 2020) and 1 year forecast (JAN-DEC of 2021).

The results obtained are summarized in Figure 3, that presents the average RRMSE of the 20 wind farms for each combination, achieved by the XGBOOST model and by the DSO.

To have a fair comparison between the different combinations, regardless of the number of months to forecast, the RRMSE of the same month (DEC of 2021) was analyzed independently of the combination and the same results were obtained.

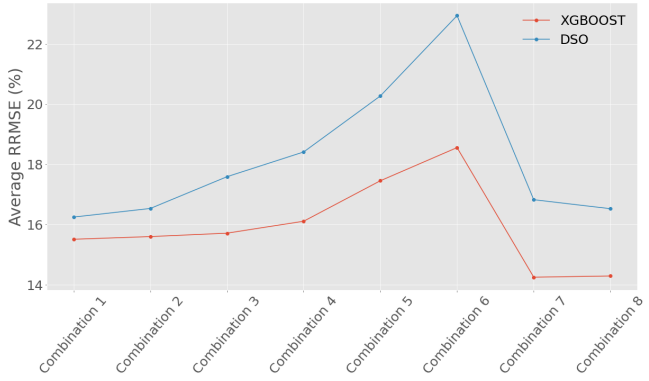

Fig. 3. Average RRMSE for each combination

Figure 3 shows that first, the error of the XGBOOST model developed is always lower than the error of the DSO for any combination of training and test sets. Second, the XGBOOST model more accurately forecasts long periods of time like 6 months (Combination 7) or 1 complete year (Combination 8) instead of short periods of time like 1 month (Combination 6) or 2 months (Combination 5). Third, the best combination found corresponds to Combination number 7: 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021) with an average RRMSE of 14.257%. From now on these training and forecast periods are used in all tests.

C. XGBOOST Hyperparameter Tuning

Hyperparameter tuning or hyperparameter optimization, is the process of determining the right combination of hyperparameters that maximizes a machine learning or AI model performance.

The hyperparameters of XGBOOST that are tuned are the following [21]:

- max depth: Maximum depth per tree. A deeper tree might increase the performance, but also the complexity and chances to overfit. The value must be an integer greater than 0. Default is 6.
- learning rate: Determines the step size at each iteration while the model optimizes toward its objective. A low learning rate makes computation slower, and requires more rounds to achieve the same reduction in residual error as a model with a high learning rate. The value must be between 0 and 1. Default is 0.3.
- n estimators: The number of trees in our ensemble. Equivalent to the number of boosting rounds. The value must be an integer greater than 0. Default is 100.
- colsample by tree: Represents the fraction of columns to be randomly sampled for each tree. It might improve overfitting. The value must be between 0 and 1. Default is 1.
- sub sample: Represents the fraction of observations to be sampled for each tree. Lower values prevent overfitting, but might lead to underfitting. The value must be between 0 and 1. Default is 1.
- min child weight: Defines the minimum sum of weights of all observations required in a child. It is used to control overfitting. The larger it is, the more conservative the algorithm will be. The value must be an integer greater than 0. Default is 1.

To find the best combination of hyperparameters for the XGBOOST model, the Random Search optimization algorithm is used. It consists in a large range of hyperparameters values, which are randomly iterated a specific number of times over combinations of the values defined. The number of iterations defined for the Random Search is 50 and the Mean Square Error (MSE) is the metric used to evaluate the performance for each combination of hyperparameters.

This process is done only once because the computation time is very high and it takes a long time to get the results.

Table IV presents the best combination of hyperparameters obtained for each wind farm after running the Random Search and the average values of each hyperparameter, when considering the 20 wind farms all together.

TABLE IV
BEST XGBOOST HYPERPARAMETERS FOR EACH WIND FARM

Wind Farm	max depth	learning rate	n estimators	colsample by tree	sub sample	min child weight
1	2	0.050	200	0.7	0.7	10
2	2	0.050	500	1.0	0.7	10
3	2	0.001	385	1.0	1.0	5
4	3	0.030	200	1.0	0.7	5
5	3	0.030	200	1.0	1.0	10
6	3	0.030	500	1.0	0.5	10
7	2	0.017	610	0.7	1.0	5
8	3	0.050	200	1.0	0.5	5
9	2	0.050	500	1.0	0.7	10
10	2	0.005	715	1.0	1.0	3
11	3	0.022	345	1.0	0.7	10
12	2	0.050	200	1.0	0.5	3
13	2	0.050	100	1.0	1.0	10
14	3	0.100	100	0.7	1.0	10
15	2	0.025	502	1.0	1.0	5
16	2	0.048	181	1.0	0.7	5
17	3	0.054	208	1.0	1.0	5
18	2	0.046	217	0.7	0.7	3
19	2	0.050	500	1.0	0.7	10
20	2	0.050	500	1.0	0.7	10
Average	2	0.04	343	0.9	0.8	7

