



A fuzzy-based cascade ensemble model for improving extreme wind speeds prediction

C. Peláez-Rodríguez ^{a,*}, J. Pérez-Aracil ^b, L. Prieto-Godino ^{c,a}, S. Ghimire ^d, R.C. Deo ^d, S. Salcedo-Sanz ^{a,d}

^a Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, 28805, Spain

^b Department of Computer Systems Engineering, Universidad Politécnica de Madrid, Madrid, 28038, Spain

^c Department of Digitalization and O&N tools, Iberdrola, Spain

^d School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, QLD, 4300, Australia

ARTICLE INFO

MSC:
0000
1111

Keywords:

Wind energy
Wind speed extremes
Wind extremes prediction
Extreme Learning Machine
Fuzzy ensemble

ABSTRACT

A novel fuzzy-based cascade ensemble of regression models is proposed to address a problem of extreme wind speed events forecasting, using data from atmospheric reanalysis models. Although this problem has been mostly explored in the context of classification tasks, the innovation of this paper arises from tackling a continuous predictive domain, aiming at providing an accurate estimation of the extreme wind speed values. The proposed layered framework involves a successive partition of the training data into fuzzy-soft clusters according to the target variable value, and further training a specific regression model within each designated cluster, so that each model can analyze a particular part of the target domain. Finally, predictions made by individual models are integrated into a fuzzy-based ensemble, where a pertinence value is designated to each model based on the previous layer's prediction, and on the defined membership functions for each cluster. A Differential Evolution (DE) optimization algorithm is adopted to find the optimal way to perform data partitioning. Fast training randomized neural networks methods are used as final regression schemes. The performance of the proposed methodology has been assessed by comparison against state-of-the-art methods in real data from three wind farms in Spain.

1. Introduction

1.1. Motivation and incitement

Wind provides an environmentally sustainable source of energy with a mature technology spread around the world that supports zero greenhouse gases emissions, negligible environmental impacts and relatively low costs (Herbert et al., 2007; Monfared et al., 2009). It is expected that 22% of the world's electricity will be supplied by wind power by 2030 (Zhang et al., 2019). Nonetheless, as wind power is directly related to wind speed, its intermittent and stochastic nature lead to volatility and unsteadiness of power generation, imposing an adverse influence on cost control and power system stability, hampering its integration into conventional power grid systems (Zhou et al., 2011). A significant alleviation to this problem may be achieved if the operation of wind farms can be estimated in advance on the basis of accurate information from dynamic wind speed or power production forecasting, improving the efficiency of wind power system management and reducing system failures (Li and Jin, 2018; Wang et al., 2016).

In the case that wind speed is used to estimate wind farm production, a power curve is necessary to convert the wind speed prediction to power prediction. Direct forecast of the wind farm production is also possible, of course. This alternative is more or less equivalent to use wind speed forecasting in terms of performance (Tascikaraoglu and Uzunoglu, 2014). However, note that there are occasions where it is not possible to deal with real production data, for example in resource studies previous to the wind farm construction, where the only possibility is to use wind speed.

In addition, wind turbine control systems also require wind speed forecasting in order to improve its performance. In this case, the prediction time-horizon must be set in the range of seconds or minutes, since its major problems are the delays associated with wind turbine systems. These delays affect the response of the system with respect to controller action (Riahy and Abedi, 2008). Accurate short-term wind speed forecasting also reduce voltage and frequency fluctuations between expected and produced electrical power, and diminishes the sudden cut-offs of wind power resulting from excessive wind speeds (Miranda

* Corresponding author.

E-mail address: cesar.pelaez@uah.es (C. Peláez-Rodríguez).

and Dunn, 2006). Consequently, this can improve the reliability of a wind power generation systems and its integration into the electricity network system (Lei et al., 2009; Zhang et al., 2013).

Short-term wind speed forecasts are usually carried out in the order of minutes to hours (Costa et al., 2008). This paper is focused on 1-hour ahead forecasting, which is useful for proper management of wind farms. Intensive research in this area has been carried out during the last two decades (Chang, 2013; Tascikaraoglu and Uzunoglu, 2014; Wang et al., 2021). In this sense, generally speaking, methods presented in the literature can be divided into two categories: (1) weather-based, which relies on the study of physical phenomena to build a model (Numerical Weather Model, NWM) (Cassola and Burlando, 2012; Yang and Tsai, 2019); and (2) time series based, where both statistical (Miranda and Dunn, 2006; Kantz et al., 2004; Huang and Chalabi, 1995) or Artificial Intelligence (AI) (Kusiak et al., 2009; Salcedo-Sanz et al., 2009a; Kani and Riahy, 2008; Mohandes et al., 2004; Bali et al., 2019; Malik and Yadav, 2021; Ren et al., 2022) methods are used to analyze historical wind speed data series, with the necessary mention to deep learning approaches that have recently emerged and provide outstanding results (Fukuoka et al., 2018; Ismail Fawaz et al., 2019; Gamboa, 2017). There are also hybrid approach mixing NWM with AI methods such as in Salcedo-Sanz et al. (2009b, 2011) and Wang et al. (2022). In this context, Artificial Neural Networks (ANN) offer a highly compelling alternative, due to its ability to handle nonlinear complex problems providing a robust solution and easy implementation. ANNs have been massively and successfully applied to address short-term wind speed forecasting problems (Cadenas and Rivera, 2009; Kaur et al., 2016; Khosravi et al., 2018; Jiang et al., 2021; Duan et al., 2021). Specifically, randomized neural networks methods (Freire et al., 2020) stand as particular ANN architectures where a random mechanism is used to reduce the setting and selection of parameters, allowing these ANN to learn with fast training speed and good generalization performance (Saavedra-Moreno et al., 2013). Among this approaches, Extreme Learning Machine (ELM) (Huang et al., 2006) and Random Vector Functional Link (RVFL) (Pao et al., 1994; Zhang and Suganthan, 2016; Shi et al., 2021) stand out as the most widely applied. In Del Ser et al. (2022) renewable energy prediction tackled with randomization-based models is reviewed. In Luo et al. (2018) a stacked ELM (SELM) was implemented to predict short-term wind speed. In Tian et al. (2019) a Particle Swarm Optimization (PSO) was applied on a ELM structure to reduce its complexity and to improve its efficiency in order to address the short-term wind speed forecasting. An hybrid short-term wind speed forecasting approach with Empirical mode decomposition (EMD) and RVFLN is proposed in Jalli et al. (2020). Also, ELMs have been used as regressors in wrapper feature selection mechanisms (Salcedo-Sanz et al., 2018) within ML approaches for short term wind speed forecasting (Salcedo-Sanz et al., 2014, 2015).

A factor of crucial importance for the wind power sector lies in the inevitable occurrence of extreme events, i.e. Extreme Wind Speeds (EWS), which represent a relatively brief but a highly intensive peak in wind speed, often responsible for the worst damages caused by wind, especially in wind farms facilities. In fact, wind farms must be restrained from their operations during such events in order to minimize the hazards caused by them. Thus, having a detailed knowledge, as well as robust and reliable assessments, to estimate the frequency and intensity of EWS is essential, not only to avoid wind turbine damage but also to minimize the cut-out events (Outten and Sobolowski, 2021).

1.2. Literature review

Several methods can be found in the literature to address the prediction of EWS, from the application of classical techniques (Palutikof et al., 1999; Burlando et al., 2014) to modern techniques (Wang et al., 2015; Sheridan, 2018) and these methods include Machine Learning (ML) approaches as well (Schulz and Lerch, 2021). In Sallis et al. (2011), the authors explore different machine learning approaches,

including ANNs, for predicting the wind extremes based on simple meteorological variables. Predicting atmospheric extremes events, such as EWS, requires one to overcome the difficulties associated with the wind speed databases that present a high degree of skewness and unbalancing since the number of instances with extreme wind speeds often represents a minimum percentage of the total dataset. This problem has been mostly explored in the context of classification tasks (López et al., 2013), for example, in Kretschmar et al. (2004) the potential of ANN classifiers based on lagged EWS data and the ECMWF analysis data from 24 h previous to predict wind extremes was evaluated. However, the challenge we are tackling in this paper concerns a continuous predictive domain: in addition to forecasting the presence or absence of EWS, it is also important to provide a reasonable estimation of its magnitude. In Wang et al. (2020) a probabilistic approach is presented to forecast EWS using ensemble learning by combining three different ML models.

In order to address this issue, the most popular strategies can be categorized into three types: preprocessing, cost-sensitive learning, and ensemble methodology. While under-sampling and oversampling preprocessing have been extensively used to balanced data (Batista et al., 2004), the ensemble methodology involves a decision-making process that combines the individual learning algorithms and their outcomes in parallel to obtaining the ultimate accurate result. In this context, different strategies applying an ensemble architecture for classification and regression problems can be found in Ren et al. (2016). In Chen et al. (2018) an ensemble of Long Short-Term Memory (LSTM) deep learning models has been implemented to forecast wind speed time series. In Farahbod et al. (2022) ensemble of residual regression deep networks was designed to deal with wind speed prediction problems. In Nourani et al. (2020), two linear and two non-linear ensemble models were developed by combining outputs of three AI-based regressors for the prediction of vehicular traffic noise. In Liu et al. (2021) a “decomposition and ensemble” framework with Variational Mode Decomposition (VMD) and ANN is applied for predicting product futures prices. In Qiu et al. (2018) a hybrid incremental learning approach composed of Discrete Wavelet Transform (DWT), EMD and RVFL is presented for short-term electric load forecasting. Among the different methods proposed, a fuzzy ensemble method have been chosen to be applied in this paper, incorporating fuzzy sets to perform a soft division of the training data. Similar architectures have been proposed in literature: in Saha et al. (2015) a fuzzy clustering-based ensemble is used to weight prediction of individual models to predict Indian Monsoon. In Sideratos et al. (2020) a fuzzy clustering method is used to classify input data into more than one cluster, and prediction of national power load is performed using ensemble radial basis function ANN. In Prado et al. (2020) an ensemble of 8 ML regressors including ANN and Adaptive Neuro Fuzzy Inference System (ANFIS) is performed on the basis of a genetic algorithm (GA) which assigns weights to models forecasts. In Gao et al. (2022) a ensemble based on envelope decomposition method and fuzzy inference evaluation of predictability was proposed to predict wind speed. Furthermore, fuzzy-based partition of the training data has also established as an important area of research (Škrjanc et al., 2019; Yasunori et al., 2009; Höppner and Klawonn, 2003).

Different approaches combining ELM and RVFL ensembles with fuzzy concepts have been proposed in the literature: in Chen et al. (2021), the carbon future prices are predicted by applying ELM methods and ensemble empirical mode decomposition (EEMD) using fuzzy entropy. In Zhai et al. (2018a) an ensemble learning method based on dropout technique is proposed to enhance ELM prediction instability and architecture selection. In Zhai et al. (2018b) a fuzzy integral-based ELM ensemble methodology is proposed to tackle imbalanced classification. In Xu et al. (2020) a novel ensemble strategy based on generalized fuzzy soft sets theory is proposed to make the imbalanced training data set balanced for different ELM classifiers. In Cao et al. (2020, 2017) fuzzy theory for semi-supervised learning is employed to improve the generalization performance of RVFL on classification problems. Finally, in Malik and Yadav (2021) an RVFL ensemble learning method is proposed using Rotation Forest (RoF) as ensemble algorithm.

1.3. Contribution and paper organization

In this paper, we propose a new methodology for accurate prediction of EWS events. Specifically, the proposed algorithm for EWS prediction includes a fuzzy-based cascade ensemble of regression models, where different regressors are fitted with a particular portion of the training data, depending on the target variable value. In this way, as each regressor is focused on a specific part of the target domain, no data balancing techniques are required, and our proposal gives a higher importance to how the models are ensembled. In our proposal, the training data are partitioned into fuzzy-soft subsets, where each sample is assigned to a pertinence value for each subsets, in accordance with the corresponding membership function, dependent on the target variable value. The shapes of the membership functions play an essential role in the method's performance, along with other hyperparameters, which are optimized by means of a Differential Evolution (DE) algorithm. Two random neural network methods (ELM and RVFL) are considered as final regression methods, due to their extremely fast training speed, which becomes a key parameter in an ensemble approach with a significant number of models to be trained. The proposed methodology has been tested on three time-series databases collected on three medium-sized wind farms located in different areas of Spain, outperforming different state-of-the-art methods for time series regression problems in terms of prediction on extreme events. Methods used as benchmarks include Machine Learning (ML) approaches, synthetic data generation for balance training data (SMOBN) and Deep Learning (DL) methods for time series forecasting.

The rest of the manuscript is organized as follows. First, Section 2 describes the novel methodology proposed in this paper. Then, specific problem definition and databases descriptions are shown in Section 3. Section 4 presents the results obtained for the extreme wind speed forecasting problem. Finally, some discussion and conclusions about the results are given in Section 5.

2. Proposed methodology

This section details the proposed methodology with a description of the complete architecture and each of its components, including the data clustering, the prediction procedure and an evolutionary optimization of the system to improve its performance.

2.1. Architecture for EWS prediction

Fig. 1 illustrates the architecture of the proposed framework. First, the training data partitioning takes place. Here, fuzzy-soft clusters are formed based on the target variable value. Specific details on the cluster formation procedure may be found in Section 2.2. Then, each cluster is used to train a different regression model, in a way that each model is focused on a specific range of the training data. Afterwards, in order to calculate the forecast value of an incoming test sample, input data are fed to each regressors independently, deriving in a prediction value ($\hat{y}_{\mathcal{M}_i}$) for each model. These individuals predictions are subsequently introduced to a fuzzy-based ensemble, where the previous layer prediction, together with the membership function shapes, are used to determine the pertinence values of each prediction. The ensemble output is further weighted with the previous layer's forecast value, on the basis of an hyperparameter named as learning rate (ϵ). Details on the implemented prediction procedure are described in Section 2.3. Being a layered-based structure, this framework is repeated sequentially. Besides, the parameter m defines the number of groups in which each cluster is divided, so the total number of subgroups belonging to a layer n would be equal to m^n . An example of the framework operation showing the first two layer is provided in Fig. 1 for a value of $m = 3$.

2.2. Data clustering

The developed method consists of a cascade architecture divided into several layers, where the clustering process is performed in each of them. The parameter m defines the number of groups to divide each cluster of the previous layer (or the total set of training data in the initial layer) into. The clustering is performed according to the target value variables in the following procedure: (1) data from a given subset are sorted according to the target value; (2) the shape of m membership functions is defined, one associated with each new data subset; (3) the bases of these membership functions divide the target domain (in percentiles) of a specific subset into m regions; (4) new data clusters are formed from each subset according to this domain division.

Fig. 2 illustrate an example of the membership functions used. It is noteworthy to point the following: first, as shown in the figure, membership functions may overlap with each other, therefore, some training data will be shared by more than one model. Second, the same membership functions are used to split the data subsets in all the layers of the framework. Whenever a data subset is to be divided into m groups, the same percentiles values are used for the data division. For example, in the case shown in Fig. 2 for a value of $m = 3$, all the data subsets of a specific layer are divided into three groups for the subsequent layer: one including the training data with a target comprised between the 0 and 40th percentile of that specific cluster, other with values between the 25th and 75th percentile, and the last one comprising data between the 60th and the 100th percentile. This means that the determination of the membership function shapes represents a critical parameter that will determine the success or failure of the algorithm. For this purpose, a powerful evolutionary optimization algorithm has been employed to obtain the most optimal membership functions shapes. In all cases, triangular membership functions as the ones shown in Fig. 2 are considered.

2.3. Prediction

The presented framework begins with the prediction of an initial regression model (\mathcal{M}_0) trained with all the training data without performing any clustering (Eq. (1)), which corresponds to the prediction of Layer 0. Afterwards, the output of a specific layer is computed as the average of the prediction of each model, weighted by its pertinence value, as can be seen in Eqs. (2), (3) and (4) for the first, second and n layers, respectively. In this equations, i lists all the regression models (\mathcal{M}_i) belonging to a layer, $\hat{y}_{\mathcal{M}_i}$ denotes the output of \mathcal{M}_i , p_i denote the pertinence value of the evaluated sample for \mathcal{M}_i , p_i^* represents the pertinence value of the parent model of \mathcal{M}_i and m represents the number of clusters formed on each division.

As shown in Fig. 1, the pertinence values of each layer are obtained according to the previous layer prediction (\hat{y}_i) and the membership functions associated to each model. For this purpose, the membership functions are computed in terms of the target value (instead of the cluster percentiles). Fig. 3 illustrates those membership functions for the first two layers and using the percentiles values shown in Fig. 2 to perform the division. Here, entering in those graphics through the x -axis with the value of the previous layer prediction, a pertinence value comprised between 0 and 1 is assigned to each individual model. Also, the pertinence values corresponding to a layer are used in the subsequent layer as p_i^* . I.e.: pertinence value of \mathcal{M}_1 in layer 1 (p_1) is used as parent pertinence value of models $\mathcal{M}_{(1,1)}$, $\mathcal{M}_{(1,2)}$ and $\mathcal{M}_{(1,3)}$ in layer 2.

