Extreme Learning Machine Ensemble model for Time Series forecasting boosted by PSO: Application to an Electric Consumption problem

Mikel Larrea[a,*], Alain Porto[b], Eloy Irigoyen[a], Antonio Javier Barragán[c], José Manuel Andújar[c]

[a] *UPV/EHU (Barrio Sarriena s/n, 48940 Leioa, Spain)*
[b] *IDEKO (Arriaga Industrialdea 2, 20870 Elgoibar, Spain)*
[c] *UHU (Avda. de las Fuerzas Armadas s/n, 21007 Huelva, Spain)*

Abstract

Ensemble Model is a tool that has attracted attention due to its capability to improve the outcome performance of emerging techniques to solve the modelling and classifying problem. However, the feasibility of applying intelligent algorithms to build the Ensemble Model presents a challenge of its own. In this work, an Extreme Learning Machine ensemble is applied to the Time Series modelling problem. We develop a thorough study of the models that will be candidates to compose the ensemble, obtaining statistical results of optimal topologies to solve each Time Series problem. The proposed method for the ensemble is the weighted averaging method, where the parameters (weights) are tuned with the Particle Swarm Optimization algorithm. Lastly, the ensemble is tested first in the well known Santa Fe Time Series Competition benchmark. Given the obtained satisfactory results, the ensemble is secondly tested in a real problem of Spain's electric consumption forecasting. It is demonstrated that the PSO is a suitable algorithm to optimize Extreme Learning Machine ensemble and that the proposed strategy can obtain good results in both Time Series problems.

Keywords: Ensemble, ELM, PSO, Time-Series, Electric Consumption Forecasting

---

1. Introduction

Time Series consist of sequences of observations collected over time. The forecasting of future data sequences based on past data is a challenge not only in engineering [1], but in finance [2], physics [3] or biology [4] as well. In all these fields, an accurate model that represents the dynamics is fundamental to

*Corresponding author
Email addresses: m.larrea@ehu.eus (Mikel Larrea), aporto@ideko.es (Alain Porto)

successfully infer properties of the system. As an established procedure to initially validate the new proposals of modelling, the research community studies the new strategies testing them with known datasets as Santa Fe [5], UCI [6], or CATS [7], among others. The methods to obtain a prediction of Time Series vary in each field, but to mention some of them these include [8, 9, 3, 10]: Auto-Regressive Moving Average (ARMA), Auto-Regressive Integrated Moving Average (ARIMA), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), Fuzzy Logic (FL), Deep Learning (DL), Bayesian Neural Networks (BNN), Simple Exponential Smoothing (SES), Wavelet Transform (WT), Holt-Winters (HW) models, and Gaussian Process (GP), along with different combinations between them.

Among the methods listed, the ANN have the interesting features of being an universal approximation for nonlinear functions, their strong prediction performance and ability to tackle unknown system modelling [11]. These properties have made the ANN a well established technique for time series forecasting, even more so in problems that present nonlinearities [3, 8]. A specific type of ANN are the Extreme Learning Machines (ELMs). An ELM is a feed-forward ANN, in a single layer configuration, with a random initialization of the input layer parameters (weights and biases), that Huang et al. introduced in their work [12]. This way, the parameters of the output layer can be easily calculated minimizing the output error of the ANN. The ELMs have been used in time series forecasting in [13], where a one step ahead predictions are performed for non-stationary time series. In [14], an online sequential ELM is also employed for non-stationary time series.

These and many other techniques work reliably and accurately, but in many cases the path to get the desired results is full of discarded models [3, 15] that do not work properly or as expected. Besides, in Time Series modelling and forecasting, accuracy is one of the most important factors that should be optimized. Nevertheless, it is usually difficult to reach acceptable accuracy ratios with a single model [16, 17, 18, 19, 20]. Many recent studies show that the combination of multiple predictions of *weaker* models can perform more accurately than any individual model of the ensemble [21, 22]. The Ensemble Models (EMs) emerge as a method for reuse the already trained models that do not perform globally as expected, but that can contribute to increase the global accuracy of ensemble [23]. EMs have been used in different contexts and with many different types of models [24]. An EM is composed of a set of models also known as 'members' or 'experts' [25], which preferably represent the diversity of solutions to the time series forecasting problem. The EMs must consider each of the members' solution and how much they should weigh for the final solution. This final solution weighting is performed with methods such as Simple Averaging, Majority Voting, Ranking or Weighted Averaging. It has been shown that weighted methods, due to their capability of emphasizing an expert among the others, yields better results [26]. However, the computation of the weights is a difficult task on account of the complexity of the solution space. Evolutionary and swarm optimization algorithms have been proposed in order to solve these difficulties. For example, in [27] a Genetic Algorithm (GA) was used for the tuning of the ensemble weights to build a robust committee of neural networks, and in [28] the same weight tuning task is performed by a Particle Swarm Optimization (PSO) for peer-to-peer credit score classification.

The aim of this paper is to present and validate a suitable technique for computing the weights of a EM composed by ELM models for time series forecasting. This work is based on our previous results in [29], but going beyond by considering current tendencies where these techniques are combined to propose different solutions. For example, [30] where a new hybrid method combining ELM with other ML techniques is proposed to estimate electricity market prices, [31] where an ELM model optimized by the PSO algorithm is applied to directly estimate landslides, [32] solving the problem of the prediction of software defects developing an alternative hybrid fault prediction system based on PSO and Ensemble ELMs, [13] where an EM of ELMs is proposed, or [33] where an ensemble of ELMs is built with PSO.