Considering the obtained results, two tests, one using the best combination of hyperparameters for each wind farm (Best combination) and the other using the same average values

of hyperparameters for all wind farms (Average values) are performed. The idea is to compare the best RRMSE achieved so far (Best results until now), with the RRMSE obtained after the hyperparameter optimization. The results are presented in Table V, using 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021) that was the best combination found in the previous test.

TABLE V
RRMSE FOR XGBOOST AFTER HYPERPARAMETER TUNING: 1 YEAR TRAINING, 6 MONTHS FORECAST

Wind Farm	Best results until now (%)	Best combination (%)	Average values (%)	DSO (%)
1	11.247	10.321	10.338	10.193
2	13.775	12.835	12.860	13.443
3	8.558	7.586	11.194	26.794
4	17.922	17.070	17.497	17.293
5	11.947	11.066	11.174	11.221
6	12.886	11.770	12.125	12.321
7	14.374	13.547	13.567	39.519
8	15.996	14.452	14.756	15.510
9	14.989	14.279	14.254	11.981
10	15.495	14.246	14.404	30.680
11	14.400	13.450	13.526	19.459
12	10.595	9.838	9.909	10.538
13	15.012	14.076	14.175	14.034
14	16.386	15.358	15.459	15.931
15	10.804	9.729	9.796	10.308
16	19.087	16.147	16.237	18.055
17	15.345	14.352	14.732	15.129
18	17.286	16.302	16.335	16.313
19	13.685	12.660	12.726	12.689
20	15.350	14.525	14.555	15.140
Average	14.257	13.180	13.481	16.827

From Table V it is possible to observe that the average RRMSE was reduced from 14.257% to 13.180% after the hyperparameter tuning done specifically for each wind farm, meaning an improvement of 7.55%. In the other case, where the RRMSE was computed using the average values of hyperparameters instead of the specific combination found for every wind farm, the average RRMSE achieved was 13.481%. In both cases a considerable reduction of the error was achieved with the hyperparameter tuning.

After the comparison between the two tests performed, it is decided that for future forecasts just the average combination of hyperparameters ($max\ depth = 2$, $learning\ rate = 0.04$, $n\ estimators = 343$, $colsample\ by\ tree = 0.9$, $sub\ sample = 0.8$, $min\ child\ weight = 7$) will be used to run the XGBOOST model independently of the wind farm. This, considering that the DSO has 200 wind farms connected to the MV distribution network of Portugal and running the Random Search for each one is not worth the computation time required (around 12 hours per wind farm) for the little extra improvement obtained when calculating the best combination of hyperparameters specific for every wind farm.

To have a graphical sense of the predictions obtained after the hyperparameter tuning, Figure 4 presents the comparison between XGBOOST predictions, DSO predictions and the real values of power for one month of the forecast period, specifically February.

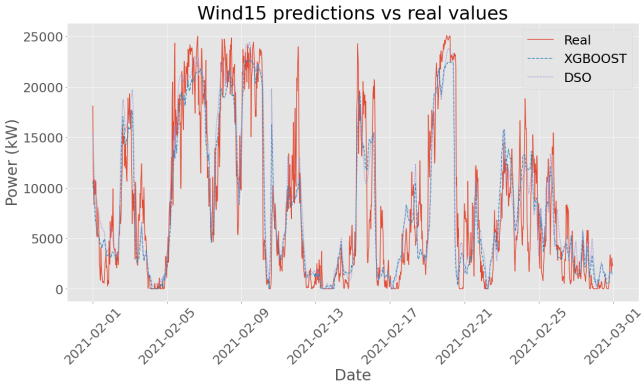

Fig. 4. Forecast vs real values for FEB of 2021

D. XGBOOST with Backtesting

Backtesting is a term used in modeling that refers to testing a predictive model on historical data. It involves moving backward in time, step-by-step, in as many stages as it is necessary. Hence, it is a special type of cross-validation applied to previous periods [22].