In addition, a learning rate (ϵ) is defined, aiming at not losing the generality of the prediction. Thus, for each layer, the previous layer output is taken into account by a factor of $1 - \epsilon$.

$$\hat{y}_0 = \hat{y}_{\mathcal{M}_0} \quad (1)$$

$$\hat{y}_1 = \frac{(\hat{y}_{\mathcal{M}_1} \cdot p_1 + \hat{y}_{\mathcal{M}_2} \cdot p_2 + \hat{y}_{\mathcal{M}_3} \cdot p_3)}{p_1 + p_2 + p_3} \cdot \epsilon + \hat{y}_0 \cdot (1 - \epsilon) \quad (2)$$

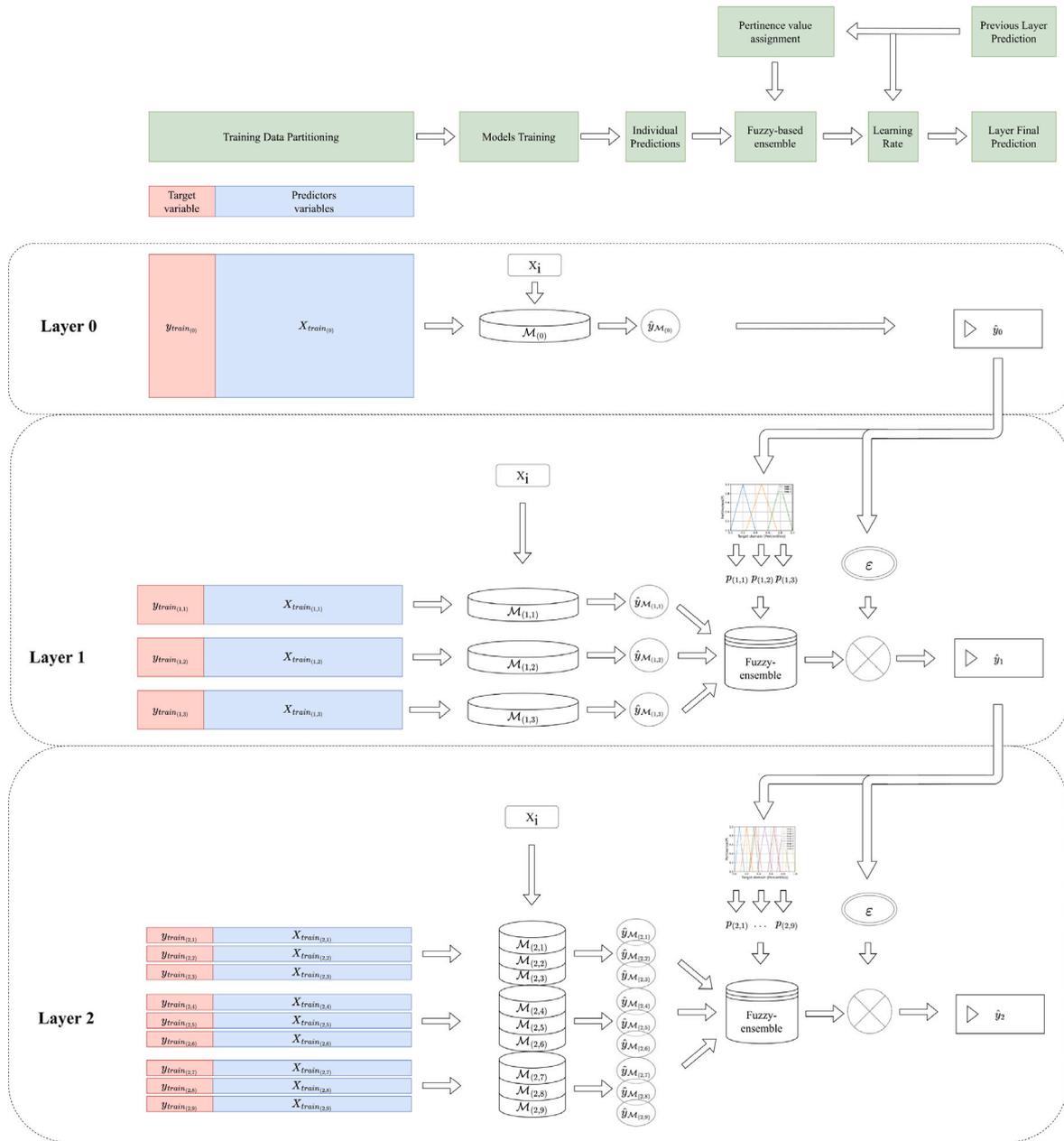


Fig. 1. Fuzzy-based cascade ensemble of regressors architecture (m = 3).

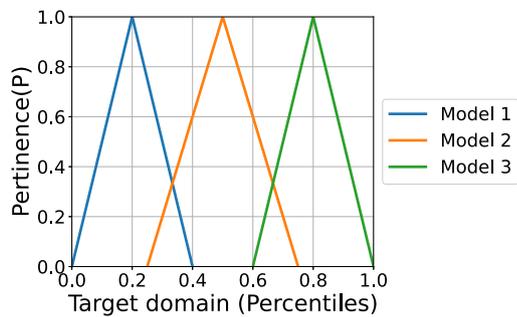


Fig. 2. Example of triangular membership functions.

$$\hat{y}_n = \frac{\sum_{i=1}^{m^n} (\hat{y}_{M_i} \cdot p_i \cdot p_i^*)}{\sum_{i=1}^{m^n} p_i \cdot p_i^*} \cdot \epsilon + \hat{y}_{n-1} \cdot (1 - \epsilon) \tag{4}$$

2.4. Evolutionary optimization algorithm for hyperparameter tuning

Given the sensitivity of the factors affecting the predicted value, such as the number of membership functions (m), the shape of these functions, and the learning rate used (ε), an evolutionary algorithm has been applied in order to find an optimal set of values for these parameters.

For this purpose, the full training data (or 70% of the total data) have been divided into a specific training and a specific validation sets (i.e., 75% and 25%, respectively), and the error encountered in the prediction of these validation data has been used as the fitness function of the optimization algorithm.

Isosceles triangular functions have been used as membership functions, starting always the first model membership function in 0, and

$$\hat{y}_2 = \frac{\sum_{i=1}^9 (\hat{y}_{M_i} \cdot p_i \cdot p_i^*)}{\sum_{i=1}^9 p_i \cdot p_i^*} \cdot \epsilon + \hat{y}_1 \cdot (1 - \epsilon) \tag{3}$$

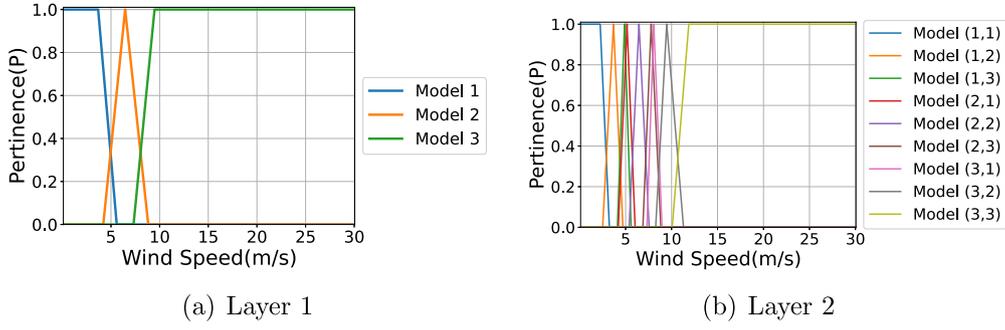


Fig. 3. Example of triangular membership functions for Layers 1 and 2.

finishing the last model membership function in 1. Therefore the evolutionary algorithm optimizes the position and the base width of these functions, along with the learning rate. Cases with 2, 3 and 4 membership functions have been tested, and thus for each case, the number of variables to optimize are 3, 5 and 7, respectively. For example, in the case shown in Fig. 2 for $m = 3$, the evolutionary algorithm would have returned the values: 0.25, 0.4, 0.6, 0.75 and ϵ .

Since no constraints have been added to force the overlapping of the membership functions, it may be the case that a value of the target variable domain has an associated pertinence value of 0 for all models. In order to prevent errors caused by a division by 0, the term $0.001 \cdot \hat{y}_0$ has been added in Eq. (4) for computing the predicted value in layer n . This means that for each sample, the prediction of model 0 is taken into account by a small factor, so that if all the membership values are 0 for a specific sample, this factor will be the only one to be considered, and the prediction of that point will be the one provided by the initial model. The updated prediction equation is shown in Eq. (5).

$$\hat{y}_n = \frac{(\sum_{i=1}^{m^n} (\hat{y}_{M_i} \cdot p_i \cdot p_i^*)) + 0.001 \hat{y}_0}{(\sum_{i=1}^{m^n} p_i \cdot p_i^*) + 0.001} \cdot \epsilon + \hat{y}_{n-1} \cdot (1 - \epsilon) \quad (5)$$

As previously mentioned, a Differential Evolution (DE) (Storn and Price, 1997) algorithm has been implemented to address the optimization problem. DE is a stochastic population-based method designed for global optimization problems (Leon and Xiong, 2014). It maintains a population with N_p individuals, where every individual within the population stands for a possible solution to the problem. Individuals are represented by a vector $X_{i,g}$, where $i = 1, \dots, N_p$ and g refers to the index of the generation. A DE cycle consists of three consecutive steps: mutation, crossover and selection, which are described as follows.

Mutation is carried out to generate random perturbations on the population. For each individual, a mutant vector is generated. There are different approaches for DE mutation in literature (Price et al., 2006). In this paper, the best mutation strategy (Xu and Wen, 2012) has been applied. It attempts to mutate the best individual of the population, according to Eq. (6), where $V_{i,g}$ denotes the mutated vector, i is the index of the vector, g stands for the generation index, $r_1, r_2 \in 1, \dots, N_p$ are randomly created integers, $X_{best,g}$ denotes the best solution in the population and F is the scaling factor in the interval $[0, 2]$. This mutation strategy uses the scaled difference between two randomly selected vectors to mutate the best individual in the population. Fig. 4 shows how a new mutant vector is obtained with this strategy, where d denotes the difference vector between $X_{r1,g}$ and $X_{r2,g}$.

$$V_{i,g} = X_{best,g} + F \cdot (X_{r1,g} - X_{r2,g}) \quad (6)$$

Crossover carries out the combination of every individual with the mutant vector created in mutation stage. The new solutions created are called trial vectors and are denoted by $T_{i,g}$ for individual i at generation g . Every parameter in the trial vector are decided following Eq. (7), where j represent the index of every parameter in a vector, CR is the probability of recombination, and J_{rand} denotes a randomly selected

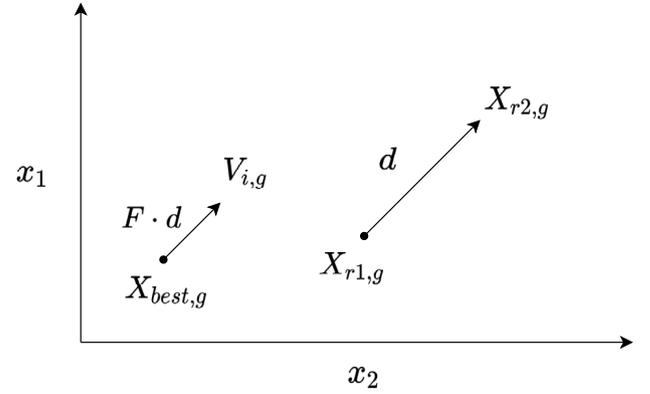


Fig. 4. Best mutation with one difference vector.

integer within $(1, \dots, N_p)$ to ensure that at least one parameter from mutant vector enters the trial vector

$$T_{i,g}[j] = \begin{cases} V_{i,g}[j] & \text{if } rand[0, 1] < CR \text{ or } j = J_{rand} \\ X_{i,g}[j] & \text{otherwise} \end{cases} \quad (7)$$

Lastly, the selection consists of comparing each trial vector with its parent solution and further decides the winner to survive into the next generation. The individuals for the new generation are chosen following Eq. (8), where $T_{i,g}$ is the trial vector, $X_{i,g}$ the individual in the population, $X_{i,g+1}$ denotes the individual in the next generation and $f()$ represents the fitness value of the corresponding individual.

$$X_{i,g+1} = \begin{cases} T_{i,g} & \text{if } f(T_{i,g}) < f(X_{i,g}) \\ X_{i,g} & \text{otherwise} \end{cases} \quad (8)$$

DE algorithm was implemented following the pseudo-code showed in the algorithm 1 and the differential evolution function from Python library `scipy.optimize` was used.

2.5. Regression methods

In order to ensure an efficient system performance, regression methods that enables a fast and efficient training and prediction process was chosen as the number of models employed (m^n) can be significantly increased when setting a high number of layers (n) or models per layer (m). In this context, randomized neural networks methods, which represent a specific type of training method for MLPs characterized as a computationally efficient method compared to the traditional gradient back-propagation, were adopted.

2.5.1. ELM

ELM regression method (Huang et al., 2006), is based on the concept that if the MLP input weights are fixed to random values, the MLP can be considered as a linear system and the output weights can be easily

Algorithm 1 Differential evolution algorithm

Initialize variables:
Crossover Rate (CR)
Mutation Rate (F)
Population Size (PS)
Maximum Iterations (maxIter)
Bounds \implies (0.05,0.95)
Initialize the population with randomly created individuals
Compute fitness value for all individuals
for iter to maxIter do
 Create mutant vector (Equation (6))
 Create trial vectors (Equation (7))
 Compute fitness values of trials vectors.
 Select winning vectors for the next generation (Equation (8))
end for

obtained using the pseudo-inverse of the hidden neurons outputs matrix H for a given training set (Huang et al., 2011). Algorithm 2 shows a summary of the ELM implementation, where the number of neurons in the hidden layer has been set to 500 ($M = 500$). Fig. 5(a) illustrates the structure of this type of regression networks.

Algorithm 2 Extreme Learning Machine (ELM)

Given a training set $D = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbb{R}^n, \mathbf{t}_i \in \mathbb{R}^m, i = 1, \dots, N\}$,
an activation function f and an hidden neuron number M ,

1: Assign arbitrary input weights \mathbf{w}_j and biases $\mathbf{b}_j, j = 1, \dots, M$

2: Compute the hidden layer output matrix \mathbf{H} ,

$$\text{where } \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_M, \mathbf{b}_1, \dots, \mathbf{b}_M, \mathbf{x}_1, \dots, \mathbf{x}_N) = \begin{bmatrix} f(\mathbf{w}_1 \cdot \mathbf{x}_1 + \mathbf{b}_1) & \dots & f(\mathbf{w}_M \cdot \mathbf{x}_1 + \mathbf{b}_M) \\ \vdots & \dots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x}_N + \mathbf{b}_1) & \dots & f(\mathbf{w}_M \cdot \mathbf{x}_N + \mathbf{b}_M) \end{bmatrix}_{N \times M}$$

3: Calculate the output weight matrix $\mathbf{B} = \mathbf{H}^\dagger \mathbf{T}$,

$$\text{where } \mathbf{B} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{M \times m} \quad \text{and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}$$

2.5.2. RVFL

RVFL method (Shi et al., 2021) is based on the structure of Single Hidden Layer Feed Forward Networks (SLFNs), with the particularity that the weights and biases of the neurons in the hidden layer are initialized randomly, and their values are kept fixed and do not need to be tuned during the training stage. Therefore only the output layer weights need to be computed to achieve the lowest estimation error, these values can be calculated by the Moore–Penrose pseudo-inverse. Fig. 5(b) shows the RVFL architecture, where the direct link between the input layer and the output layer is an effective and simple regularization technique preventing RVFL networks from overfitting (Zhang and Yang, 2020).

Note that there are small differences between ELMs and RVFLs algorithms. Both methods have a similar internal structure, except that in the case of ELMs the direct links between input and output are omitted. In fact, both methods are interesting in terms of learning capability. Thanks to their good performance in different problems, it is only a matter of deciding which in favor of the two is better suited for a specific task.

3. Data and predictive variables

This section describes the data available for this research. First, we describe the wind speed data from 3 wind farms in Spain. Then, the predictive variables from the ERA5 Reanalysis are described.

3.1. Wind speed data

Three medium-size wind farms located in Spain have been selected, whose locations can be seen in Fig. 6, labeled as “A”, “B” and “C” in the map. Note that the wind farms selected cover different parts of Spain, north, center and south, characterized by different wind regimes. Time series with hourly wind speed data covering a period of 10 years (2003–2013) in all wind farms have been used for training and validation of the models.

Fig. 7 depicts the histograms for the three wind farms studied, showing a distribution centered at 5 m/s. One may observe the long-tailed distribution for high speeds. It is precisely these EWS that we are particularly interested in accurately forecasting in this research. This can help anticipate the possibility of damage to the infrastructure as well as providing us an ability to predict potential extreme scenarios that could lead to the stoppage of the turbine.

3.2. Predictive variables

The EWS prediction presented in this paper is carried out based on meteorological data from ERA5 reanalysis (Hersbach et al., 2018). ERA5 provides hourly information on meteorological variables related to temperature, pressure, precipitation and snowfall among many others, with a resolution of 0.25 degrees of longitude and latitude in a regular grid. Note that ERA5 has been spotted as one of the most reliable Reanalysis in different works comparing some of these products for different meteorological variables, including studies about wind speed and wind energy potential (Ramon et al., 2019; Samal, 2021). This makes ERA5 an invaluable source of data for both research and practical applications in the fields of meteorology, climatology and weather forecasting. Additionally, the high resolution of ERA5 also means that it is able to provide more accurate results for local weather phenomena, such as storms and other localized weather events. Furthermore, ERA5’s high resolution and reliability makes it a particularly useful tool for accurately forecasting extreme weather events, allowing for better preparation and mitigation plans to be developed as well as its importance in wind energy engineering applications.

To tackle the EWS prediction problem, the predictive variables comprised of temperature, pressure, speed of different wind components at different heights, and the proportion of the grid box covered by clouds. Table 1 lists the 14 predictive variables selected per node. For the wind speed forecasting in each of the wind farms, five geographical nodes have been selected, located at the corresponding farm coordinates within 100 km to the north, south, east and west, respectively. Therefore, a total of $N = 70$ predictor variables (inputs) have been considered for each case under study. Fig. 8 shows the correlation values between target and predictive variables for each case, considering the geographic node corresponding to each wind farm location. Several conclusions can be drawn from this Figure: (1) as expected, in all cases the variables most correlated with the target are those related to wind. Also, it is noteworthy to mention that the most significant wind component varies according to the latitude of the wind farm: while in the north of the Peninsula (cases “A” and “B”) the ‘u’ component exhibits a strong positive correlation whereas the ‘v’ component has nearly zero correlation, in the south of the Peninsula (case “C”) the correlation of the target with the u-component reanalysis variables is negative, while with the v-component ones is very strongly positive; (2) pressure-related variables exhibit in all cases a negative correlation with the target variable, with the magnitude of this correlation decreasing as the wind farm is further south; (3) the four variables related to cloud cover show a positive correlation in all cases; and (4) temperature-based variables appear to be of limited importance in databases “A” and “C”, with correlation coefficients very close to 0, while in database “B” the correlation is more significant and of negative sign.