Firstly, an introductory section will present the techniques and methods to be used. The section finalizes with the proposed ensemble that will be tested in the third section called "Experiments and Results". This section explore the methodology involved in order to select the different parameters of the EM in each presented experiment. The validation phase is carried out with two experiments; the well known Santa Fe laser benchmark and the Spanish electric consumption forecast problem. Finally, in the forth section, the discussion of the performance reached as well as the conclusions of using this methodology are presented.

2. Background

The foundation of this work is the hybridization of techniques. Each technique has its own mission: ELM will be used to create individual models of the Time Series. EM will gather different ELMs to fit into one unique model. Finally, PSO will be used to tune the weights of each model of the EM.

2.1. Extreme Learning Machines

The Extreme Learning Machine is a learning algorithm for Single Hidden Layer Feed-forward Neural Networks (SLFN) [12]. ELM randomly establishes the weights that connects the inputs and the hidden layer nodes. Once those weights are set, the output layer weights are analytically calculated. A generic ELM topology is represented in Fig.1. Being the input vector:

$$\mathbf{X}_{M\times 1} = [x_1, \ldots, x_M]^T. \quad (1)$$

The weight matrix connecting the input vector and the hidden layer nodes is represented by

$$\mathbf{W^1}_{N\times M} = \begin{bmatrix} w^1_{1,1} & \ldots & w^1_{1,M} \\ \vdots & \ddots & \vdots \\ w^1_{N,1} & \ldots & w^1_{N,M} \end{bmatrix}. \quad (2)$$

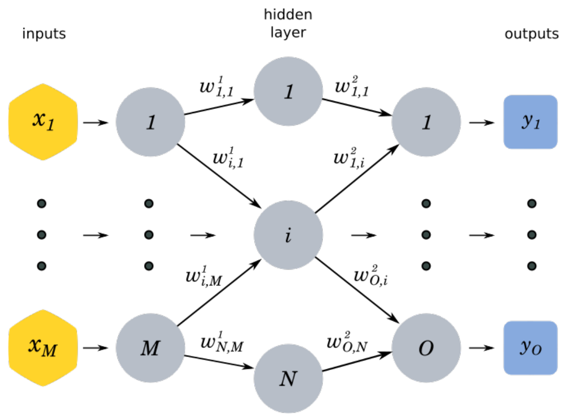


Figure 1: ELM general scheme

And the hidden layer bias vector as

$$\mathbf{b^1}_{N\times 1} = [b_1^1, \ldots, b_N^1]^T. \tag{3}$$

The activation function of the hidden layer is denoted by $\sigma()$ and can be almost any nonzero function. The weight matrix connecting the hidden layer nodes with the output layer nodes is represented by

$$\mathbf{W^2}_{O\times N} = \begin{bmatrix} w_{1,1}^2 & \ldots & w_{1,N}^2 \\ \vdots & \ddots & \vdots \\ w_{O,1}^2 & \ldots & w_{O,N}^2 \end{bmatrix}. \tag{4}$$

The relation of the ELM output ($\hat{\mathbf{Y}}_{O\times 1}$) and the input ($\mathbf{X}_{M\times 1}$) is obtained by

$$\hat{\mathbf{Y}}_{O\times 1} = \mathbf{W^2}_{O\times N} \cdot \sigma\left(\mathbf{W^1}_{N\times M} \cdot \mathbf{X}_{M\times 1} + \mathbf{b^1}_{N\times 1}\right). \tag{5}$$

Given D training samples ($\mathbf{X}_{M\times D}, \mathbf{Y}_{O\times D}$), with $M \times D$ dimensional input matrix, the ELM output can be described as

$$\hat{\mathbf{Y}}_{O\times D} = \mathbf{W^2}_{O\times N} \cdot \left[\sigma\left(\mathbf{W^1}_{N\times M} \cdot \mathbf{X}_{M\times D} + \mathbf{b^1}_{N\times D}\right)\right]. \tag{6}$$

Defining $\mathbf{H}_{N\times D}$ as

$$\mathbf{H}_{N\times D} = \sigma\left(\mathbf{W^1}_{N\times M} \cdot \mathbf{X}_{M\times D} + \mathbf{b^1}_{N\times D}\right). \tag{7}$$

Substituting (7) in (6), the output layer weights can be analytically calculated with (8), where the smallest norm $\mathbf{H}^\dagger_{D\times N}$ denotes the Moore-Penrose generalized inverse matrix of $\mathbf{H}_{D\times N}$:

$$\mathbf{W^2}_{O\times N} = \hat{\mathbf{Y}}_{O\times D} \cdot \mathbf{H}^\dagger_{D\times N}. \tag{8}$$

2.2. Ensemble Models

An Ensemble Model is a combination of different single models which aims the improvement of the final prediction. To create an EM, machine learning techniques could be used as they provide good individual forecasting performances. Among the most used are Support Vector Machines [34], Artificial Neural Networks [35] or Decision Trees [36].

The work carried out by various researches confirms the improvement of using EM. This improvement has been mathematically formalized in [25], where it is shown that the variance of the ensemble is lower than the average variance of all the individual models. They proved that the error of an ensemble is given by

$$E_{ens} = \frac{1}{k} E_{avg}, \tag{9}$$

where k is the number of models in the ensemble and the EM error is clearly lower than that of the individual models. This expression suggests that the average error of an ensemble model can be reduced by k introducing more models into the ensemble, but this is only true when the individual model's errors are uncorrelated [11]. Moreover, [25] states that in practice, with large number of models in the ensemble, (9) does not hold. Despite this drawback, the improvement of the EM over the individual model can be proven, leading to a research interest in creating better ensembles.