The purpose of this test is then to apply the backtesting with refit and increasing training size strategy inside the XGBOOST model, to see if the RRMSE can be reduced. To do that, the model is trained each time before making a new prediction, then that prediction is included in the training set and the process is repeated until all the predictions are made. That means that the model uses all the data available so far, while the training set increases sequentially, maintaining the temporal order of the data.

The initial training set in our case corresponds to 1 year of data (JAN-DEC of 2020), the prediction horizon correspond to 1 day (meaning that the model is trained in each iteration to forecast each day separately) and the retraining is done until the 6 months (JAN-JUN of 2021) that correspond to the forecast period are predicted.

Table VI presents the RRMSE achieved when using the backtesting strategy implemented inside the XGBOOST model. The results obtained with backtesting are better than the best RRMSE achieved until now. There is a little improvement of 2.8%, since the error was reduced from 13.481% to 13.097%. However, when considering the computation time that backtesting requires, which is in average 10 hours per wind farm, the small reduction of the error makes not worth to implement this strategy into the model.

For the DSO the main point is that the model is able to do the forecast in a short computing time because they have 200 wind farms connected to the MV distribution network of Portugal. The implemented XGBOOST model takes between $20 - 30$ seconds per wind farm to run and with backtesting

it takes 1500 times more. Since the accuracy of the forecast with backtesting does not represent a significant improvement, the inclusion of backtesting is discarded.

TABLE VI
RRMSE FOR XGBOOST USING BACKTESTING STRATEGY

Wind Farm	Best Results (%)	Backtesting (%)	DSO (%)
1	10.338	10.053	10.193
2	12.860	12.568	13.443
3	11.194	9.212	26.794
4	17.497	16.475	17.293
5	11.174	10.683	11.221
6	12.125	-	12.321
7	13.567	13.247	39.519
8	14.756	14.109	15.510
9	14.254	13.740	11.981
10	14.404	14.031	30.680
11	13.526	12.901	19.459
12	9.909	9.355	10.538
13	14.175	13.983	14.034
14	15.459	14.809	15.931
15	9.796	9.575	10.308
16	16.237	17.668	18.055
17	14.732	14.153	15.129
18	16.335	15.899	16.313
19	12.726	12.547	12.689
20	14.555	13.832	15.140
Average	13.481	13.097	16.827

E. Stacking

Stacking is the process of using different machine learning and AI models one after another, where the predictions from each model are added as new features. It is done in layers, and there can be arbitrarily many layers, dependent on exactly how many models are trained, along with the best combination of these models. At the end, the final dataset combining the initial features plus the predictions created after each layer are feed into a last model. The last model is called a meta-learner, and its purpose is to generalize all the features from each layer into the final predictions [23].

Figure 5 presents the diagram of the stacking process implemented in this case.

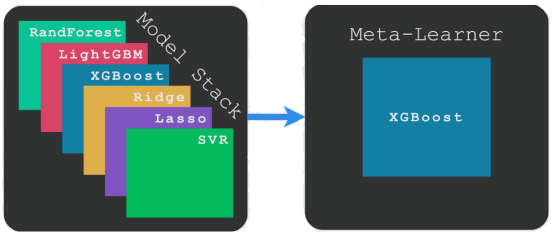

Fig. 5. Stacking process implemented [23]

First, six layers were defined using the following models: RF, Light Gradient Boosting Machine (LGBM), XGBOOST, Ridge, Lasso and SVM. Then, the XGBOOST model was used again as meta-learner to obtain the final predictions.

Table VII presents the RRMSE obtained using the stacking approach with RF, LGBM, XGBOOST, Ridge, Lasso and SVM layers and XGBOOST meta-learner, for 1 year training and 6 months forecast.