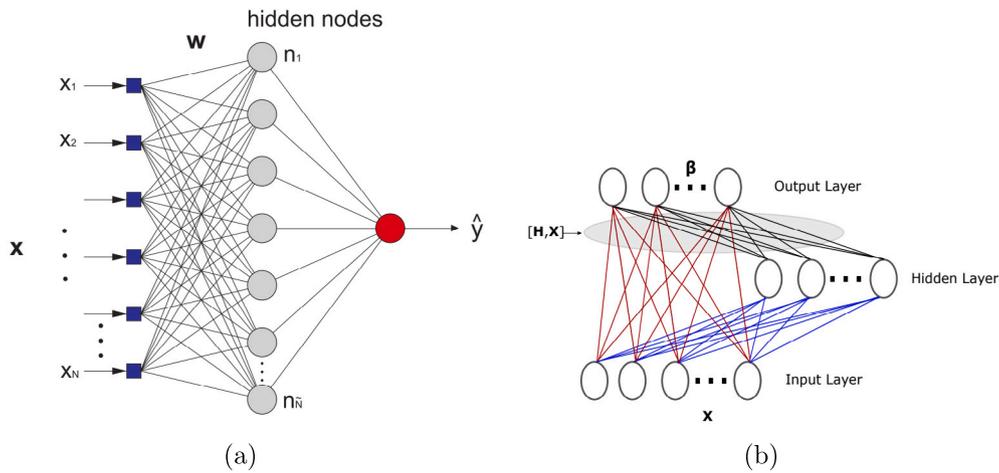


Fig. 5. Structure of (a) an ELM network; (b) an RVFL.

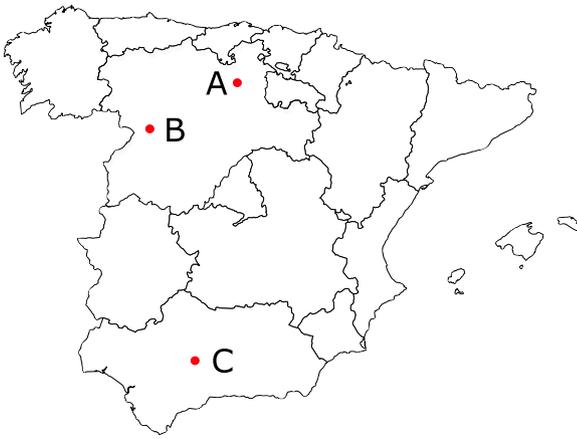


Fig. 6. Geographical location of the wind farms (labeled as “A”, “B” and “C”).

The preliminary step in the processing of these time series entailed the elimination of the data corresponding to missing target values. Following this, the prediction time-horizon is set, and a shift is performed accordingly on the wind speed data (target) with respect to the input variables. That is, for the prediction horizon set to be 1 h, we use x_t to predict y_{t+1} whereby the target value is shifted one row (1 h) in the timed dataset.

4. Experiments and results

4.1. Evaluation metrics

In order to assess the performance of the EWS prediction models, different evaluation metrics were used. Generic regression error metrics together with specific metrics that reflect the models performance in the prediction of extreme wind events were utilized.

4.1.1. Generic regression metrics

First, two common metrics for regression problems, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), has been considered:

$$MAE = \frac{1}{N} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

Table 1
Predictive variables considered at each node from the ERA-5 reanalysis dataset.

Variable name	ERA5 Variable
d2 m	2 m dewpoint temperature
t2 m	2 m temperature
sp	Surface pressure
msl	Mean sea level pressure
u10	10 m u-component of wind
u10n	10 m u-component of neutral wind
u100	100 m u-component of wind
v10	10 m v-component of wind
v10n	10 m v-component of neutral wind
v100	100 m v-component of wind
hcc	High cloud cover
mcc	Medium cloud cover
lcc	Low cloud cover
tcc	Total cloud cover

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

where \hat{y} represents predicted values (provided by the proposed model) and y are the actual observed values. The subscript i is used to refer to a single sample $y_i = y[i]$.

4.1.2. Extreme events specific error metrics

Here, we also evaluate the prediction of extreme events as the accurate prediction of such data points is precisely the scope of this research work. For this purpose, it is first necessary to define the criteria for deciding which samples are considered as extremes. Among the different methods presented in the literature, such as (Palutikof et al., 1999) and An and Pandey (2005), two are considered in this work, based on extreme-values theory: the Peak Over Threshold (POT) (Simiu and Heckert, 1996) and the Method of Independent Storm (MIS) (Cook, 1982; Harris, 1999).

The POT method considers several of the largest order statistics exceeding a sufficiently high threshold in the collected data. An asymptotic distribution, the generalized Pareto distribution (GPD), is used to describe the behavior of the events above the specified threshold. For sufficiently high thresholds, the number of observations above threshold per year (the crossing rate) is low, and Poisson-distributed. The cumulative distribution function for the GPD is

$$F(x) = 1 - \left[1 - \frac{k}{\alpha}(x - \xi)\right]^{1/k} \quad (11)$$

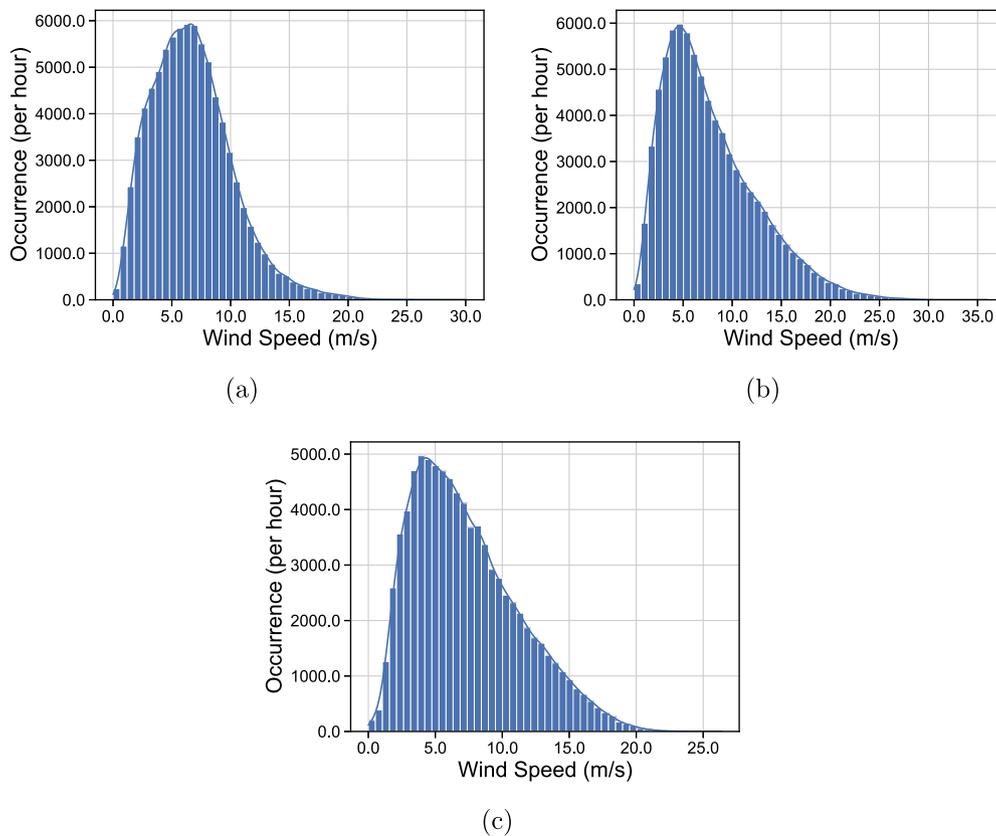


Fig. 7. Wind speed histogram.

	d2m	hcc	lcc	mcc	msl	sp	t2m	tcc	u10	u10n	u100	v10	v10n	v100
Target A	-0.048	0.032	0.15	0.15	-0.24	-0.27	-0.0026	0.16	0.27	0.29	0.28	0.051	0.056	0.061
Target B	-0.12	0.039	0.16	0.11	-0.1	-0.15	-0.17	0.12	0.13	0.11	0.13	-0.044	-0.039	-0.043
Target C	0.051	0.095	0.09	0.08	-0.083	-0.086	0.075	0.12	-0.25	-0.21	-0.25	0.52	0.54	0.58

Fig. 8. Correlation matrix between target and predictive variables for each wind farm.

where ξ is the selected threshold, i.e., the values of $x - \xi$ are the exceedances.

Extreme value theory rests on the assumption of independence in the underlying observations. For POT method, independence requires a suitable combination of both threshold and minimum separation time between events. With a high threshold, the separation can be reduced without compromising independence, but with a low threshold the separation must be increased if independence is to be maintained (Walshaw, 1994) (see Fig. 9).

On the other hand, the MIS method examines a continuous time series of wind speed records to identify storms. The downward crossings of the wind speed below a chosen threshold define the beginning of the “lulls” (period of wind speeds below a selected low threshold) in the record, and the wind speed records between each pair of lulls are considered to be part of an independent storm. The maximum speed in each independent storm is selected to form a sample of extreme values. This method ensures the independence of the extremes by separating the parent time series of wind speeds into independent storms, and then selecting the highest value from each storm. The storms are separated by a lull. Fig. 11 shows the differences between POT and MIS extreme events selection for a selected threshold.

Therefore, the selection of an optimal threshold is a key point in the definition of extremes with both methods, especially in the case

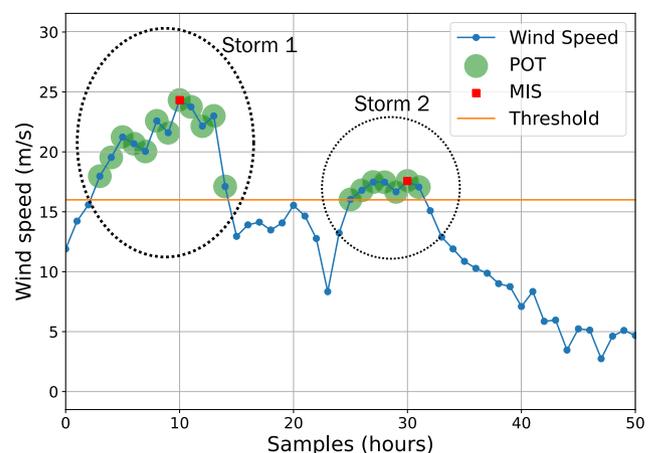


Fig. 9. Illustration of difference between MIS points and POT points.

of POT, which exhibits large variation with respect to the threshold speed. On one hand, the threshold must be set high enough that only true peaks, with Poisson arrival rates, are selected. If this is not the

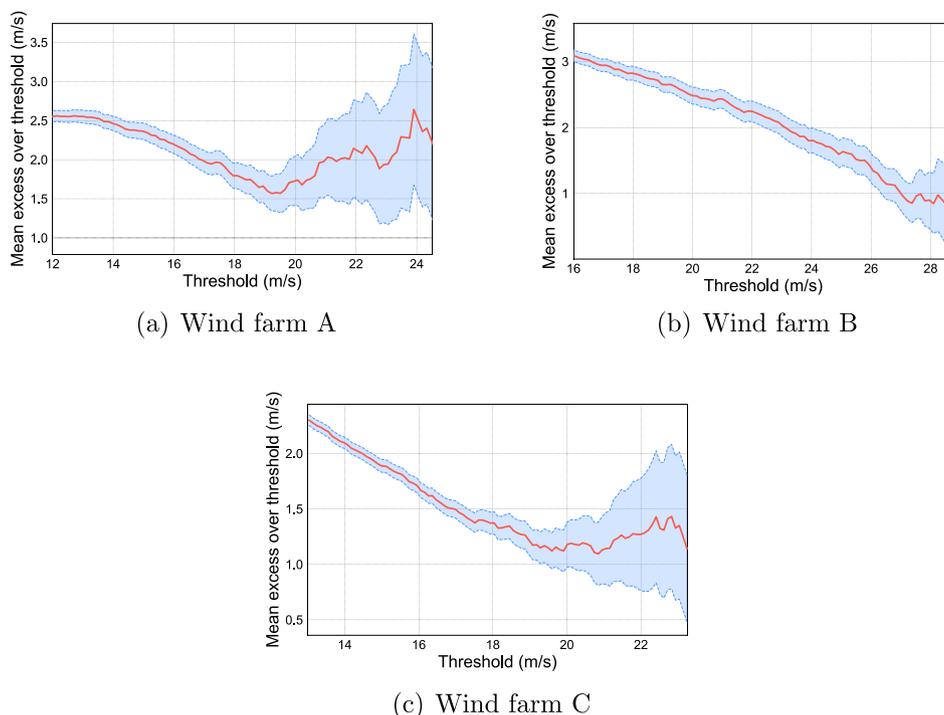


Fig. 10. Conditional mean exceedance graphs.

case, the distribution of selected extremes will fail to converge to the GPD asymptote. On the other, the threshold must be set low enough to ensure that enough data are selected for satisfactory determination of the distribution parameters (Abild et al., 1992). Such high threshold sensitivity is recognized as a limitation of POT method, as it makes it difficult to identify a representative quantile value. An and Pandey (2005). Several techniques are present in the literature in order to aid threshold selection. For example, conditional mean exceedance (CME) graphs, also known as mean residual life graphs, plot the mean excess over threshold as a function of threshold (Davison, 1984; Kilmister, 1990). For a GPD distributed spectrum, the CME graph should plot as a straight line. An appropriate threshold can be chosen by selecting the lowest value above which the graph is a straight line. Fig. 10 illustrates the CME graphs for the three cases considered, following these graphs, the threshold for each database have been set at 19, 27 and 19 m/s, accounting for a 0.43%, 0.07% and 0.46% of the total number of events.

In the case of MIS, increasing threshold in MIS only reduces the number of independent values in the sample (Fig. 11). Here, the threshold value has been selected in accordance with Cook (Cook, 1982), who defined the annual rate of occurrence of independent storms ($r = N/R$) to be around 100 events/year, where N corresponds to the number of independent maxima identified from a collection of R wind years of record. This typical storm frequency rate around 100 events/year is also consistent with a prediction made independently by Davenport (Davenport, 1967) using a spectral method. Consequently, bearing in mind that the length of the wind records we are dealing with is 10 years, we set the threshold value so that 1000 independent maxima appear in each case, establishing values of 13, 16.2 and 14.2, respectively (Fig. 11).

Fig. 12 depicts the wind speed time series for each location, with the data divided into train (70) and test (30%). This figure also illustrates the thresholds defined to separate extreme data (outliers) from non-extreme data (non-outliers), using both the POT and the MIS methodologies. In order to measure the performance of the regressors on these outlier data, first, two new indicators, named Extreme Events Mean Absolute Error (EEMAE) and Extreme Events Root Mean

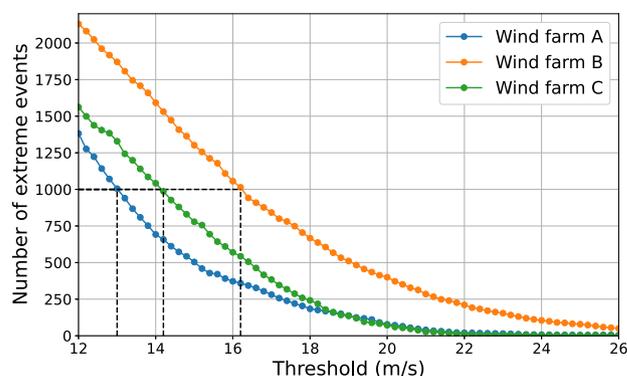
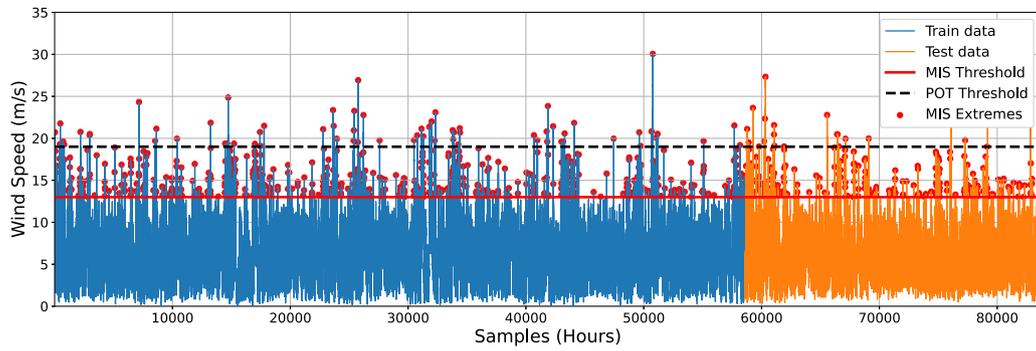


Fig. 11. Number of extreme events as a function of the threshold value.

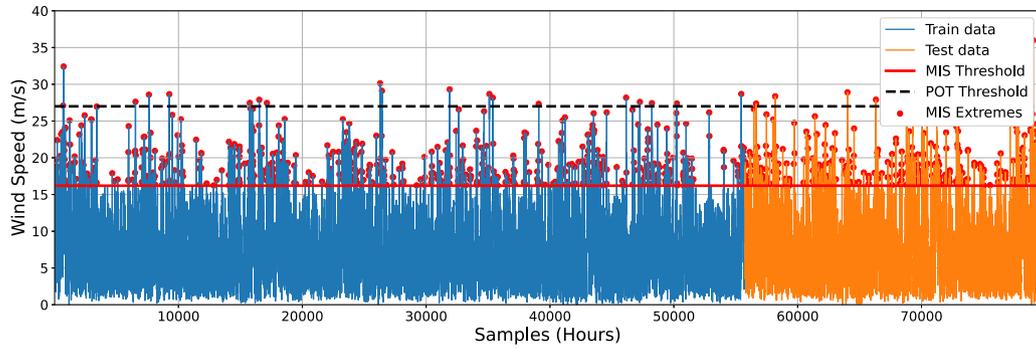
Square Error (EERMSE), have been used, which corresponds to the MAE (Eq. (9)) and RMSE (Eq. (10)) calculation on the MIS extreme points, accounting for the independent maxima of the wind speed series.