There are diverse techniques to create the combination of models in the ensemble. The technique to use depends on the problem to solve, leading to some techniques more suitable for classification as majority voting or weighted majority, and others for regression problems, such as weighted averaging or ranking [27].

2.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) [37] is one of the most spread swarm optimization algorithms. It was developed to mimic the movements of bird flocks or fish schools. PSO has been successfully applied to many problems: neural network training, function optimization, pattern classification and so on. [38] presents an up to date review of applications and research trends in PSO.

Particle position update formula is given by

$$\mathbf{p}_{l+1} = \mathbf{p}_l + \mathbf{v}_{l+1}, \tag{10}$$

where $\mathbf{p}_l$ is the particle position and $\mathbf{v}_l$ the particle velocity in the l-th iteration. The velocity can be calculated by

$$\mathbf{v}_{l+1} = a \cdot \mathbf{v}_l + c_1 \cdot r_1(\mathbf{p}_l - \mathbf{p}_b) + c_2 \cdot r_2(\mathbf{p}_l - \mathbf{p}_g), \tag{11}$$

being a the inertia weight, c_1 and c_2 the acceleration coefficients, r_1 and r_2 two coefficients randomly generated each iteration. $\mathbf{p}_b$ and $\mathbf{p}_g$ represent the personal best value for each particle and the best known swarm position respectively at iteration l.

The update process for all particle position is sequentially performed until the target criteria or the maximum number of iterations is reached.

2.4. Proposed Ensemble

The hybridization of these three techniques was proposed in [33] for a online sequential system, and in [29] for a time series problem. In this work, we apply a hybridization of the three techniques, performing previously a thorough analysis of the models introduced into the EM. The ensemble is built with PSO. The result is then tested in a benchmark and, as a practical example, a real problem to forecast the consumption of a Spanish electrical system is solved. To understand the proposed strategy, a clear picture of each techniques role will be drawn firstly. The ELM role in the ensemble is modelling of the data. Several ELMs are trained with the data available in order to have different members in the ensemble that would potentially provide diversity in the predictions. More details on the topology of the ELMs are given in the experiments section below.

The ensemble is constructed using the weighted averaging method. The weighted method adds complexity to the ensemble building since one of the most challenging task is to weigh the contribution of each member to the ensemble. Despite this increased complexity, the created ensembles have more adaptability and yield better results [26]. To overcome this increased complexity, some authors propose heuristic approaches as Genetic Algorithms to optimize the initial weights (the weights assigned before training) of an ensemble of neural networks [39]. In the present work, PSO has been used as it converges faster and presents more density of the search space. Another favorable feature of the PSO comparing to other methods is that it presents the ability to reach good solution without local search [40]. In [33], the selection of the models to build the ensemble is done with a PSO algorithm and combines the ensemble with an averaging method. The idea presented in [29], which we follow in this work, is to employ a ranking-based selection combined with PSO weighted averaging. This selection has been made due to the best performance shown by the PSO weighted averaging over the simple averaging.

The output of the ensemble is computed in weighted averaging by

$$\hat{\mathbf{Y}}_{ens}(\mathbf{X}_{M\times 1}) = \sum_{j=1}^{k} \alpha_j \hat{\mathbf{Y}}_j(\mathbf{X}_{M\times 1}), \tag{12}$$

where α_j is the weight assigned to the j-th model or expert in the ensemble, $\hat{\mathbf{Y}}_j()$ is the output of the j-th expert to the $\mathbf{X}_{M\times 1}$ input. It is assumed that the sum of all the weights sum to one [24]:

$$\sum_{j=1}^{k} \alpha_j = 1. \tag{13}$$

The weight assigned to each member of the ensemble is tuned with the PSO algorithm. In order to obtain these weights the PSO needs a loss function to evaluate its performance. The loss function will assess how good is a particle in the search space, and in the type of problem that this work handles this means how well the time series is predicted. The selected loss function is the

Root Mean Squared Error (RMSE) of the predicted samples. This loss function is widely used in the time series forecasting. The expression for obtaining the RMSE is

$$RMSE = \sqrt{\frac{1}{F}\sum_{h=1}^{F}(\hat{\mathbf{Y}}_h - \mathbf{Y}_h)^2}, \tag{14}$$

where F is the total amount of forecasted samples, $\hat{\mathbf{Y}}_h$ is the prediction of the model to a $\mathbf{X}_{M\times 1}$ input and $\mathbf{Y}_h$ is the hth real value.

The PSO, by definition, explores the whole search space without any constraint [41]. In this work, a constraint has been set in equation (13) leading to a reduction of the search space. In order to respect this constraint, the weights that the algorithm calculates are normalized ($A_{\mathbf{normalized}}$) before being evaluated by the loss function using

$$A_{\mathbf{normalized}} = \frac{A}{\sum_{j=1}^{k}\alpha_j}, \tag{15}$$

where A represents the vector of weights $A = [\alpha_1, \ldots, \alpha_k]$ and α_j stands for the individual weight of the jth model or expert in the ensemble. With this approach, the PSO can explore the whole search space with the subsequent minimal modification of the weights to fulfill the imposed constraint.