TABLE VII
RRMSE FOR STACKING APPROACH

Wind Farm	Best Results (%)	Stacking (%)	DSO (%)
1	10.338	10.358	10.193
2	12.860	12.856	13.443
3	11.194	8.793	26.794
4	17.497	17.685	17.293
5	11.174	11.136	11.221
6	12.125	12.225	12.321
7	13.567	13.589	39.519
8	14.756	14.619	15.510
9	14.254	14.077	11.981
10	14.404	14.731	30.680
11	13.526	13.569	19.459
12	9.909	9.974	10.538
13	14.175	14.585	14.034
14	15.459	15.394	15.931
15	9.796	9.853	10.308
16	16.237	16.288	18.055
17	14.732	14.912	15.129
18	16.335	16.134	16.313
19	12.726	12.689	12.689
20	14.555	14.375	15.140
Average	13.481	13.392	16.827

In this case, by using stacking the RRMSE passed from 13.481% to 13.392%, equivalent to a 0.66% improvement. Regarding the computation time required by this approach, for each wind farm it took on average 15 minutes to run, that is 40 times more than the normal XGBOOST (that takes between $20 - 30$ seconds to run). Therefore, even when the RRMSE results are better when using stacking, the little reduction of the error is not worth the extra computation time and this approach is discarded.

V. CONCLUSIONS

In this work eight different forecasting models namely, Persistence, AR, ARX, LSTM neural network, DT, RF, XGBOOST and SVM were developed and tested to predict the power generation of 20 wind farms connected to the secondary substations of the MV distribution network of Portugal.

After comparing the eight models between them and with the DSO predictions, the results showed that for 6 months training (JUN-NOV of 2021) and 1 month forecast (DEC of 2021), XGBOOST obtained the best performance with a RRMSE of 18.613%, followed by RF with a RRMSE of 20.354% and ARX with a RRMSE of 21.046%. The rest of the models obtained an error that is higher than the error of the DSO predictions for the same period, which corresponds to a RRMSE of 22.904%. Specifically, LSTM neural network, DT, AR, SVM and Persistence obtained respectively a RRMSE of 24.160%, 24.381%, 27.522%, 29.822% and 31.838%.

With XGBOOST as the best-suited forecasting model for the wind farms analyzed, some tests and improvements were performed to this method in order to reduce the error as much as possible. It was found that the best combination of training and test periods based on the two years of information available for IPMA, corresponds to 1 year of training (JAN-DEC of 2020) and 6 months of forecast (JAN-JUN of 2021). When using this specific combination the average RRMSE gets reduced to 14.257%.

A hyperparameter tuning of XGBOOST using Random Search optimization was carried out to improve the previous result. The best combination of hyperparameters were found for each wind farm and the average RRMSE got reduced to 13.180%. However, since the computation time to run Random Search (around 12 hours) is very high, it was decided to use the average values of the hyperparameters independently of the wind farm. Using the average values of the hyperparameters the RRMSE achieved was 13.481%, that is not so far from the value obtained using the best combination of hyperparameters, and therefore this approach should be used for future forecasts or with new wind farms.

Other improvements that lowered the best RRMSE (13.481%) of the developed XGBOOST model were achieved using backtesting and stacking approaches. In the case of backtesting the RRMSE got reduced to 13.097%, while for stacking the RRMSE got reduced to 13.392%. Nevertheless, both processes require a longer computation time, 10 hours per wind farm for backtesting and 15 minutes per wind farm for stacking, than the normal XGBOOST model which takes only between 20 to 30 seconds per wind farm to run. Since one of the most important characteristics of a forecasting model is to make predictions in an efficient way, meaning rapidly and with accuracy, it was concluded that the small reduction of the error achieved with this strategies is not worth the large computation time needed and consequently, backtesting and stacking were discarded.

After all, using the proposed XGBOOST model for 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021), the best average RRMSE achieved for the 20 wind farms studied, corresponds to 13.481%; after discarding Random Search, backtesting and stacking of course. The results successfully fulfilled the main goal of this thesis, that was to improve the performance of the actual DSO forecasting system, which for the same period of analysis presents a RRMSE of 16.827%. With the XGBOOST model developed an improvement of 20% is achieved. The framework is scalable, computationally efficient and can be used for future wind power forecasting, if the DSO want to obtain predictions with higher accuracy.

REFERENCES

[1] S. Micheli, "Policy strategy cooperation in the 2030 climate and energy policy framework," *Atlantic Economic Journal*, vol. 48, no. 2, pp. 265–267, 2020.