The main drawback in the metrics presented previously, EEMAE and EERMSE, is that they are only focused on the prediction of extremes values, but they do not penalize the situation where the model predicts an extreme, but the actual value is normal (False Positive). These false alarms may lead to severe economic damages, such as a disruption in energy production due to a extreme wind prediction that does not occur, or the deployment of emergency equipment to reinforce installations when it is not necessary. Two popular classification error metrics are used to account for these cases: True Positive Rate (TPR) and False Positive Rate (FPR).

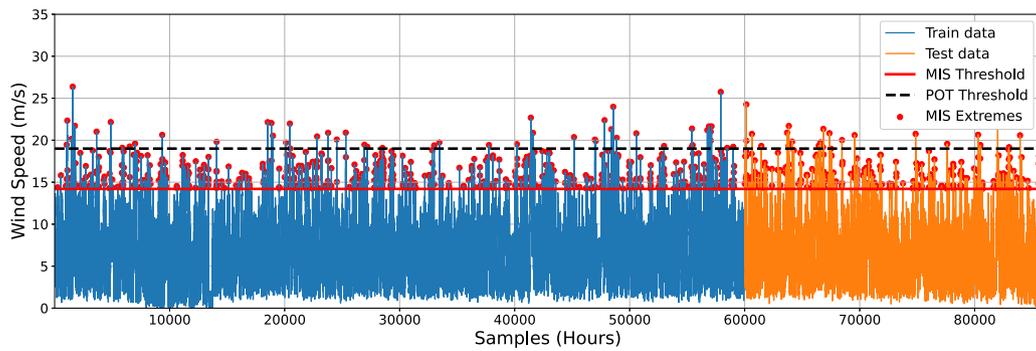
TPR, also referred to as Recall, determines the ability of a model to find all the relevant cases within a dataset. It is computed by dividing the number of relevant cases truly predicted, True Positives (TP), by the total number of relevant cases present in the data, Positives (P). In this context, since we are working with a regression model, we define a threshold above which both actual and predicted values are considered as extreme (or positive). Consequently, each sample of the test data set



(a) Wind farm A



(b) Wind farm B



(c) Wind farm C

Fig. 12. Wind speed time series considered.

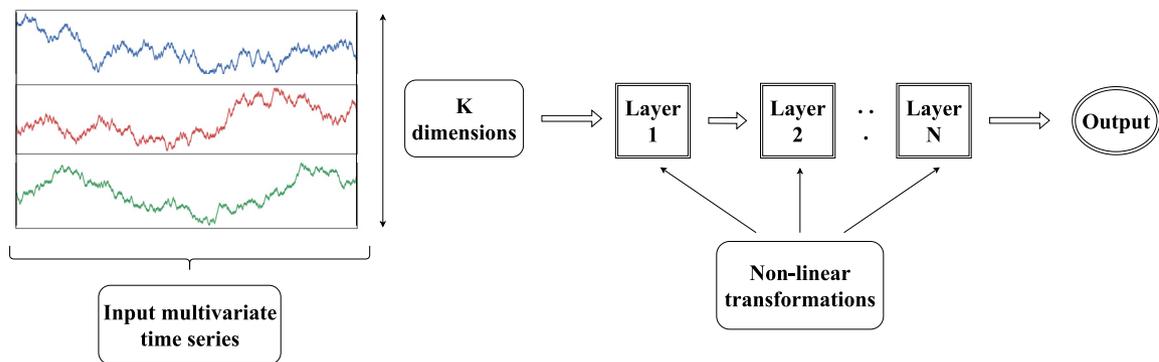


Fig. 13. Deep learning framework for time series forecasting.

is assigned with a boolean value of TP (1 if both the prediction and the actual value are above T , Eq. (12)) and P (1 if the actual value

is above T , Eq. (13)). Therefore TPR is computed following Eq. (14), where a value of 1 indicates that all extremes are predicted correctly

Table 2
Experimental setup.

	RT		RF		
		max depth	400	n estimators	400
ML Methods	SVR		LSSVR		
		regularization parameter	400	gamma	0.01
		epsilon	0.1	n estimators	400
	MLP		ELM		
		hidden layers	2	hidden size	500
		neurons per layer	64		
	activation	'relu'			
	solver	'adam'			
	RVFL				
	hidden nodes	2000			
	regularization parameter	0.001			
SMOBN	k neighbors	9	samp method	'balance'	
	rel thres	0.80	pert	0.02	
	rel method	'auto'	rel xtrm type	'both'	
	rel coef	1.50			
DL Methods	RNN		GRU		
		number of layers	2	number of layers	1
		neurons per layer	64	neurons per layer	64
	LSTM		1D-CNN		
		number of layers	2	number of CNN layers	1
		neurons per layer	64,32	filters	128
			kernel size	4	
DE Algorithm	CR	0.70	F	1.00	
	PS	10 · m	maxIter	10	

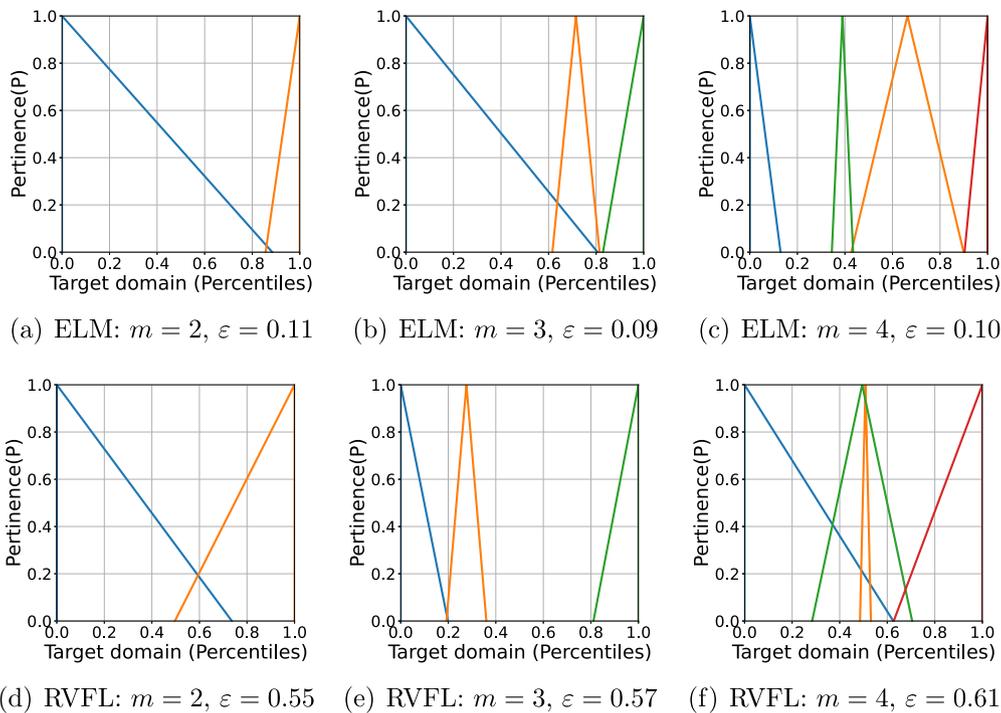


Fig. 14. Optimized membership functions for database A and different values of m .

and 0 denotes that none extremes have been anticipated. In this case, POT method is used to defined the events considered as extremes, since the MIS method complicates the classification of a prediction as a false alarm, as not all samples above the established threshold are considered extreme events. The threshold values are set, as commented previously, at 19, 27 and 19 m/s, for the three databases, respectively.

$$TP_i = \begin{cases} 0 & \text{if } \hat{y}_i \leq T \text{ or } y_i \leq T \\ 1 & \text{if } \hat{y}_i > T \text{ and } y_i > T \end{cases} \quad (12)$$

$$P_i = \begin{cases} 0 & \text{if } \hat{y}_i \leq T \\ 1 & \text{if } \hat{y}_i > T \end{cases} \quad (13)$$

$$TPR = \frac{\sum_{i=0}^N TP_i}{\sum_{i=0}^N P_i} \quad (14)$$

Similarly, the FPR is computed by dividing the number of False Positives (FP), i.e. the number of false alarms or events falsely predicted as extreme, by the number of Negatives (N), i.e. the sum of non-extreme events. According to Eqs. (12) and (13), a boolean value of FP and N is

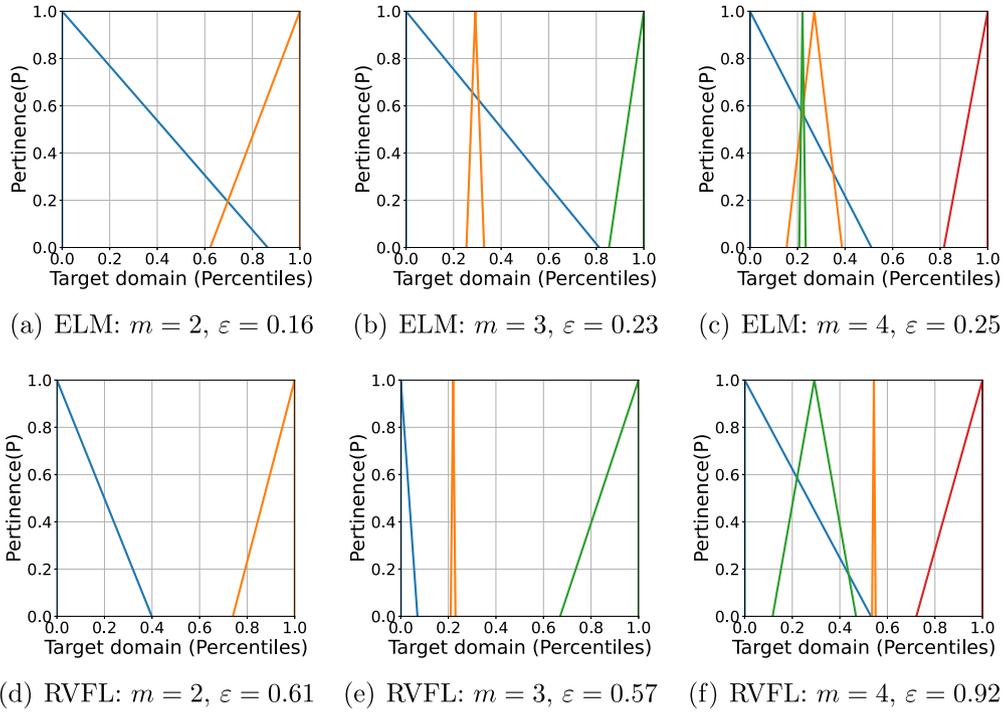


Fig. 15. Optimized membership functions for database B and different values of m .

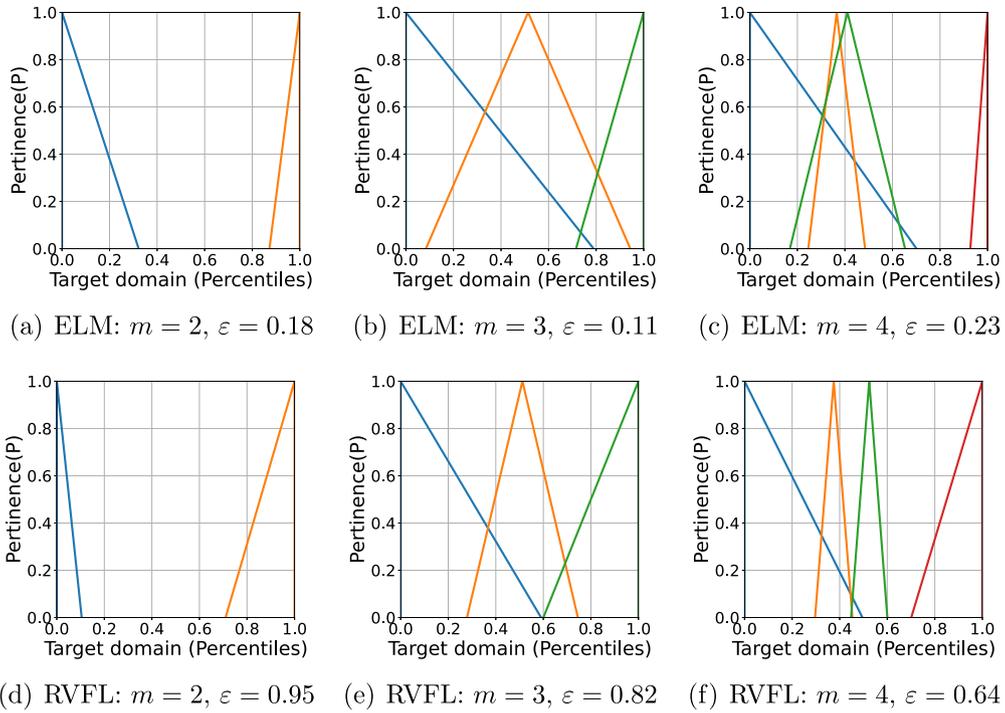


Fig. 16. Optimized membership functions for database C and different values of m .

given to each data set sample (Eqs. (15) and (16), respectively). Then FPR is calculated as shown in Eq. (17), where a value of 0 indicates that all non-extremes events have been predicted correctly, and 1 denotes that all of them were predicted as false extremes.

$$FPR = \frac{\sum_{i=0}^N FP_i}{\sum_{i=0}^N N_i} \quad (17)$$

$$FP_i = \begin{cases} 0 & \text{if } \hat{y}_i \leq T \text{ or } y_i > T \\ 1 & \text{if } \hat{y}_i > T \text{ and } y_i \leq T \end{cases} \quad (15)$$

$$N_i = \begin{cases} 0 & \text{if } \hat{y}_i > T \\ 1 & \text{if } \hat{y}_i \leq T \end{cases} \quad (16)$$

4.2. Algorithms for comparison

The performance of the novel fuzzy-based cascade ensemble have been assessed by employing different time series regression algorithms for comparison (benchmarks hereafter). Shallow ML methods on the

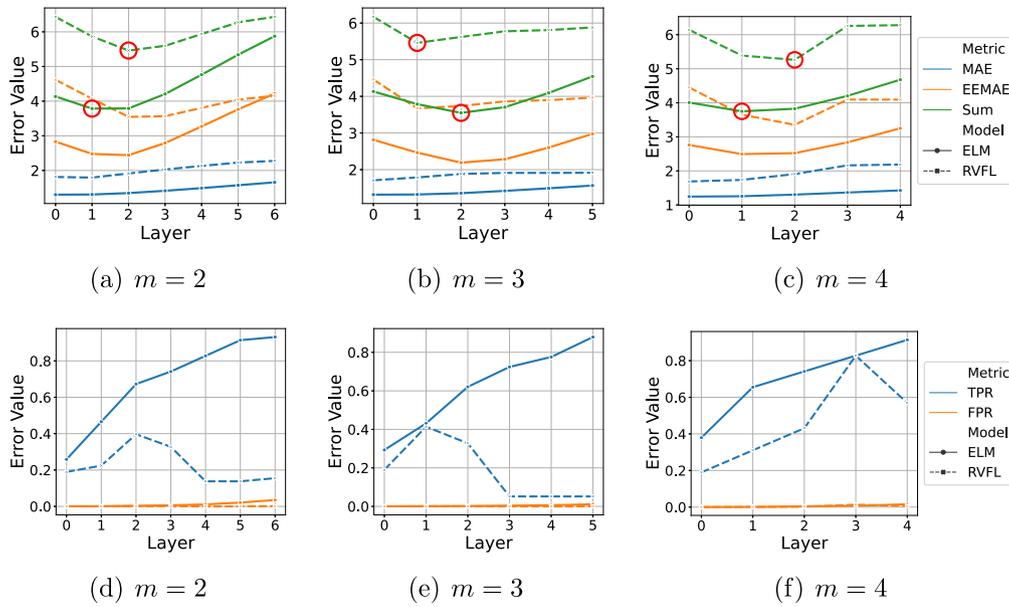


Fig. 17. Error metrics evolution on validation data when increasing the number of layers for $m = 2$, $m = 3$ and $m = 4$ in database A.

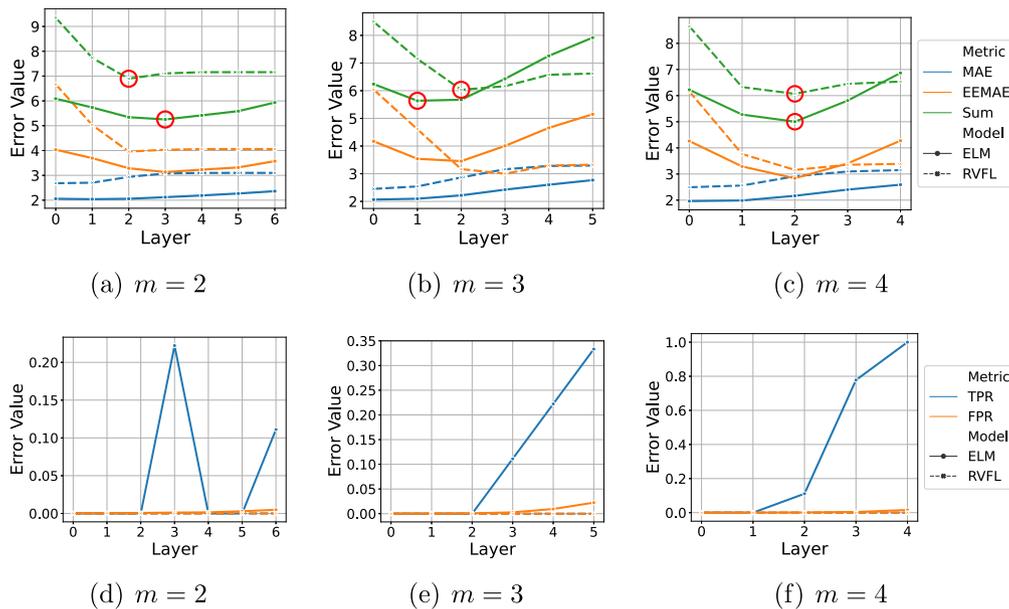


Fig. 18. Error metrics evolution on validation data when increasing the number of layers for $m = 2$, $m = 3$ and $m = 4$ in database B.

original data and after applying SMOGN for balancing the training data; and state-of-the-art deep learning methods were tested. A brief description of this methods is presented here.