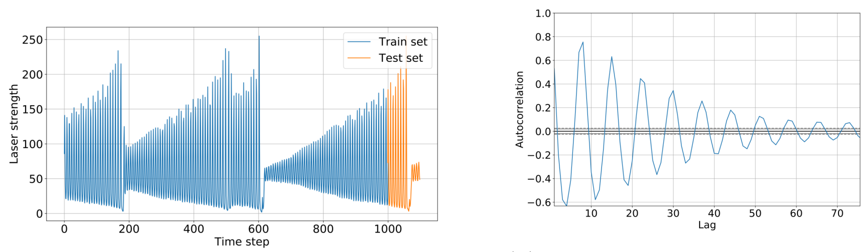
The final step to build the ensemble is the tuning of the parameters of the PSO algorithm described in equation (11). These three parameters are the inertia weight a and the two acceleration coefficients, c_1 and c_2. Due to the practical approach of the present work, the parameters have been set to values suggested by other authors. This way, according to [42] the weight inertia has been set to $a = 0.7298$ as the authors claim to be the best relation between the velocity of the particle and the swarm dependence velocity. The acceleration coefficients have been set $c_1 = c_2 = 1.49618$ as it was experimentally shown to be one of the best options [43].

3. Experiments and Results

The PSO boosted ELM ensemble is used to perform a benchmark and a real case application. In this section the most important aspects of the experiments will be detailed.

In the first part of the experiment, some dataset characteristics will be extracted to ease the design phase. In this step, an autocorrelation analysis will be performed to obtain the number of significant past samples on which the signal depends. Despite the information obtained with the autocorrelation, which informs on the range of inputs to use, a methodological procedure has been followed to determine the best ELM's topology.

The procedure consists of a batch training of ELMs within a range defined by the input vector ($\mathbf{X}_{M\times 1}$) and the range for the number of hidden nodes. A total of 10 ELMs are created for a specific combination, giving the performance results



(a) Santa Fe's Laser dataset

(b) Autocorrelation plot of Santa Fe time series

Figure 2: Santa Fe benchmark

a more reliable statistical base. The step size in the range is also 10, i.e, the number of inputs within a range between 10 and 100 will be $[10, 20, \ldots, 90, 100]$. The performance of each topology leads to a ranking that contains the best 100 ELMs and is used to build the ensemble. An aspect that remains in the topology along the experiments is the activation function of the hidden and output layer nodes, which is set to *Sigmoid* and *Linear*, respectively.

Once the ELMs are tuned, the ensemble building process is carried out following the next steps:

1. The ELMs are sorted by their prediction error in ascending order.
2. The first model in the sorted list is added to the ensemble with a weight of 1.
3. The following model in the list is added to the ensemble, and the PSO is called in order to compute the best weights.
4. Repeat step 3 until the maximum number of experts in the EM is reached.

The PSO in both experiments has 100 particles composed by all the weights assigned to each ELM in the ensemble. These weights are randomly initialized in the beginning of the search and the PSO performs 100 iterations to find the best possible ensemble weights.

In the following subsections, two experimental applications are presented.

3.1. Santa Fe Benchmark

Santa Fe time series benchmark was created in 1991 for the purpose of with other interesting time series benchmarks to facilitate open data to work with [5]. As time series forecasting is a multidisciplinary research field, advisors from the most relevant disciplines helped creating these datasets. The dataset selected in this work is the Santa Fe laser benchmark that consists of a 1000 points of the fluctuations in a far-infrared laser, obtained in a clean physics laboratory experiment. The Santa Fe laser dataset is presented in Fig.6a, where the first 1000 samples (blue line) is the data provided for training and the 1000 to 1100 sample range (orange line) is the data to forecast.

The objective of this benchmark is to perform a 100 sample prediction starting in sample 1000. The error calculation is determined by the normalized mean square error (NMSE) represented by

$$\mathrm{NMSE}(F)=\frac{\sum_{h\in\mathcal{T}}(\text{ observation }_h-\text{ prediction }_h)^2}{\sum_{h\in\mathcal{T}}(\text{ observation }_h-\text{mean}_{\mathcal{T}})^2}\approx\frac{1}{\widehat{\sigma}_{\mathcal{T}}^2}\frac{1}{F}\sum_{h\in\mathcal{T}}(x_h-\widehat{x}_h)^2\,, \tag{16}$$

where $h=1\dots F$ denotes number of samples to calculate the error, $\text{mean}_{\mathcal{T}}$ and $\widehat{\sigma}_{\mathcal{T}}^2$ denote average and variance of the target values in $\mathcal{T}$

For training the ELMs the first 1000 points have been used. Those points are pre-processed using the Gaussian normalization function. The inputs of the ELM are the past samples of the signal appending the signal value at time step h, and the output will be the forecasted value at $h+1$. The forecasted output serves as input to generate the following $h+2$ output, repeating this process until the $h+100$ prediction is completed.

To help determine the ELM topology and the number of inputs represented by past samples of the dataset, an autocorrelation plot is presented in Fig.2b. With this analysis, it can be concluded that the most significant past samples that the signal depends on are between 1 and 10 previous samples because of the change of the correlation sign and its seasonality.

In addition to the autocorrelation plot, a series of experiments are carried out in order to identify the best ELM topology. These experiments search for more hints on the selection of ELMs number of inputs as well as the ELMs number of hidden nodes. To gain statistical significance, each tested topology has 10 different ELMs, creating an experiment where plenty of the ELM combinations are covered in the selected range. The results of the batch experiments carried out to obtain the best ELM topology are represented in Fig.3, where the RMSE of the 10 ELMs of each topology is presented.

According to the results, the more inputs are involved in the ELM, the higher RMSE become and the more the hidden layer nodes reduce this error. With this characteristic in mind, we are able to select the best topologies not only to obtain the best model but to build a proper ensemble.

The best ELMs are classified and presented in Table.1, where the 'ELM Name' tag represents the ELMs configuration; with the number following the 'M' representing the number of ELM inputs and the number after 'N' standing for the number of nodes in the hidden layer (see Fig.1). Note that only the best 5 out of 300 ELMs are listed in the table.