[2] M. S. Javed, T. Ma, J. Jurasz, and M. Y. Amin, "Solar and wind power generation systems with pumped hydro storage: Review and future perspectives," *Renewable Energy*, vol. 148, pp. 176–192, 2020.

[3] P. Wilczek, "Connecting the dots: distribution grid investments to power the energy transition," in *11th Solar & Storage Power System Integration Workshop (SIW 2021)*, vol. 2021. IET, 2021, pp. 1–18.

[4] S. A. Vargas, G. R. T. Esteves, P. M. Maçaira, B. Q. Bastos, F. L. C. Oliveira, and R. C. Souza, "Wind power generation: A review and a research agenda," *Journal of Cleaner Production*, vol. 218, pp. 850–870, 2019.

[5] C. Monteiro, R. Bessa, V. Miranda, A. Botterud, J. Wang, and G. Conzelmann, "Wind power forecasting: State-of-the-art 2009." Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2009.

[6] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *North American Power Symposium 2010*. IEEE, 2010, pp. 1–8.

[7] Y. Wu and J. Hong, "A literature review of wind forecasting technology in the world," in *2007 IEEE Lausanne Power Tech*. IEEE, 2007, pp. 504–509.

[8] E. Machado, T. Pinto, V. Guedes, and H. Morais, "Electrical load demand forecasting using feed-forward neural networks," *Energies*, vol. 14, no. 22, p. 7644, 2021.

[9] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, "A critical review of wind power forecasting methods—past, present and future," *Energies*, vol. 13, no. 15, p. 3764, 2020.

[10] M. J. Duran, D. Cros, and J. Riquelme, "Short-term wind power forecast based on arx models," *Journal of Energy Engineering*, vol. 133, no. 3, pp. 172–180, 2007.

[11] J. d. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, "Short-term wind power forecasting in portugal by neural networks and wavelet transform," *Renewable energy*, vol. 36, no. 4, pp. 1245–1251, 2011.

[12] M. C. Mabel and E. Fernandez, "Analysis of wind power generation and prediction using ann: A case study," *Renewable energy*, vol. 33, no. 5, pp. 986–992, 2008.

[13] H. Zheng and Y. Wu, "A xgboost model with weather similarity analysis and feature engineering for short-term wind power forecasting," *Applied Sciences*, vol. 9, no. 15, p. 3019, 2019.

[14] Y. Kassa, J. Zhang, D. Zheng, and D. Wei, "Short term wind power prediction using anfis," in *2016 IEEE international conference on power and renewable energy (ICPRE)*. IEEE, 2016, pp. 388–393.

[15] L. Fugon, J. Juban, and G. Kariniotakis, "Data mining for wind power forecasting," in *European Wind Energy Conference & Exhibition EWEC 2008*. EWEC, 2008, pp. 6–pages.

[16] Feature selection techniques in machine learning. [Online]. Available: https://www.javatpoint.com/feature-selection-techniques-in-machine-learning

[17] G. Santamaría-Bonfil, A. Reyes-Ballesteros, and C. Gershenson, "Wind speed forecasting for wind farms: A method based on support vector regression," *Renewable Energy*, vol. 85, pp. 790–809, 2016.

[18] H. N. Akouemo and R. J. Povinelli, "Data improving in time series using arx and ann models," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3352–3359, 2017.

[19] Seldon, "Decision trees in machine learning explained," Nov 2021. [Online]. Available: https://www.seldon.io/decision-trees-in-machine-learning

[20] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "Extreme gradient boosting machine learning algorithm for safe auto insurance operations," in *2019 IEEE international conference on vehicular electronics and safety (ICVES)*. IEEE, 2019, pp. 1–5.

[21] D. Martins, "Xgboost: A complete guide to fine-tune and optimize your model," Dec 2021. [Online]. Available: https://towardsdatascience.com/xgboost-fine-tune-and-optimize-your-model-23d996fab663

[22] J. Amat and J. Ortiz, "Skforecast: Time series forecasting with python and scikit-learn," Feb 2021. [Online]. Available: https://www.cienciadedatos.net/documentos/py27-time-series-forecasting-python-scikitlearn.html

[23] C. Hansen, "Model stacking explained and python code," Jan 2020. [Online]. Available: https://mlfromscratch.com/model-stacking-explained/