4.2.1. Shallow machine learning methods and SMOGN

Eight classical ML methods for regression problems have been considered: Linear Regression (LR) (Draper and Smith, 1998), Regression Trees (RT) (Loh, 2011), Random Forest (RF) (Breiman, 2001), Support Vector Regression (SVR) (Awad and Khanna, 2015), Least Square Support Vector Regression (LSSVR) (Wang and Hu, 2005), Multilayer Perceptron (MLP) (Gardner and Dorling, 1998), ELM and RVFL. In order to address the issue of dealing with a highly unbalanced datasets, a data balancing technique applied to regression, SMOGN, has been implemented before fitting the ML methods. In principle, the SMOGN (Branco et al., 2017) approach aims to deal with imbalanced regression problems, where the most important cases to the user are

poorly represented in the available data. It combines random under-sampling with two oversampling techniques: SmoteR (Torgo et al., 2013) and introduction of Gaussian Noise. SMOGN generates new synthetic examples with SmoteR only when the seed example and the k-NN selected are “close enough” and uses the introduction of Gaussian Noise when the two examples are “more distant”.

4.2.2. Deep learning regression methods

In recent decades, deep learning approaches have become increasingly popular for performing tasks related to time series classification or regression (Ismail Fawaz et al., 2019; Gamboa, 2017). A general deep learning framework for time series forecasting is depicted in Fig. 13. These networks are designed to learn hierarchical representations of the data. A deep neural network is a composition of N parametric functions referred to as layers where each layer is considered a representation of the input domain.

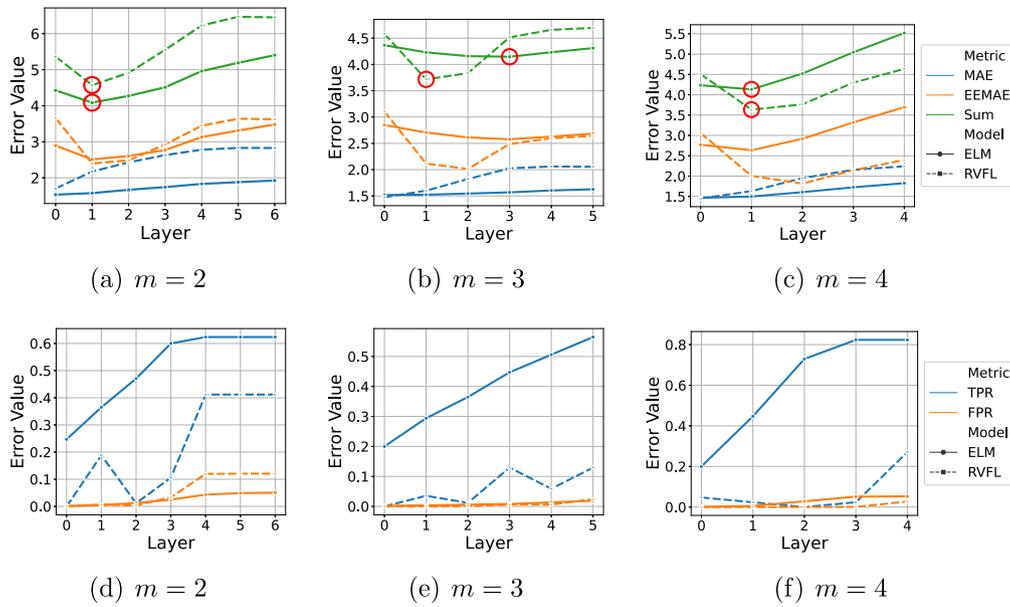


Fig. 19. Error metrics evolution on validation data when increasing the number of layers for $m = 2$, $m = 3$ and $m = 4$ in database C.

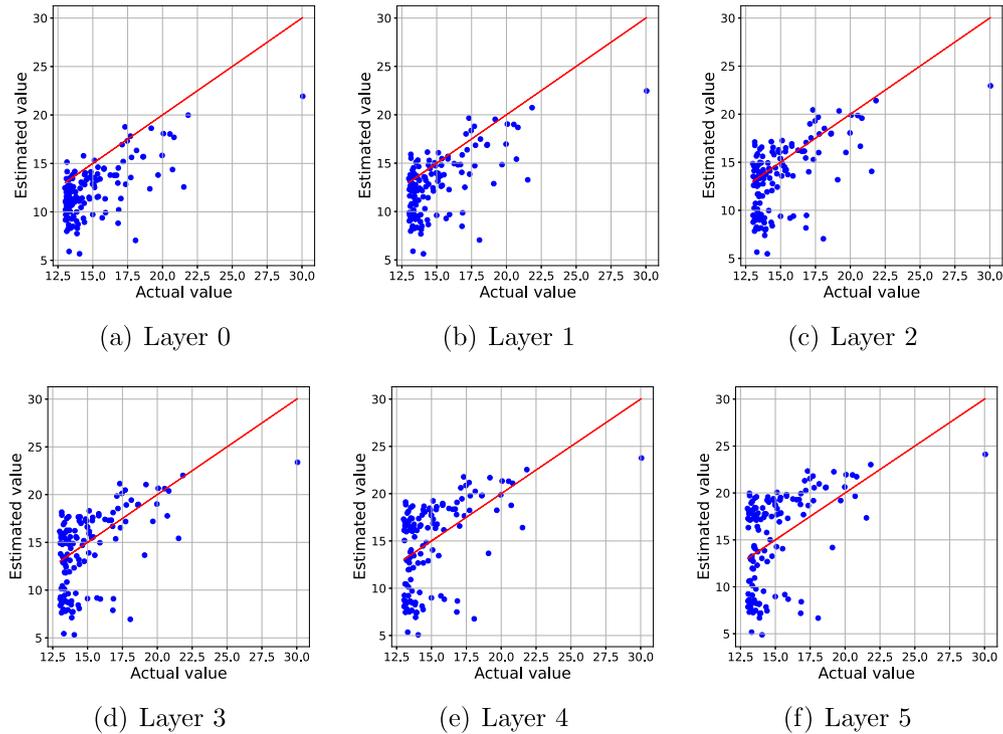


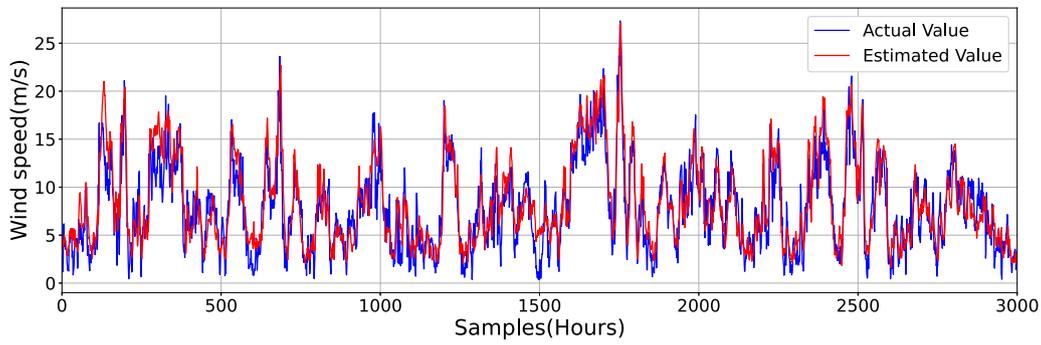
Fig. 20. Extreme events forecasting vs actual extreme values (MIS) on validation data using fuzzy-based ELM cascade ensemble with $m = 2$ for database A.

Among the existing deep learning methodologies for time series forecasting, the following were selected as benchmarks: including some mature architectures as Recurrent Neural Networks (RNNs) (Hüsken and Stagge, 2003), Gated Recurrent Units (GRUs) (Chung et al., 2014), Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or 1-D Convolutional Neural Networks (1D-CNN); a combination of two of these methods (1D-CNN + LSTM) (Zhao et al., 2019); and state-of-the-art architectures that recently emerged and have gained great visibility due to their strong performance as Residual Network (ResNet) (Wang et al., 2017) and InceptionTime (Ismail Fawaz et al., 2020).

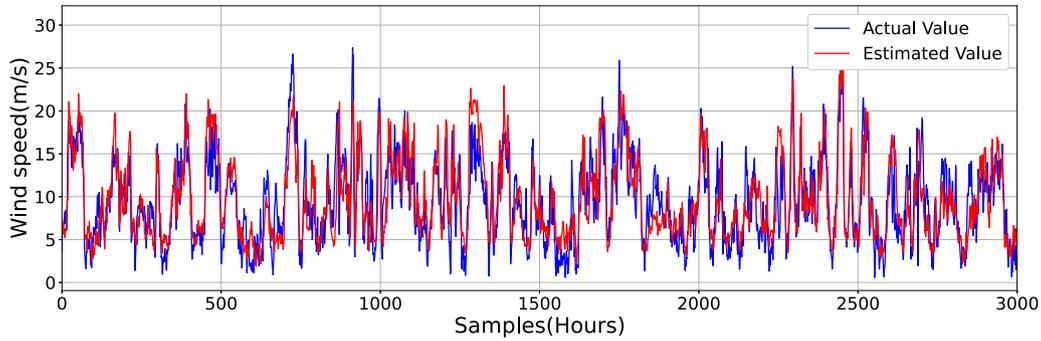
Preliminary work prior to the application of these methods involves the transformation of the data expressed as matrices into three-dimensional tensors (features \times time sequence of length $L \times$ samples). The parameter L was determined separately for each method and each database by computing the validation error when L ranged between 2 and 10 and selecting the L value that provided the minimum error.

4.3. Experimental setup

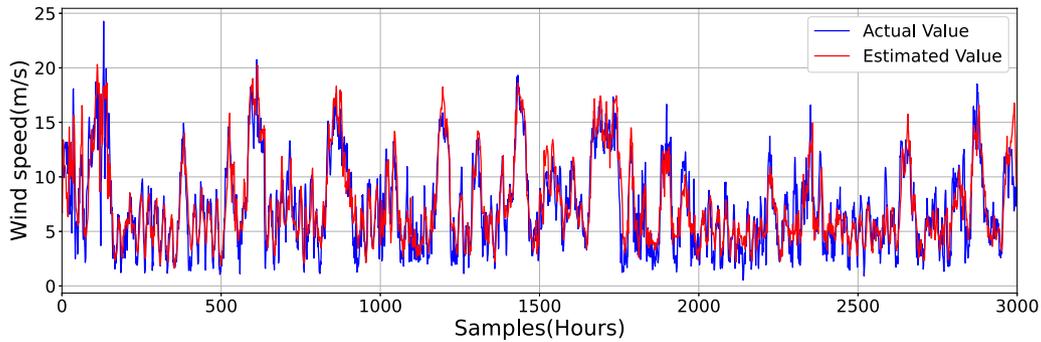
The experimental setup, along with the parameters set for the different ML methods, are detailed in this section. First, a preliminary



(a) Wind farm A (Fuzzy-based ELM cascade ensemble with $m = 3$ and $N = 3$)

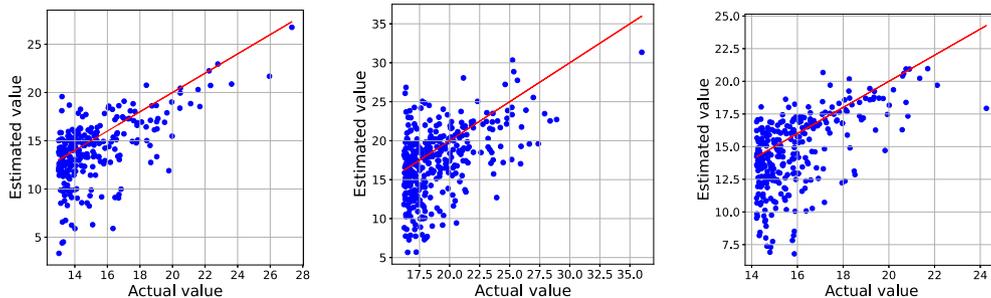


(b) Wind farm B (Fuzzy-based ELM cascade ensemble with $m = 2$ and $N = 3$)



(c) Wind farm C (Fuzzy-based ELM cascade ensemble with $m = 3$ and $N = 3$)

Fig. 21. Comparison of actual wind speed (blue) with forecasted wind speed (red) for the first 3000 test samples.

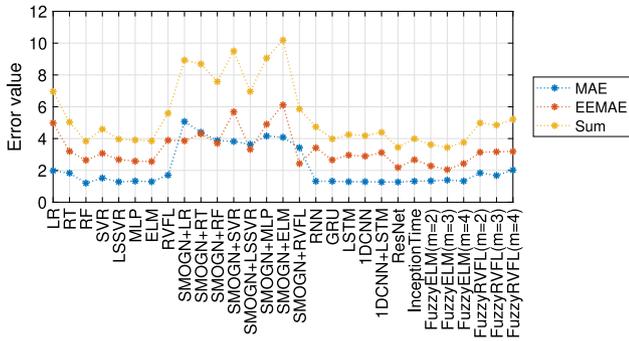


(a) Farm A (ELM: $m = 3$) (b) Farm B (ELM: $m = 2$) (c) Farm C (ELM: $m = 3$)

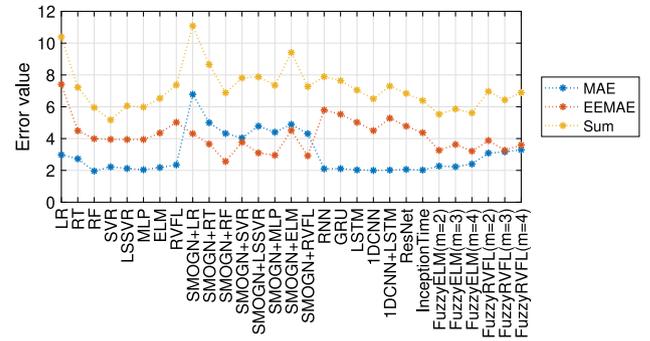
Fig. 22. Comparison of actual wind speed with forecasted wind speed on actual extremes (MIS) using fuzzy-based ELM cascade ensemble methodology.

dataset preparation is performed. The steps of this preprocessing are: (1) Splitting the dataset into training and test (70%–30%) subsets, assuring that no test instance was seen by the ML/DL methods during

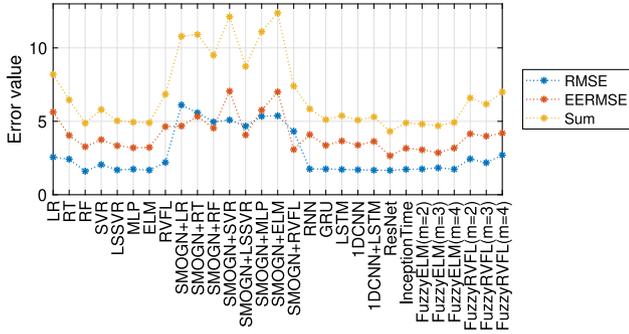
the training. Since dealing with timed-series data, instead of randomly splitting the datasets, last 20% of the data were separated as test data. Then, the training data are splitted again into training (first 75% data)



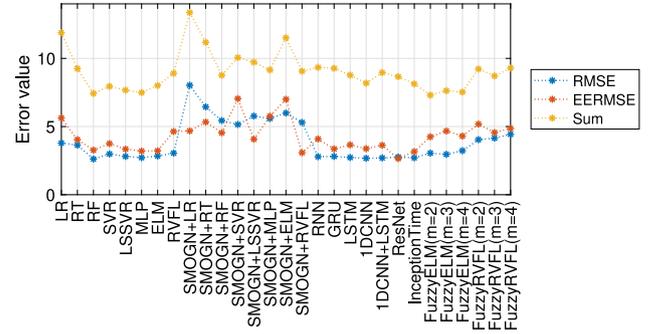
(a) MAE, EEMAE and Sum of both comparison.



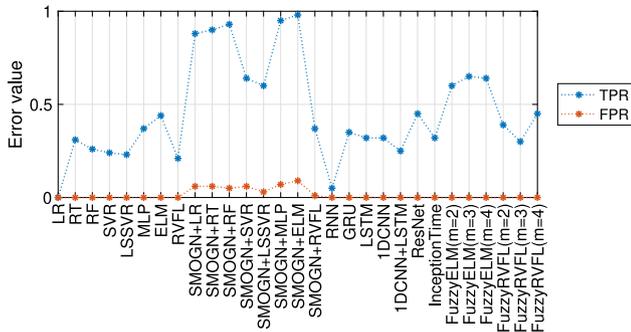
(a) MAE, EEMAE and Sum of both comparison.



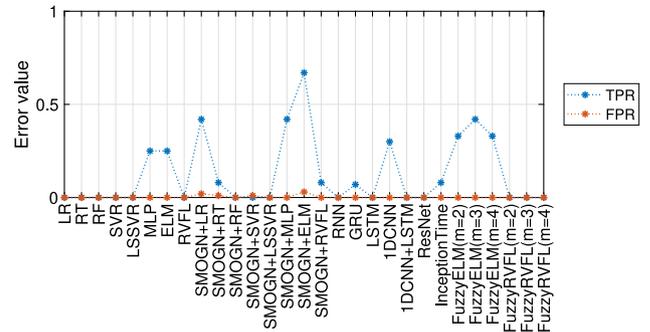
(b) RMSE, EERMSE and Sum of both comparison.



(b) RMSE, EERMSE and Sum of both comparison.



(c) TPR and FPR comparison.



(c) TPR and FPR comparison.

Fig. 23. Performance comparison for the different methodologies applied to database A.