Once the ranking is completed, the ensemble building takes place. The PSO algorithm computes the weights searching to minimize the loss function represented in (16). An interesting yet predictable result is shown in Fig.4, where the evolution of the RMSE can be seen as more ELMs are introduced in the ensemble. There is a clear trend to decrease the RMSE while introducing more ELMs in the ensemble. The PSO algorithm is most of the times capable of finding weights that produce a lower RMSE.

The last figure in Santa Fe benchmark, Fig.5, represents the comparative

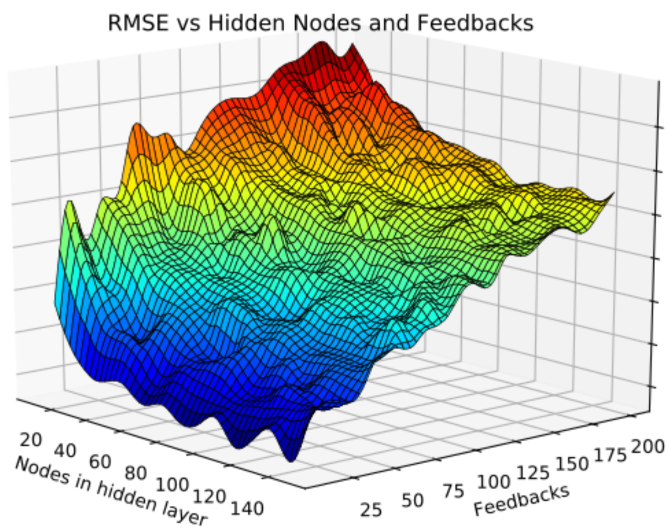


Figure 3: ELM topology analysis for Santa Fe benchmark

ELM Name	RMSE
M-10-N-160	29
M-10-N-170	29
M-10-N-150	29
M-10-N-110	30
M-10-N-100	31

Table 1: Best ELM prediction error

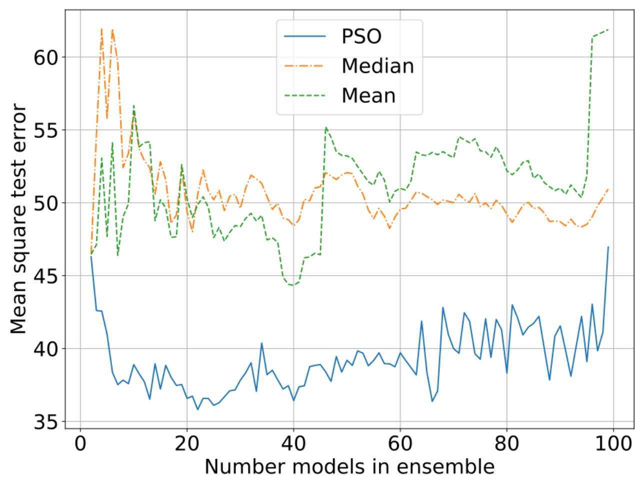


Figure 4: Root Mean Squared Error vs number of ELMs in the ensemble

of the Real or Expected output vs. the prediction of the Best ELM vs. the prediction made by the Ensemble.

The numerical results for PSO, mean and median are NMSE = 0.4575 , NMSE = 0.8877, NMSE = 1.3099 respectively and show that the PSO algorithm performs better than the other methods. The result provided by PSO can be considered as good comparing to the Santa Fe competition results (see *Table 2* in [5]), although the EM can not reproduce appropriately the collapse point around sample 70. This is one of the hardest issue in the benchmark as in the dataset this collapse phenomenon occurs twice. The prediction has degradation along the samples but it does not present a remarkable phase degradation. No emphasize has been done in the collapse data range and this lack of data preparation prior to the ELM training may have some influence in the results obtained.

3.2. Electric consumption model

The power consumption forecasting has become a must due to the increasing demand of electricity worldwide. One of the main objectives of a supply company of electric power is to have reliable short-term consumption forecasts, in order to optimize the management of that consumption through the resources available. A correct prediction improves the operations plan management of the supply companies, a task that is carried out not only based on available resources but also on the expected demand to supply and on the fulfillment of contractual commitments with supplying, distributing and marketing companies. There are many factors that may influence the electric consumption such as geographical location or regional temperature [44]. The temperature is considered one of the

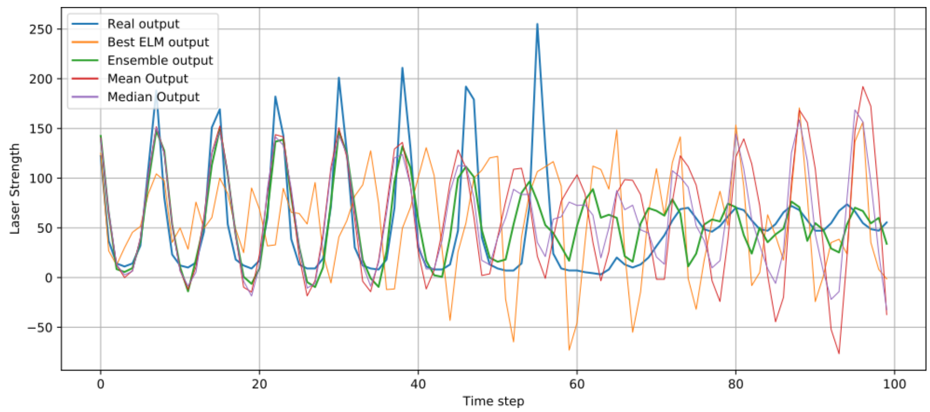


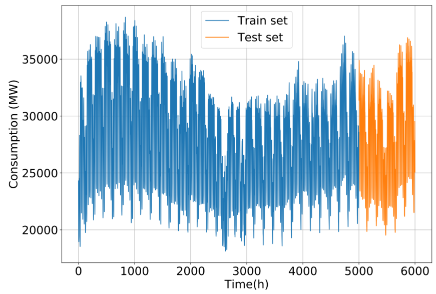
Figure 5: Santa Fe benchmark 100 sample prediction