Fig. 24. Performance comparison for the different methodologies applied to database B.

and validation (last 25% data). (2) Scaling of the features, which is important to ensure the upper and lower limits of data in the given predefined range. Feature standardization was performed, causing data to have zero-mean and a unit-variance (Eq. (18)), where x is the original feature vector, \bar{x} denotes the feature mean and σ its standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma} \quad (18)$$

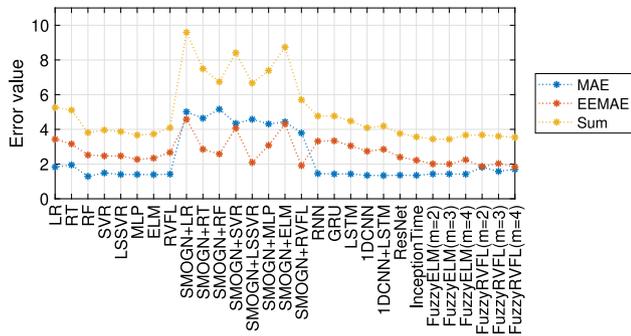
Table 2 shows the parameters used for the benchmark methods considered, as well as for the DE optimization algorithm applied. Furthermore, for the DL methods employed, a dense neural network was implemented after the GRU, LSTM, 1D-CNN and 1D-CNN + LSTM cases, containing two layers with 64 and 32 neurons in the first two cases and 4 layers with 256, 1258, 64 and 32 neurons in the last two cases.

All of the simulations were run on a Intel(R) Core(TM) i7-10700 CPU with 2.90 GHz and 16 GB RAM using the Python libraries, imblearn, sklearn, tensorflow, scipy and tsai.

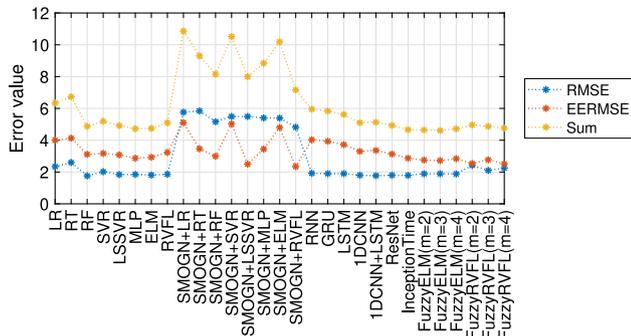
Regarding the computation efficiency of the proposed methodology, the training times of the fuzzy-based ensembles are in the range of the other ML/DL models, being up to 2 min in the case of ELM-ensembles and up to 2 h in the case of RVFL-ensembles. The largest computational cost is associated with the use of DE to optimize the membership functions, this process, which is performed offline prior to the deployment of the method in operation, takes about 30 min per generation for the ELM ensemble and 5 h for RVFL.

4.4. Results

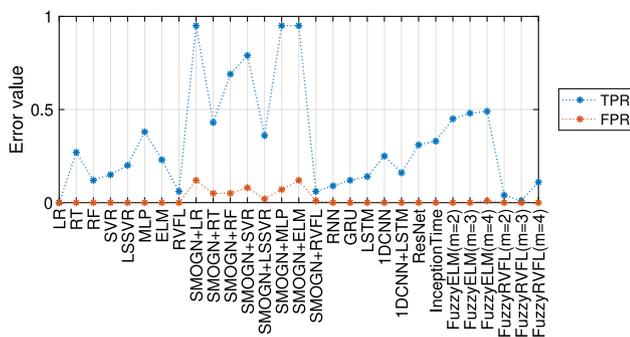
This section shows the experimental results obtained in this paper. First, Section 4.4.1 reports the results obtained when implementing the developed framework for the wind forecasting on the 3 wind farms. Then, Section 4.4.2 displays the results obtained using the methods established as benchmarks, along with a comparison and discussion of the results.



(a) MAE, EEMAE and Sum of both comparison.



(b) RMSE, EERMSE and Sum of both comparison.



(c) TPR and FPR comparison.

Fig. 25. Performance comparison for the different methodologies applied to database C.

4.4.1. Fuzzy-based ELM cascade ensemble results

The results achieved when applying the proposed fuzzy-based cascade ensemble methodology are presented below. First, the DE optimization algorithm detailed in Section 2.4 was applied for three different values of m in each database, using ELM and RVFL as regression model in each case. The fitness function of the DE algorithm was set as the minimization of the sum of MAE and EEMAE, seeking for a model that performs consistently well in both extremes and non-extremes values. Different pairs or combinations of metrics were tested in a preliminary phase, showing similar performance as long as information on both extreme events and overall forecast performance, so as to avoid over-prediction, was provided. Thus, for each individual in the DE algorithm, the fitness function computes the sum of MAE and EEMAE of each of the fuzzy-based ensemble layers and returns the minimum value, using always the validation data.

Optimized membership functions and learning rates returned by the optimization algorithm for $m = 2$, $m = 3$ and $m = 4$ are shown in Figs. 14–16 for databases A, B and C, respectively. It can be observed that different strategies are selected for each case, exhibiting differing

degrees of overlap. Also, optimized learning rates exhibit great differences when using ELM or RVFL as regressors, with low learning rates in the case of ELM, meaning that the system consider the prediction of the general model (layer 0) to be of high importance; and higher rates in the case of RVFL, which means that the predictions made by new layers are considered as more important. An analysis of the physical interpretability of the fuzzy-based partitioning of the training data is provided in Appendix B.

Subsequently, for the three cases under study ($m = 2$, $m = 3$ and $m = 4$), the optimal number of layers of the cascade ensemble (n) is determined according to the performance on the validation data. Figs. 17–19 shows the evolution of the validation order when increasing the number of layers for the different values of m and the two regression models assessed (ELM and RVFL). Since ELM and RVFL are stochastic regression models, simulations were run 10 times and the average values are shown in these figures. A clear trend is observed in all cases: metrics related to extremes (EEMAE and TPR) substantially improve in the initial layers, and only with a slight deterioration of non-extremes related metrics (MAE and FPR). Then, after the optimal number of layers is surpassed, results for all metrics, except for TPR, get significantly worse, meaning that predictions in all values is greatly overestimated. The reason behind this behavior is that when increasing the number of layers, the number of models involved in the prediction also becomes larger, while the size of the data training clusters gets narrower, meaning that either the models are trained with insufficient data or these data are too specific, causing the models to only behave optimally in a very tight range. Fig. 20 shows an example of the evolution of validation extreme events forecasting (extracted with by MIS method) when increasing the number of layers for database A and $m = 2$. It is observed that the prediction of this events substantially improve with a higher number of layer, until exceeding the optimum layer, where all predictions tend to be overestimated. Bearing this in mind, the selection of one number of layers or another will depend on the user's priority, whether it is to achieve a high detection rate of extreme events, predicting them as accurately as possible, or to accomplish this while keeping the number of false alarms as low as possible. In this study, the selection criterion consisted of selecting the models with the most balanced performance, so the sum of the MAE and EEMAE metrics was used as the parameter selection metric. Thus, according to Figs. 17–19, the number of layers chosen is indicated in these figures with red circles: 1, 2 and 1 for the 3 cases ($m = 2$, $m = 3$ and $m = 4$) with ELM and 1, 1 and 2 with RVFL for database A is; 3, 1 and 2 for ELM, and 2, 2 and 2 for RVFL in database B; and 1, 3 and 1 for ELM, and 1, 1 and 1 for RVFL in database C.

Next, results obtained on test data for the different cases under study are shown in Tables 3–5, where the fuzzy-based ensemble parameters defined previously are used. It can be observed how the ELM regression model outperforms the RVFL in the three datasets considered. This, however, is due to the fact that the initial prediction using ELM is superior in all cases. It is possible to appreciate how the proposed framework performs remarkably well with both regressors, leading to significant improvements in the extremes related metrics (EEMAE, EERMSE, TPR), with only slight deterioration in the MAE and RMSE metrics.

The best overall models for each data base can be selected as ($m = 3, N = 2$) for database A, ($m = 2, N = 3$) for database B and ($m = 3, N = 3$) for database C, using in all cases ELM as regression model, this cases present the lowest sum of MAE and EEMAE, meaning that the models are able to perform accurately in both extremes and non-extremes events. The number of ELM models ensemble in the final layer for each case is 9, 8 and 27, respectively. Fig. 21 shows the temporal wind speed prediction for the first 3000 test samples using the selected models in each database, and Fig. 22 illustrates it prediction performance on extreme events (MIS).

Table 3
Error metrics for proposed methodology applied on database A.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ELM	Baseline model	1.29	2.56	3.85	1.68	3.22	0.44	0.00
	2 MFs ($N = 1$)	1.34	2.28	3.62	1.75	3.06	0.60	0.00
	3 MFs ($N = 2$)	1.39	2.05	3.44	1.83	2.86	0.65	0.00
	4 MFs ($N = 1$)	1.33	2.43	3.76	1.74	3.18	0.64	0.00
RVFL	Baseline model	1.70	3.89	5.59	2.20	4.64	0.21	0.00
	2 MFs ($N = 2$)	1.84	3.14	4.98	2.44	4.15	0.39	0.00
	3 MFs ($N = 1$)	1.68	3.17	4.85	2.18	3.99	0.30	0.00
	4 MFs ($N = 2$)	2.02	3.19	5.21	2.70	4.19	0.45	0.00

Table 4
Error metrics for proposed methodology applied on database B.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ELM	Baseline model	2.18	4.35	6.53	2.84	5.17	0.25	0.00
	2 MFs ($N = 3$)	2.27	3.26	5.53	3.05	4.25	0.33	0.00
	3 MFs ($N = 1$)	2.23	3.63	5.86	2.96	4.67	0.42	0.00
	4 MFs ($N = 2$)	2.40	3.21	5.61	3.23	4.30	0.33	0.00
RVFL	Baseline model	2.35	5.02	7.37	3.05	5.85	0.00	0.00
	2 MFs ($N = 2$)	3.09	3.88	6.97	4.04	5.18	0.00	0.00
	3 MFs ($N = 2$)	3.17	3.26	6.43	4.15	4.56	0.00	0.00
	4 MFs ($N = 2$)	3.29	3.60	6.89	4.43	4.86	0.00	0.00

Table 5
Error metrics for proposed methodology applied on database C.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ELM	Baseline model	1.39	2.34	3.73	1.81	2.93	0.23	0.00
	2 MFs ($N = 1$)	1.43	2.01	3.44	1.89	2.75	0.45	0.00
	3 MFs ($N = 3$)	1.43	2.00	3.43	1.89	2.72	0.48	0.00
	4 MFs ($N = 1$)	1.42	2.25	3.67	1.88	2.84	0.49	0.01
RVFL	Baseline model	1.42	2.67	4.09	1.86	3.23	0.06	0.00
	2 MFs ($N = 1$)	1.82	1.86	3.68	2.42	2.54	0.04	0.00
	3 MFs ($N = 1$)	1.58	2.03	3.61	2.10	2.77	0.01	0.00
	4 MFs ($N = 1$)	1.70	1.84	3.54	2.25	2.51	0.11	0.00

4.4.2. Algorithms for comparison: results and discussion

The results for the benchmark methods for comparison, along with a discussion, are presented in this section for the three databases. Tables 6–8 show the error metrics for the ML methods with and without applying SMOGN and for the DL methods for wind farms A, B and C, respectively. It can be seen how for the three cases under study, a substantial improvement in the metrics related to the extremes (EEMAE, EERMSE and TPR) is obtained when SMOGN is applied. However, this improvement is achieved at the cost of a major degradation of the metrics related to non-extreme events (MAE, RMSE and FPR). DL methods offer more balanced results but do not yield satisfactory results in terms of EWS prediction, with TPR below 0.5 in all cases.

Then, a comparison of performance for all the methods proposed in this paper is shown in Figs. 23–25, for the 8 error metrics evaluated. Several conclusions may be extracted from this figures. First, it can be seen how the fuzzy ensemble models achieve the best performance in terms of EEMAE, EERMSE and sum of MAEs and RMSEs for the three databases, significantly outperforming the ML and DL methods, while these models only display a slightly better MAE and RMSE compared to the proposed approach. In the figures concerning TPR and FPR the trend becomes even clearer: the proposed models allow a major increase in the extreme detection rate without worsening the false positive rate, contrary to what happens when applying SMOGN. Rates higher than 40% for the 3 databases are achieved with the proposed framework.

This comparison illustrates the potential of the proposed methodology in the detection and prediction of extreme events: It is able to outperform other state-of-the-arts methods in terms of EEMAE, EERMSE and TPR, with only a small worsening in the prediction of non-extreme

events (MAE and RMSE). Therefore, the fuzzy-based cascade ensemble methodology proposed becomes a solution to be taken into account when the interest of the problem lies in the prediction of extremes events of a given meteorological variable.

5. Conclusions and future work

A novel fuzzy-based cascade ensemble of regression models is proposed to address a problem of predicting imbalance datasets for wind speed modeling, where the importance of an accurate prediction of Extreme Wind Speed (EWS) is high to support safety and operation of wind energy conversion systems. Specifically, using the newly developed approach, short-term EWS predictions in wind farms have been tackled at three medium-size wind energy farms located in Spain. Remarkable results have been achieved, outperforming the state-of-the-art methods used as benchmark models. The methodology proposed provides excellent results in the prediction of wind extremes, without deteriorating the prediction accuracy of the remaining events, keeping a very low false alarm ratio essential in the application of these models to the wind electricity power industries. This is due to the fact that the model takes into account several factors that influence the prediction of extreme wind events, such as the spatial-temporal evolution of the wind field, and the large-scale atmospheric circulation. The proposed methodology also incorporates a weighting system which allows the model to prioritize the prediction of certain events over others, which further improves its accuracy.

In addition, one of the advantages of the proposed approach lies in its simplicity, since it does not require any data balancing technique. Instead, it is based on the ensemble of different regression models, where

Table 6
Error metrics for benchmarks method on database A.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ML Methods	LR	1.98	4.98	6.96	2.56	5.63	0.00	0.00
	RT	1.83	3.20	5.03	2.42	4.04	0.31	0.00
	RF	1.20	2.64	3.84	1.60	3.27	0.26	0.00
	SVR	1.52	3.07	4.59	2.05	3.75	0.24	0.00
	LSSVR	1.28	2.68	3.96	1.69	3.34	0.23	0.00
	MLP	1.33	2.58	3.91	1.74	3.20	0.37	0.00
	ELM	1.29	2.56	3.85	1.68	3.22	0.44	0.00
	RVFL	1.70	3.89	5.59	2.20	4.64	0.21	0.00
SMOBN + ML Methods	LR	5.07	3.86	8.93	6.11	4.68	0.88	0.06
	RT	4.39	4.30	8.69	5.58	5.33	0.90	0.06
	RF	3.88	3.70	7.58	4.96	4.54	0.93	0.05
	SVR	3.82	5.68	9.50	5.09	7.05	0.64	0.06
	LSSVR	3.64	3.32	6.96	4.67	4.07	0.60	0.03
	MLP	4.16	4.90	9.06	5.34	5.76	0.95	0.07
	ELM	4.08	6.11	10.19	5.38	7.00	0.98	0.09
	RVFL	3.43	2.43	5.86	4.32	3.08	0.37	0.01
DL Methods	RNN	1.32	3.42	4.74	1.75	4.09	0.05	0.00
	GRU	1.32	2.66	3.98	1.75	3.36	0.35	0.00
	LSTM	1.29	2.96	4.25	1.72	3.66	0.32	0.00
	1D-CNN	1.29	2.89	4.18	1.70	3.38	0.32	0.00
	1D-CNN + LSTM	1.27	3.12	4.39	1.67	3.63	0.25	0.00
	ResNet	1.27	2.18	3.45	1.66	2.65	0.45	0.00
	InceptionTime	1.32	2.67	3.99	1.73	3.16	0.32	0.00

Table 7
Error metrics for benchmark methods on database B.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ML Methods	LR	2.98	7.41	10.39	3.79	8.09	0.00	0.00
	RT	2.73	4.49	7.22	3.64	5.61	0.00	0.00
	RF	1.96	3.99	5.95	2.62	4.81	0.00	0.00
	SVR	2.22	3.95	5.17	2.99	4.96	0.00	0.00
	LSSVR	2.12	3.94	6.06	2.81	4.86	0.00	0.00
	MLP	2.04	3.94	5.98	2.72	4.77	0.25	0.00
	ELM	2.18	4.35	6.53	2.84	5.17	0.25	0.00
	RVFL	2.35	5.02	7.37	3.05	5.85	0.00	0.00
SMOBN + ML Methods	LR	6.78	4.31	11.09	8.03	5.34	0.42	0.02
	RT	5.00	3.66	8.66	6.44	4.74	0.08	0.01
	RF	4.32	2.56	6.88	5.44	3.32	0.00	0.00
	SVR	4.04	3.77	7.81	5.15	4.91	0.00	0.01
	LSSVR	4.78	3.10	7.88	5.77	3.95	0.00	0.00
	MLP	4.40	2.95	7.35	5.59	3.56	0.42	0.00
	ELM	4.89	4.52	9.41	5.99	5.52	0.67	0.03
	RVFL	4.31	2.92	7.27	5.30	3.76	0.08	0.00
DL Methods	RNN	2.10	5.79	7.89	2.79	6.55	0.00	0.00
	GRU	2.11	5.53	7.64	2.81	6.46	0.07	0.00
	LSTM	2.03	5.02	7.05	2.73	6.04	0.00	0.00
	1D-CNN	2.00	4.50	6.50	2.67	5.51	0.30	0.00
	1D-CNN + LSTM	2.02	5.28	7.30	2.70	6.26	0.00	0.00
	ResNet	2.06	4.78	6.84	2.76	5.90	0.00	0.00
	InceptionTime	2.02	4.37	6.39	2.71	5.42	0.08	0.00

the key is to correctly divide the training data in subsets so that each model is focused on a specific part of the data spectrum. This optimal data partitioning is specific to the problem: as it has been shown, the shape, position and overlapping of the membership functions depend on the database, as well as on the degree of unbalancing of the problem. Moreover, it is important to divide the training data into a reduced number of subsets. Otherwise, if the number of ensemble models is too high, the models will be trained with an insufficient amount of data. Thus, their predictions will be too specific, causing the ensemble not to work optimally. The number of ensemble models that achieve better results is between 8 and 27.

Future research lines expanding this work include assessing the method efficiency to accurately predict on skewed databases, in applications different from wind speed forecasting, thus demonstrating the validity of the method with respect to other possible alternatives.

This research can be used to assess the efficiency of the method for data sets that are not normally distributed, as well as for forecasting applications other than wind speed. This will help validate the method's effectiveness compared to other potential alternatives.

The proposed methodology might be increased in the following ways: (1) testing different membership functions, with higher values of m and with distinct shapes (Gaussian, sigmoid, trapezoidal, etc.); (2) applying a multi-objective optimization in order to find the pareto front between the models that perform best in extremes, thus achieving the minimum Extreme Events Mean Absolute Error, and those that perform best in non-extreme events, which are measured by reaching the minimum Mean Absolute Error. Furthermore, different optimization techniques, such as genetic algorithms or particle swarm optimization, can be used to seek for optimal fuzzy logic parameters that enable the best possible performance. These techniques can help minimize the

Table 8
Error metrics for benchmark methods on database C.

		MAE	EEMAE	Sum	RMSE	EERMSE	TPR	FPR
ML Methods	LR	1.83	3.43	5.26	2.34	4.00	0.00	0.00
	RT	1.95	3.16	5.11	2.60	4.13	0.27	0.00
	RF	1.29	2.52	3.81	1.76	3.11	0.12	0.00
	SVR	1.49	2.47	3.96	2.02	3.17	0.15	0.00
	LSSVR	1.40	2.47	3.87	1.84	3.08	0.20	0.00
	MLP	1.40	2.27	3.67	1.85	2.87	0.38	0.00
	ELM	1.39	2.34	3.73	1.81	2.93	0.23	0.00
	RVFL	1.42	2.67	4.09	1.86	3.23	0.06	0.00
SMOBN + ML Methods	LR	5.01	4.58	9.59	5.76	5.10	0.95	0.12
	RT	4.64	2.85	7.49	5.84	3.46	0.43	0.05
	RF	4.16	2.58	6.74	5.16	2.99	0.69	0.05
	SVR	4.34	4.07	8.41	5.49	5.03	0.79	0.08
	LSSVR	4.58	2.09	6.67	5.49	2.50	0.36	0.02
	MLP	4.31	3.08	7.39	5.40	3.44	0.95	0.07
	ELM	4.43	4.31	8.74	5.39	4.80	0.95	0.12
	RVFL	3.79	1.91	5.70	4.82	2.34	0.06	0.01
DL Methods	RNN	1.45	3.32	4.77	1.92	4.03	0.09	0.00
	GRU	1.43	3.34	4.77	1.90	3.93	0.12	0.00
	LSTM	1.43	3.05	4.48	1.90	3.72	0.14	0.00
	1D-CNN	1.35	2.74	4.09	1.80	3.30	0.25	0.00
	1D-CNN + LSTM	1.34	2.85	4.19	1.77	3.36	0.16	0.00
	ResNet	1.36	2.40	3.76	1.80	3.13	0.31	0.00
	InceptionTime	1.35	2.22	3.57	1.79	2.87	0.33	0.00

error between the output of the fuzzy logic system and the desired output. By minimizing this error, the fuzzy logic system can be tuned to perform better and more accurately.

CRedit authorship contribution statement

C. Peláez-Rodríguez: Software, Methodology, Conceptualization, Investigation, Writing – original draft. **J. Pérez-Aracil:** Conceptualization, Software, Validation, Writing – review & editing. **L. Prieto-Godino:** Resources, Visualization. **S. Ghimire:** Writing – review & editing, Investigation, Methodology. **R.C. Deo:** Writing – review & editing, Methodology, Supervision. **S. Salcedo-Sanz:** Writing – review & editing, Supervision, Project administration, Funding acquisition, 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.

Acknowledgments

This research has been partially supported by the European Union, through H2020 Project “CLIMATE INTELLIGENCE Extreme events detection, attribution and adaptation design using machine learning (CLINT)”, Ref: 101003876-CLINT. This research has also been partially supported by the project PID2020-115454GB-C21 of the Spanish Ministry of Science and Innovation (MICINN).

Appendix A. Acronyms

Table A.9 provides the acronyms used in this research paper.

Table A.9

Acronyms.

Term	Acronyms
Artificial Intelligence	AI
Artificial Neural Network	ANN
Convolutional Neural Network	CNN
Deep Learning	DL
Differential Evolution	DE
Ensemble Empirical Mode Decomposition	EEMD
Extreme Events Mean Absolute Error	EEMAE
Empirical Mode Decomposition	EMD
Extreme Learning Machine	ELM
Extreme Wind Speed	EWS
False Positive Rate	FPR
Gated Recurrent Unit	GRU
k Nearest Neighbours	k-NN
Least Square Support Vector Regression	LSSVR
Linear Regression	LR
Long Short-Term Memory	LSTM
Machine Learning	ML
Mean Absolute Error	MAE
Numerical Weather Model	NWMM
Particle Swarm Optimization	PSO
Random Forest	RF
Random Vector Functional Links	RVFL
Recurrent Neural Networks	RNN
Stacked Extreme Learning Machine	SELM
Support Vector Regression	SVR
Synthetic Minority Over-Sampling with Gaussian Noise	SMOBN
True Positive Rate	TPR
Variational Mode Decomposition	VMD

Appendix B. Physical interpretation of fuzzy clustering

This appendix section discusses an analysis of the physical interpretability of the fuzzy-based partitioning of the training data. The division of the training data into different clusters is performed according to the membership functions provided by the optimization algorithm. These membership functions (Figs. 14–16) are only based on the value of the target variable, but it is possible to observe the differences between the predictor variables among the different clusters.

For this purpose, Figs. B.26, B.27 and B.28 show the normalized averages of the different predictive variables for each of the ELM

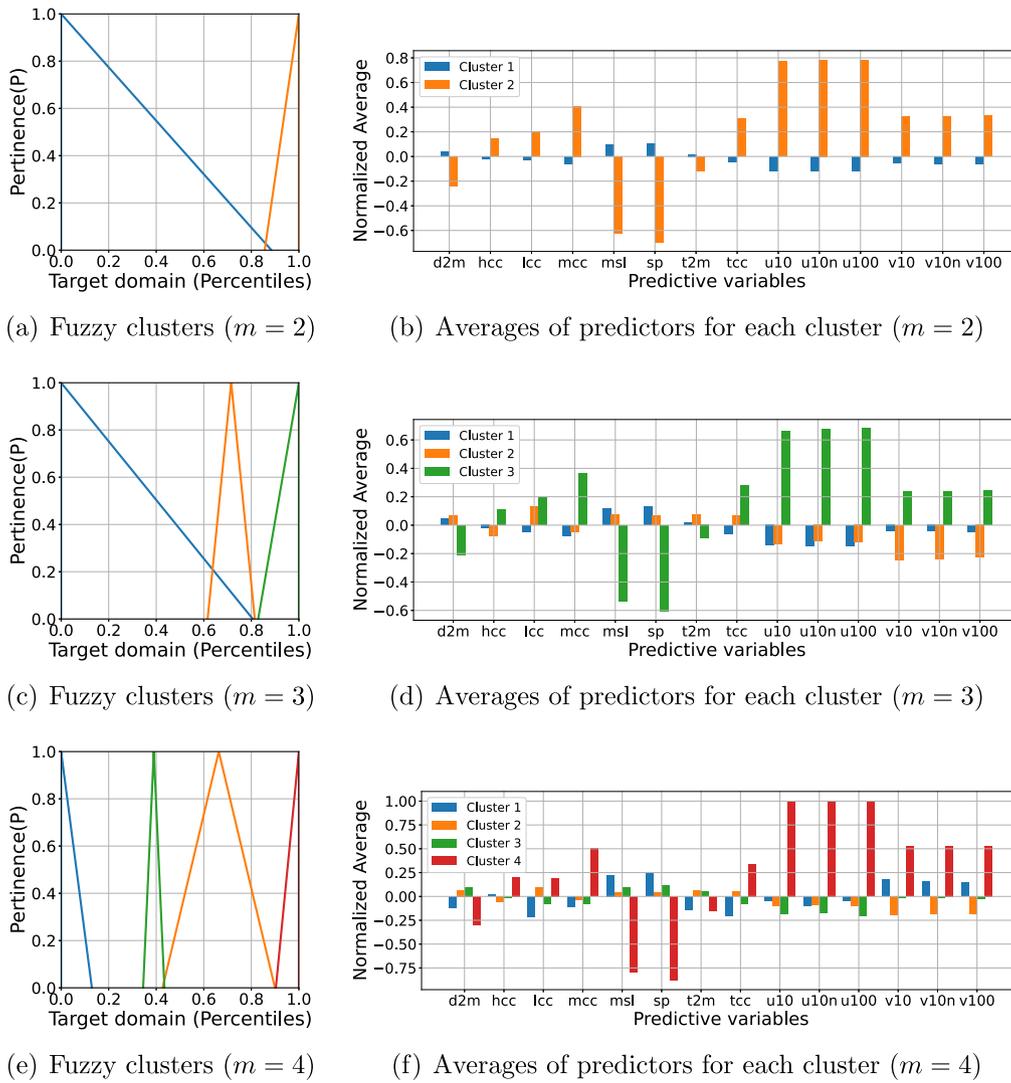
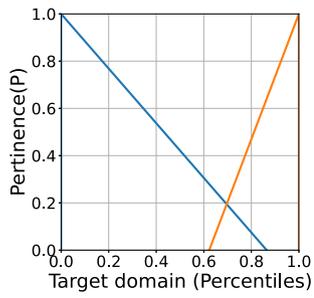
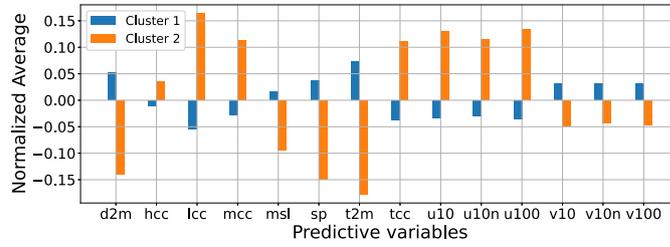


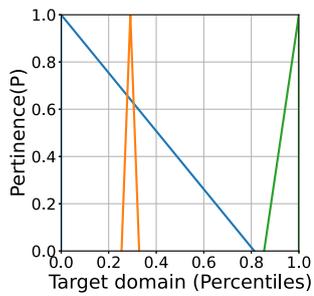
Fig. B.26. Wind Farm A (ELM fuzzy clusters).



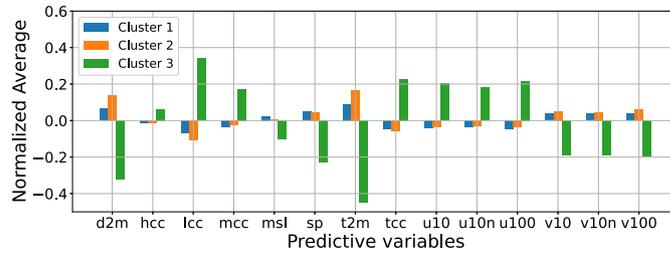
(a) Fuzzy clusters ($m = 2$)



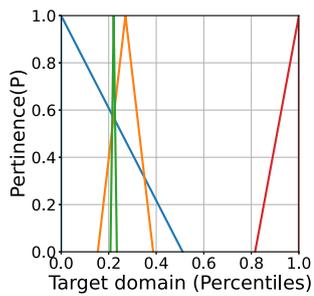
(b) Averages of predictors for each cluster ($m = 2$)



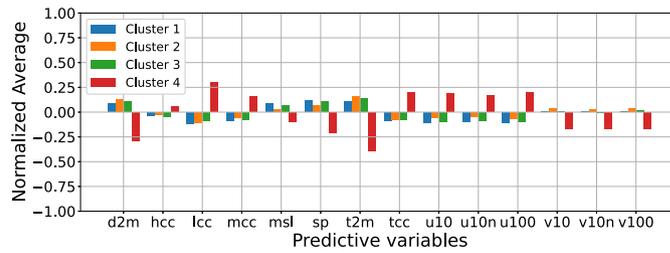
(c) Fuzzy clusters ($m = 3$)



(d) Averages of predictors for each cluster ($m = 3$)



(e) Fuzzy clusters ($m = 4$)



(f) Averages of predictors for each cluster ($m = 4$)

Fig. B.27. Wind Farm B (ELM fuzzy clusters).

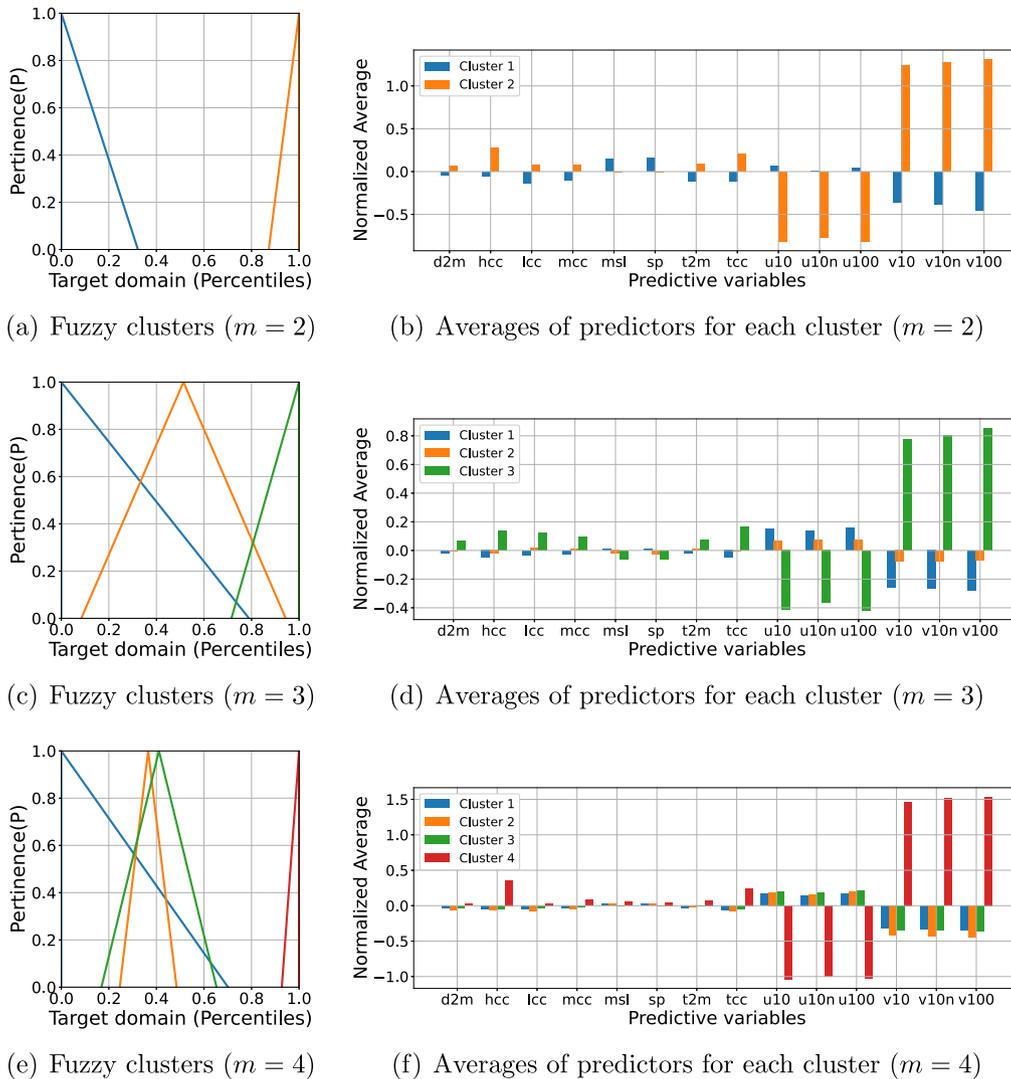


Fig. B.28. Wind Farm C (ELM fuzzy clusters).

training clusters, for the wind farms A, B and C, respectively. This predictors corresponds with the geographic nodes placed at the wind farms locations. Similar conclusions can be drawn from these figures, firstly, it is observed in all cases that the clusters corresponding to higher values of the target variable (i.e., those associated with wind speeds close to extreme values), exhibit much higher values of wind-related predictor variables (u10, u10n, u100, v10, v10n, v100). Furthermore, it is worth noting that the signs of these ‘extremes’ cluster are dependent on each database, providing physical information on the nature of the extreme events at the respective wind farm (wind farm A extremes exhibit positive u and v components, wind farm B extremes display positive u and negative v components, while wind farm C extremes exhibit negative u and positive v components).

Regarding the variables related to pressure (msl and sp), in the A and B database, they present a strong negative factor when the target variable exhibits high wind speed values. This behavior is also observed with the temperature predictors (d2 m and t2 m) for the first two databases, while no significant differences between clusters are observed for the wind farm C).

References

Abild, J., Andersen, E., Rosbjerg, d.D., 1992. The climate of extreme winds at the Great Belt, Denmark. *J. Wind Eng. Ind. Aerodyn.* 41 (1–3), 521–532.

An, Y., Pandey, M., 2005. A comparison of methods of extreme wind speed estimation. *J. Wind Eng. Ind. Aerodyn.* 93 (7), 535–545.
 Awad, M., Khanna, R., 2015. Support vector regression. In: *Efficient Learning Machines*. Springer, pp. 67–80.
 Bali, V., Kumar, A., Gangwar, S., 2019. Deep learning based wind speed forecasting-A review. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, pp. 426–431.
 Batista, G.E., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* 6 (1), 20–29.
 Branco, P., Ribeiro, R.P., Torgo, L., Krawczyk, B., Moniz, N., 2017. SMOGN: a pre-processing approach for imbalanced regression. *Proc. Mach. Learn. Res.* 74, 36–50.
 Breiman, L., 2001. Random forests. *Mach. Learn.* 45 (1), 5–32.
 Burlando, M., Pizzo, M., Repetto, M., Solari, G., De Gaetano, P., Tizzi, M., 2014. Short-term wind forecast for the safety management of complex areas during hazardous wind events. *J. Wind Eng. Ind. Aerodyn.* 135, 170–181.
 Cadenas, E., Rivera, W., 2009. Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks. *Renew. Energy* 34 (1), 274–278.
 Cao, W., Gao, J., Ming, Z., Cai, S., Shan, Z., 2017. Fuzziness based random vector functional-link network for semi-supervised learning. In: *2017 International Conference on Computational Science and Computational Intelligence*. CSCI, IEEE, pp. 782–786.
 Cao, W., Yang, P., Ming, Z., Cai, S., Zhang, J., 2020. An improved fuzziness based random vector functional link network for liver disease detection. In: *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, pp. 42–48.
 Cassola, F., Burlando, M., 2012. Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Appl. Energy* 99, 154–166.