most influential factor in the case that this experiment addresses, but not only, as other climate factors are also very influential. Despite the foregoing, as first step in the electric consumption experiment presented in this work, the main goal is to create a model that could be able to forecast the global electricity demand in Spain for a 24-hours temporal window with hourly predictions. To that end, only the previous consumption data will be used. This prediction is needed by the company in order to be able to auction, as is done quarterly, in a more efficient manner the price of the energy. The published predictions that are currently used in this context by Spanish Electrical Network company (Red Eléctrica Española), a corporation exclusively involved in electricity system operation, transportation and supplying in Spain, have these horizons of up to 24 hours.

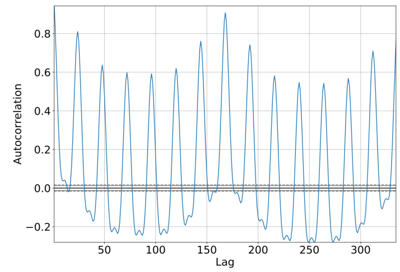
To analyse the accuracy and feasibility of the proposed strategy, the integration of an ELM ensemble boosted by PSO, a 12-month dataset provided by Spanish Electrical Network company has been used. The consumption distributions analyzed is shown in Fig.9. The dataset is composed of hourly power consumption data (MW) provided by Red Eléctrica Española in its website. Dataset consists of data from Jan 1st 2014 00:00h to Nov 5th 2016 11:00h, divided in a train set of 5.000 samples and a test set of $19,203$ samples. The dataset is pre-processed using the Gaussian normalization function.

As stated in the Santa Fe benchmark, an autocorrelation plot has been carried out to help determine the ELM topology. The results of the autocorrelation plot is presented in Fig.6b. In this case, the plot presents a 24 hourly repeat pattern of the data, with a change each 7 peaks. Hence, around 175 samples the plot presents another repeat pattern that in this case will be attributed to weekly pattern. With this analysis, it can be concluded that the most significant past samples that the signal depends on, are between 24 to 175 previous samples because of the change of the correlation sign and its seasonality.

In addition to the autocorrelation plot, an ELM topology sweep has been performed in a range from 10 to 300 inputs and from 10 to 300 hidden layer



(a) Electric consumption dataset



(b) Autocorrelation plot

Figure 6: Electric consumption data analysis

ELM Name	RMSE
M-40-N-140	3488
M-60-N-130	3830
M-90-N-120	3957
M-30-N-140	4033
M-60-N-140	4067

Table 2: Best ELM prediction error

nodes. The results of the batch experiments carried out to obtain the best ELM topology are represented in Fig.7, where the RMSE of the 10 ELMs of each topology is presented.

According to the results, less inputs involved in the the ELM lead to a higher RMSE and more hidden layer nodes increase this error. Analogously to the procedure followed in the Santa Fe benchmark, the top five best ELMs are classified and presented in Table.2.

Considering the best ELM Table.2, the EM building is carried out leading to the Fig.8, where the evolution of the RMSE can be seen as more ELMs are introduced in the ensemble. In this case, not all the added ELMs reduce the error, but still a clear RMSE reduction trend is shown. The figure plots the ensemble building for different methods as mean and median compared to PSO, showing how the PSO obtains a better result in comparison to cited methods.

Fig.9 represents the comparative of the electric consumption prediction in MW. The Real output (in blue) represents the electric consumption in a 24h window. The orange and green lines represent the prediction performed by the best ELM and by the PSO ensemble, respectively. The red and purple signals represent the prediction of the mean ensemble and the median ensemble.

The prediction performed by the PSO EM has a good performance considering the resulting RMSE, and clearly outperforms the best ELM,the mean EM and the median EM.