- Chang, W.Y., 2013. Short-term wind power forecasting using the enhanced particle swarm optimization based hybrid method. *Energies* 6, 4879–4896.
- Chen, P., Vivian, A., Ye, C., 2021. Forecasting carbon futures price: a hybrid method incorporating fuzzy entropy and extreme learning machine. *Ann. Oper. Res.* 1–43.
- Chen, J., Zeng, G.-Q., Zhou, W., Du, W., Lu, K.-D., 2018. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. *Energy Convers. Manage.* 165, 681–695.
- Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Cook, N., 1982. Towards better estimation of extreme winds. *J. Wind Eng. Ind. Aerodyn.* 9 (3), 295–323.
- Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., Feitosa, E., 2008. A review on the young history of the wind power short-term prediction. *Renew. Sustain. Energy Rev.* 12 (6), 1725–1744.
- Davenport, A., 1967. The dependence of wind loads on meteorological parameters. In: *Proceedings of the International Research Seminar on Wind Effects on Buildings and Structures, Vol. 1*. University of Toronto Press, pp. 19–82.
- Davison, A.C., 1984. Modelling excesses over high thresholds, with an application. *Stat. Extrem. Appl.* 461–482.
- Del Ser, J., Casillas-Perez, D., Cornejo-Bueno, L., Prieto-Godino, L., Sanz-Justo, J., Casanova-Mateo, C., Salcedo-Sanz, S., 2022. Randomization-based machine learning in renewable energy prediction problems: critical literature review, new results and perspectives. *Appl. Soft Comput.* 108526.
- Draper, N.R., Smith, H., 1998. *Applied Regression Analysis, Vol. 326*. John Wiley & Sons.
- Duan, J., Zuo, H., Bai, Y., Duan, J., Chang, M., Chen, B., 2021. Short-term wind speed forecasting using recurrent neural networks with error correction. *Energy* 217, 119397.
- Farahbod, S., Niknam, T., Mohammadi, M., Aghaei, J., Shojaiyan, S., 2022. Probabilistic and deterministic wind speed prediction: Ensemble statistical deep regression network. *IEEE Access*.
- Freire, A.L., Rocha-Neto, A.R., Barreto, G.A., 2020. On robust randomized neural networks for regression: a comprehensive review and evaluation. *Neural Comput. Appl.* 32 (22), 16931–16950.
- Fukuoka, R., Suzuki, H., Kitajima, T., Kuwahara, A., Yasuno, T., 2018. Wind speed prediction model using LSTM and 1D-CNN. *J. Signal Process.* 22 (4), 207–210.
- Gamboa, J.C.B., 2017. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.
- Gao, Y., Wang, J., Zhang, X., Li, R., 2022. Ensemble wind speed prediction system based on envelope decomposition method and fuzzy inference evaluation of predictability. *Appl. Soft Comput.* 109010.
- Gardner, M.W., Dorling, S.R., 1998. Artificial neural networks (The multilayer perceptron)-A review of applications in the atmospheric sciences. *Atmos. Environ.* 32, 2627–2636.
- Harris, R., 1999. Improvements to the method of independent storms'. *J. Wind Eng. Ind. Aerodyn.* 80 (1–2), 1–30.
- Herbert, G.J., Iniyar, S., Sreevalsan, E., Rajapandian, S., 2007. A review of wind energy technologies. *Renew. Sustain. Energy Rev.* 11 (6), 1117–1145.
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., et al., 2018. Era5 hourly data on single levels from 1979 to present. *Copernicus Clim. Chang. Serv. (C3S) Clim. Data Store (CDS)* 10.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Höppner, F., Klawonn, F., 2003. Improved fuzzy partitions for fuzzy regression models. *Int. J. Approx. Reason.* 32 (2–3), 85–102.
- Huang, Z., Chalabi, Z., 1995. Use of time-series analysis to model and forecast wind speed. *J. Wind Eng. Ind. Aerodyn.* 56 (2–3), 311–322.
- Huang, G.-B., Wang, D.H., Lan, Y., 2011. Extreme learning machines: a survey. *Int. J. Mach. Learn. Cybern.* 2 (2), 107–122.
- Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. *Neurocomputing* 70, 489–501.
- Hüsken, M., Stagge, P., 2003. Recurrent neural networks for time series classification. *Neurocomputing* 50, 223–235.
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2019. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* 33 (4), 917–963.
- Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.-A., Petitjean, F., 2020. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.* 34 (6), 1936–1962.
- Jalli, R.K., Mishra, S.P., Naik, J., Sharma, P., Senapati, R., 2020. Prediction of wind speed with optimized EMD based RVFLN. In: 2020 IEEE-HYDICON. IEEE, pp. 1–5.
- Jiang, P., Liu, Z., Niu, X., Zhang, L., 2021. A combined forecasting system based on statistical method, artificial neural networks, and deep learning methods for short-term wind speed forecasting. *Energy* 217, 119361.
- Kani, S.P., Riahy, G., 2008. A new ANN-based methodology for very short-term wind speed prediction using Markov chain approach. In: 2008 IEEE Canada Electric Power Conference. IEEE, pp. 1–6.
- Kantz, H., Holstein, D., Ragwitz, M., Vitanov, N.K., 2004. Markov chain model for turbulent wind speed data. *Physica A* 342 (1–2), 315–321.
- Kaur, T., Kumar, S., Segal, R., 2016. Application of artificial neural network for short term wind speed forecasting. In: 2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy. PESTSE, IEEE, pp. 1–5.
- Khosravi, A., Koury, R., Machado, L., Pabon, J., 2018. Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. *Sustain. Energy Technol. Assess.* 25, 146–160.
- Kilmister, C., 1990. Handbook of applicable mathematics; supplement, edited by Walter Ledermann, Emlyn Lloyd, Steven Vajda and Carol Alexander. pp 479.f 52. 50. 1990. ISBN 0-471-91825-3 (John Wiley). *Math. Gaz.* 74 (470), 405.
- Kretzschmar, R., Eckert, P., Cattani, D., Eggimann, F., 2004. Neural network classifiers for local wind prediction. *J. Appl. Meteorol.* 43 (5), 727–738.
- Kusiak, A., Zheng, H., Song, Z., 2009. Short-term prediction of wind farm power: a data mining approach. *IEEE Trans. Energy Convers.* 24 (1), 125–136.
- Lei, M., Shiyang, L., Chuanwen, J., Hongling, L., Yan, Z., 2009. A review on the forecasting of wind speed and generated power. *Renew. Sustain. Energy Rev.* 13 (4), 915–920.
- Leon, M., Xiong, N., 2014. Investigation of mutation strategies in differential evolution for solving global optimization problems. In: *International Conference on Artificial Intelligence and Soft Computing*. Springer, pp. 372–383.
- Li, R., Jin, Y., 2018. A wind speed interval prediction system based on multi-objective optimization for machine learning method. *Appl. Energy* 228, 2207–2220.
- Liu, W., Wang, C., Li, Y., Liu, Y., Huang, K., 2021. Ensemble forecasting for product futures prices using variational mode decomposition and artificial neural networks. *Chaos Solitons Fractals* 146, 110822.
- Loh, W.-Y., 2011. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 1 (1), 14–23.
- López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inform. Sci.* 250, 113–141.
- Luo, X., Sun, J., Wang, L., Wang, W., Zhao, W., Wu, J., Wang, J.-H., Zhang, Z., 2018. Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy. *IEEE Trans. Ind. Inform.* 14 (11), 4963–4971.
- Malik, H., Yadav, A.K., 2021. A novel hybrid approach based on relief algorithm and fuzzy reinforcement learning approach for predicting wind speed. *Sustain. Energy Technol. Assess.* 43, 100920.
- Miranda, M.S., Dunn, R.W., 2006. One-hour-ahead wind speed prediction using a Bayesian methodology. In: 2006 IEEE Power Engineering Society General Meeting. IEEE, pp. 6–pp.
- Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, A.A., 2004. Support vector machines for wind speed prediction. *Renew. Energy* 29 (6), 939–947.
- Monfared, M., Rastegar, H., Kojabadi, H.M., 2009. A new strategy for wind speed forecasting using artificial intelligent methods. *Renew. Energy* 34 (3), 845–848.
- Nourani, V., Gökçekuş, H., Umar, I.K., 2020. Artificial intelligence based ensemble model for prediction of vehicular traffic noise. *Environ. Res.* 180, 108852.
- Outten, S., Sobolowski, S., 2021. Extreme wind projections over Europe from the Euro-CORDEX regional climate models. *Weather Clim. Extrem.* 33.
- Palutikof, J.P., Brabson, B., Lister, D.H., Adcock, S., 1999. A review of methods to calculate extreme wind speeds. *Meteorol. Appl.* 6 (2), 119–132.
- Pao, Y.-H., Park, G.-H., Sobajic, D.J., 1994. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6 (2), 163–180.
- Prado, F., Minutolo, M.C., Kristjanpoller, W., 2020. Forecasting based on an ensemble autoregressive moving average-adaptive neuro-fuzzy inference system–neural network-genetic algorithm framework. *Energy* 197, 117159.
- Price, K., Storn, R.M., Lampinen, J.A., 2006. *Differential Evolution: a Practical Approach to Global Optimization*. Springer Science & Business Media.
- Qiu, X., Suganthan, P.N., Amaratunga, G.A., 2018. Ensemble incremental learning random vector functional link network for short-term electric load forecasting. *Knowl.-Based Syst.* 145, 182–196.
- Ramon, J., Lledó, L., Torralba, V., Soret, A., Doblas-Reyes, F.J., 2019. What global reanalysis best represents near-surface winds? *Q. J. R. Meteorol. Soc.* 145 (724), 3236–3251.
- Ren, Y., Wen, Y., Liu, F., Zhang, Y., 2022. A short-term wind speed prediction method based on interval type 2 fuzzy model considering the selection of important input variables. *J. Wind Eng. Ind. Aerodyn.* 225, 104990.
- Ren, Y., Zhang, L., Suganthan, P.N., 2016. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Comput. Intell. Magazine* 11 (1), 41–53.
- Riahy, G., Abedi, M., 2008. Short term wind speed forecasting for wind turbine applications using linear prediction method. *Renew. Energy* 33 (1), 35–41.
- Saavedra-Moreno, B., Salcedo-Sanz, S., Carro-Calvo, L., Gascón-Moreno, J., Jiménez-Fernández, S., Prieto, L., 2013. Very fast training neural-computation techniques for real measure-correlate-predict wind operations in wind farms. *J. Wind Eng. Ind. Aerodyn.* 116, 49–60.
- Saha, M., Mitra, P., Chakraborty, A., 2015. Fuzzy clustering-based ensemble approach to predicting Indian monsoon. *Adv. Meteorol.* 2015.
- Salcedo-Sanz, S., Cornejo-Bueno, L., Prieto, L., Paredes, D., García-Herrera, R., 2018. Feature selection in machine learning prediction systems for renewable energy applications. *Renew. Sustain. Energy Rev.* 90, 728–741.

- Salcedo-Sanz, S., Ortiz-García, E.G., Pérez-Bellido, Á.M., Portilla-Figueras, A., Prieto, L., et al., 2011. Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Syst. Appl.* 38 (4), 4052–4057.
- Salcedo-Sanz, S., Pastor-Sánchez, A., Del Ser, J., Prieto, L., Geem, Z.-W., 2015. A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction. *Renew. Energy* 75, 93–101.
- Salcedo-Sanz, S., Pastor-Sánchez, A., Prieto, L., Blanco-Aguilera, A., García-Herrera, R., 2014. Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization–Extreme learning machine approach. *Energy Convers. Manage.* 87, 10–18.
- Salcedo-Sanz, S., Perez-Bellido, A.M., Ortiz-García, E.G., Portilla-Figueras, A., Prieto, L., Correo, F., 2009a. Accurate short-term wind speed prediction by exploiting diversity in input data using banks of artificial neural networks. *Neurocomputing* 72 (4–6), 1336–1341.
- Salcedo-Sanz, S., Pérez-Bellido, Á.M., Ortiz-García, E.G., Portilla-Figueras, A., Prieto, L., Paredes, D., 2009b. Hybridizing the fifth generation mesoscale model with artificial neural networks for short-term wind speed prediction. *Renew. Energy* 34 (6), 1451–1457.
- Sallis, P.J., Claster, W., Hernández, S., 2011. A machine-learning algorithm for wind gust prediction. *Comput. Geosci.* 37 (9), 1337–1344.
- Samal, R.K., 2021. Assessment of wind energy potential using reanalysis data: A comparison with mast measurements. *J. Clean. Prod.* 313, 127933.
- Schulz, B., Lerch, S., 2021. Machine learning methods for postprocessing ensemble forecasts of wind gusts: a systematic comparison. *arXiv preprint arXiv:2106.09512*.
- Sheridan, P., 2018. Current gust forecasting techniques, developments and challenges. *Adv. Sci. Res.* 15, 159–172.
- Shi, Q., Katuwal, R., Suganthan, P.N., Tanveer, M., 2021. Random vector functional link neural network based ensemble deep learning. *Pattern Recognit.* 117, 107978.
- Sideratos, G., Ikononopoulos, A., Hatzigiorgianni, N.D., 2020. A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks. *Electr. Power Syst. Res.* 178, 106025.
- Simiu, E., Heckert, N.A., 1996. Extreme wind distribution tails: a “peaks over threshold” approach. *J. Struct. Eng.* 122 (5), 539–547.
- Škrjanc, I., Iglesias, J.A., Sanchis, A., Leite, D., Lughofer, E., Gomide, F., 2019. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Inform. Sci.* 490, 344–368.
- Storn, R., Price, K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11 (4), 341–359.
- Tascikaraoglu, A., Uzunoglu, M., 2014. A review of combined approaches for prediction of short-term wind speed and power. *Renew. Sustain. Energy Rev.* 34, 243–254.
- Tian, Z., Ren, Y., Wang, G., 2019. Short-term wind speed prediction based on improved PSO algorithm optimized EM-ELM. *Energy Sources A Recovery Util. Environ. Eff.* 41 (1), 26–46.
- Torgo, L., Ribeiro, R.P., Pfahringer, B., Branco, P., 2013. Smote for regression. In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 378–389.
- Walshaw, D., 1994. Getting the most from your extreme wind data: a step by step guide. *J. Res. Natl. Inst. Stand. Technol.* 99, 399.
- Wang, H., Hu, D., 2005. Comparison of SVM and LS-SVM for regression. In: *2005 International Conference on Neural Networks and Brain*, Vol. 1. IEEE, pp. 279–283.
- Wang, J., Qin, S., Jin, S., Wu, J., 2015. Estimation methods review and analysis of offshore extreme wind speeds and wind energy resources. *Renew. Sustain. Energy Rev.* 42, 26–42.
- Wang, Z., Yan, W., Oates, T., 2017. Time series classification from scratch with deep neural networks: A strong baseline. In: *2017 International Joint Conference on Neural Networks. IJCNN, IEEE*, pp. 1578–1585.
- Wang, L., Zhang, Z., Long, H., Xu, J., Liu, R., 2016. Wind turbine gearbox failure identification with deep neural networks. *IEEE Trans. Ind. Inform.* 13 (3), 1360–1368.
- Wang, H., Zhang, Y.-M., Mao, J.-X., 2022. Sparse gaussian process regression for multi-step ahead forecasting of wind gusts combining numerical weather predictions and on-site measurements. *J. Wind Eng. Ind. Aerodyn.* 220, 104873.
- Wang, H., Zhang, Y.-M., Mao, J.-X., Wan, H.-P., 2020. A probabilistic approach for short-term prediction of wind gust speed using ensemble learning. *J. Wind Eng. Ind. Aerodyn.* 202, 104198.
- Wang, Y., Zou, R., Liu, F., Zhang, L., Liu, Q., 2021. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* 304, 117766.
- Xu, H., Wen, J., 2012. Differential evolution algorithm for the optimization of the vehicle routing problem in logistics. In: *2012 Eighth International Conference on Computational Intelligence and Security*. IEEE, pp. 48–51.
- Xu, D., Zhang, X., Hu, J., Chen, J., 2020. A novel ensemble credit scoring model based on extreme learning machine and generalized fuzzy soft sets. *Math. Probl. Eng.* 2020.
- Yang, T.-H., Tsai, C.-C., 2019. Using numerical weather model outputs to forecast wind gusts during typhoons. *J. Wind Eng. Ind. Aerodyn.* 188, 247–259.
- Yasunori, E., Yukihiko, H., Makito, Y., Sadaaki, M., 2009. On semi-supervised fuzzy c-means clustering. In: *2009 IEEE International Conference on Fuzzy Systems*. IEEE, pp. 1119–1124.
- Zhai, J., Zang, L., Zhou, Z., 2018a. Ensemble dropout extreme learning machine via fuzzy integral for data classification. *Neurocomputing* 275, 1043–1052.
- Zhai, J., Zhang, S., Zhang, M., Liu, X., 2018b. Fuzzy integral-based ELM ensemble for imbalanced big data classification. *Soft Comput.* 22 (11), 3519–3531.
- Zhang, K., Qu, Z., Dong, Y., Lu, H., Leng, W., Wang, J., Zhang, W., 2019. Research on a combined model based on linear and nonlinear features—A case study of wind speed forecasting. *Renew. Energy* 130, 814–830.
- Zhang, L., Suganthan, P.N., 2016. A comprehensive evaluation of random vector functional link networks. *Inform. Sci.* 367, 1094–1105.
- Zhang, W., Wang, J., Wang, J., Zhao, Z., Tian, M., 2013. Short-term wind speed forecasting based on a hybrid model. *Appl. Soft Comput.* 13 (7), 3225–3233.
- Zhang, P.-B., Yang, Z.-X., 2020. A new learning paradigm for random vector functional-link network: RVFL+. *Neural Netw.* 122, 94–105.
- Zhao, J., Mao, X., Chen, L., 2019. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* 47, 312–323.
- Zhou, J., Shi, J., Li, G., 2011. Fine tuning support vector machines for short-term wind speed forecasting. *Energy Convers. Manage.* 52 (4), 1990–1998.