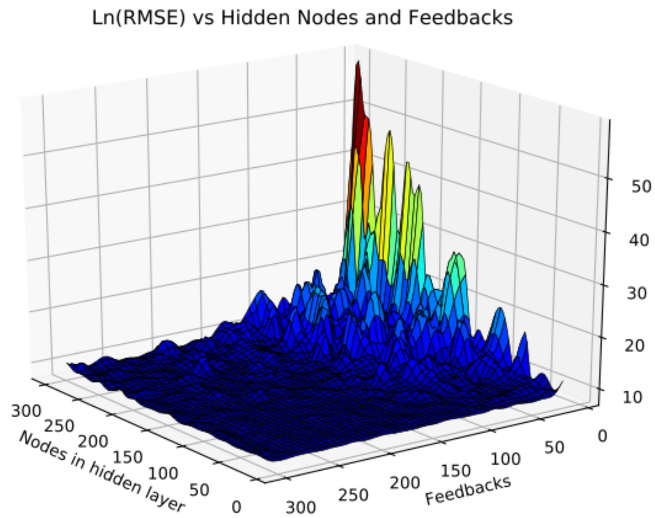


Figure 7: ELM topology analysis for electric consumption case

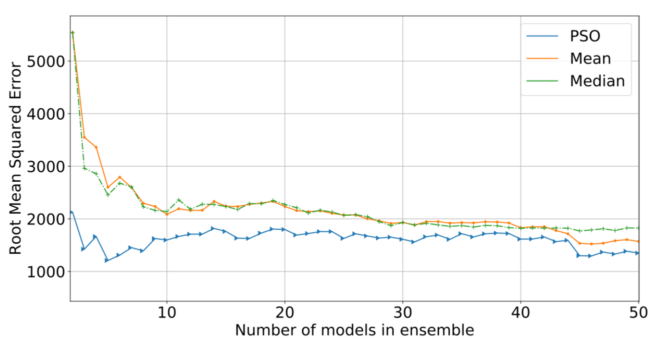


Figure 8: Root Mean Squared Error vs number of ELMs in the ensemble

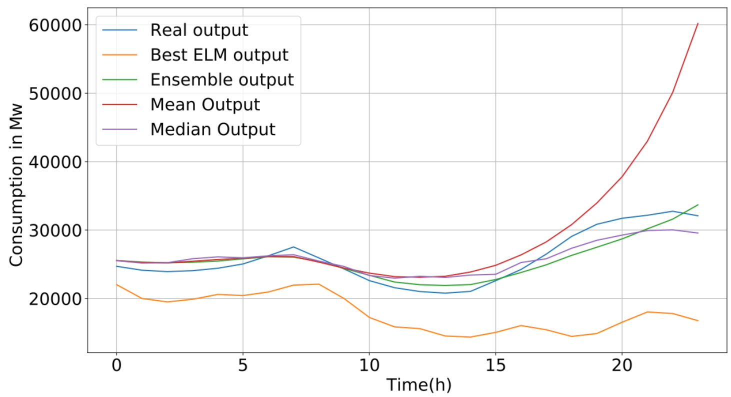


Figure 9: 24 hour electric consumption prediction

4. Conclusions

In this work, the application of the Particle Swarm Optimization algorithm for ELM based Ensemble Model tuning is proposed. The validation of the proposed strategy consists in the evaluation of a time series benchmark and a real application based in electrical power consumption prediction. The combination method selected to build the ensemble is the weighted averaging. This selection implies that the sum of the weights assigned to each model in the ensemble must sum one, and to respect this constrain, the weights calculated by the PSO are normalized. It has been demonstrated that the PSO is a suitable algorithm to optimize Extreme Learning Machine ensemble and that the strategy can obtain good results in both benchmarks analyzed in this work. Notwithstanding the selection of weighted averaging is discouraged over simple averaging by some researchers due to its tendency to overfit, the applications presented in this work has not shown such a problem. Moreover, PSO based weight tuning has shown better results than the mean and median weight calculation for the ensemble in both experiments.

Notwithstanding some inaccuracies in predictions in the Santa Fe benchmark, mainly attributable to the ELM training and its capacity to perform the task undertaken, the EM created with these ELMs has carried out a satisfactory prediction with good numerical results comparing to the Santa Fe competition results [5].

The electric power consumption prediction application has a promising performance, though it is on the first stages of the project. Obviously, the seasonality, the temperature, the climate and many other factors would arise in a more wide context prediction. Nevertheless, the main objective of the application of PSO in the ensemble building has also been fulfilled in this experiment.

One of the main aspects that this work benefits of is the methodological approach to the ELM topology selection. This approach has been used in other works, producing several ELMs that can be reused instead of discarding them all. Creating a ranking of the best ELMs helps the EM building although some problems, not found in the presented experiments, could arise. The problem of reduced diversity in the model's prediction and the problem of overfitting have to be analysed in future works. The usage of alternative ranking and ensemble building strategies are also interesting for the improvement of the proposal presented in this work.

5. Acknowledgements

This work comes under the framework of the project IT1284-19 granted by the Regional Government of the Basque Country. The authors would like to thank the company IK4-IDEKO that has supported this work.

References

[1] H. Wang, C. Hu, Y. Chen, B. Zhou, Z. Chen, Time-series based ensemble forecasting algorithm for out-limit detection on stable section of power network, in: Q. Liu, M. Mısır, X. Wang, W. Liu (Eds.), The 8th International Conference on Computer Engineering and Networks (CENet2018), Springer International Publishing, Cham, 2020, pp. 904–916.

[2] I. Bou-Hamad, I. Jamali, Forecasting financial time-series using data mining models: A simulation study, Research in International Business and Finance 51 (2020) 101072.

[3] S. Sobri, S. Koohi-Kamali, N. A. Rahim, Solar photovoltaic generation forecasting methods: A review, Energy Conversion and Management 156 (2018) 459 – 497.

[4] M. de Groot, N. Ogris, Short-term forecasting of bark beetle outbreaks on two economically important conifer tree species, Forest Ecology and Management 450 (2019) 117495.

[5] A. S. Weigend, N. A. Gershenfeld, Results of the time series prediction competition at the santa fe institute, in: Neural Networks, 1993., IEEE International Conference on, IEEE, 1993, pp. 1786–1793.

[6] C. B. D. Newman, C. Merz, UCI repository of machine learning databases (1998).
URL `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html`

[7] A. Lendasse, E. Oja, O. Simula, M. Verleysen, Time series prediction competition: The cats benchmark, Neurocomputing 70 (2007) 2325–2329.

[8] J. G. D. Gooijer, R. J. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (3) (2006) 443 – 473.

[9] G. C. R. G. M. L. George E. P. Box, Gwilym M. Jenkins, Time Series Analysis: Forecasting and Control, 5th Edition, Wiley, 2015.

[10] A. R. S. Parmezan, V. M. Souza, G. E. Batista, Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model, Information Sciences 484 (2019) 302 – 337.

[11] C. M. Bishop, Pattern recognition and machine learning, springer, 2006.

[12] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 IEEE International Joint Conference on Neural Networks, Vol. 2, 2004, pp. 985–990 vol.2.

[13] M. Van Heeswijk, Y. Miche, T. Lindh-Knuutila, P. A. Hilbers, T. Honkela, E. Oja, A. Lendasse, Adaptive ensemble models of extreme learning machines for time series prediction, in: International Conference on Artificial Neural Networks, Springer, 2009, pp. 305–314.

[14] X. Wang, M. Han, Online sequential extreme learning machine with kernels for nonstationary time series prediction, Neurocomputing 145 (2014) 90 – 97.

[15] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, Neurocomputing 10 (3) (1996) 215–236.

[16] S. Kurogi, M. Sawa, T. Ueno, Time series prediction of the cats benchmark using fourier bandpass filters and competitive associative nets, Neurocomputing 70 (13) (2007) 2354–2362.

[17] S. Leva, A. Dolara, F. Grimaccia, M. Mussetta, E. Ogliari, Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power, Mathematics and Computers in Simulation 131 (2017) 88 – 100.

[18] A. Safarishahrbijari, Workforce forecasting models: A systematic review, Journal of Forecasting 37 (7) (2018) 739–753.

[19] T. Rajaee, H. Ebrahimi, V. Nourani, A review of the artificial intelligence methods in groundwater level modeling, Journal of Hydrology 572 (2019) 336 – 351.

[20] F. Rubio, S. Navas, P. Ollero, J. Lemos, M. Ortega, Optimal control applied to distributed solar collector fields, Revista Iberoamericana de Automtica e Informtica Industrial 15 (3) (2018) 327–338.

[21] D. W. Opitz, R. Maclin, Popular ensemble methods: An empirical study, J. Artif. Intell. Res.(JAIR) 11 (1999) 169–198.

[22] V. Boln-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Information Fusion 52 (2019) 1 – 12.

[23] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001.

[24] J. D. Wichard, M. Ogorzalek, Time series prediction with ensemble models, in: 2004 IEEE International Joint Conference on Neural Networks, Vol. 2, 2004, pp. 1625–1630 vol.2.

[25] M. P. Perrone, L. N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, in: How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper, World Scientific, 1995, pp. 342–358.

[26] D. W. Opitz, J. W. Shavlik, Actively searching for an effective neural network ensemble, Connection Science 8 (3-4) (1996) 337–354.

[27] S. A. Jafari, S. Mashohor, Robust combining methods in committee neural networks, in: 2011 IEEE Symposium on Computers Informatics, 2011, pp. 18–22.

[28] C. Wang, R. Chiong, Y. Chen, Z. Hu, S. Dhakal, Y. Bao, A pso-based ensemble model for peer-to-peer credit scoring, 2019, pp. 412–418.

[29] A. Porto, E. Irigoyen, M. Larrea, A pso boosted ensemble of extreme learning machines for time series forecasting, in: International Joint Conference SOCO'18-CISIS'18-ICEUTE'18, Springer International Publishing, 2019, pp. 324–333.

[30] R. Tahmasebifar, M. K. Sheikh-El-Eslami, R. Kheirollahi, Point and interval forecasting of real-time and day-ahead electricity prices by a novel hybrid approach, IET Generation, Transmission Distribution 11 (9) (2017) 2173–2183.

[31] Y. Wang, H. Tang, T. Wen, J. Ma, A hybrid intelligent approach for constructing landslide displacement prediction intervals, Applied Soft Computing 81 (2019) 105506.

[32] Sumangalapatil, A. NagarajaRao, C. Shobabindu, Subspace clustering based class level software fault prediction using symbiotic particle swarm optimization and ensemble extreme learning machine, Journal of Computational and Theoretical Nanoscience 16 (5-6).

[33] Y. Liu, B. He, D. Dong, Y. Shen, T. Yan, R. Nian, A. Lendasse, Particle swarm optimization based selective ensemble of online sequential extreme learning machine (2015).

[34] L. Zhou, K. K. Lai, L. Yu, Least squares support vector machines ensemble models for credit scoring, Expert Systems with Applications 37 (1) (2010) 127 – 133.

[35] M. Amozegar, K. Khorasani, An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines, Neural Networks 76 (2016) 106 – 121.

[36] T. G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Springer Berlin Heidelberg, 2000, pp. 1–15.

[37] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948 vol.4.

[38] M. R. Bonyadi, Z. Michalewicz, Particle swarm optimization for single objective continuous space problems: A review, Evolutionary Computation 25 (1) (2017) 1–54.

[39] Z.-Q. Shen, F.-S. Kong, Optimizing weights by genetic algorithm for neural network ensemble, Advances in Neural Networks–ISNN 2004 (2004) 323–331.

[40] V. Kachitvichyanukul, Comparison of three evolutionary algorithms: Ga, pso, and de, Industrial engineering & management systems : an international journal 11 (3) (2012) 215–223.

[41] J. F. Kennedy, R. C. Eberhart, Y. Shi, R. C. Eberhart, Y. Shi, J. Kennedy, James. Kennedy, Swarm intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.

[42] F. van den Bergh, A. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (8) (2006) 937 – 971.

[43] R. C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), Vol. 1, 2000, pp. 84–88 vol.1.

[44] J. Diaz, J. Vuelvas, F. Ruiz, D. Patio, A set-membership approach to short-term electric load forecasting, Revista Iberoamericana de Automtica e Informtica industrial 16 (4) (2019) 467–479. `doi:10.4995/riai.2019.9819`.