Abstract—A novel hybrid approach, combining wavelet transform, particle swarm optimization, and adaptive-network-based fuzzy inference system, is proposed in this paper for short-term electricity prices forecasting in a competitive market. Results from a case study based on the electricity market of mainland Spain are presented. A thorough comparison is carried out, taking into account the results of previous publications. Finally, conclusions are duly drawn.

Index Terms—Electricity market, fuzzy logic, neural networks, price forecasting, swarm optimization, wavelet transform.

I. INTRODUCTION

During the last two decades, the electric power industry all over the world has undertaken significant restructuring. In most countries, a cost minimization paradigm has been replaced by a profit maximization one. In the profit maximization framework, producers, retailers, and consumers interact through a market seeking to maximize their respective profits. Two market structures are usually considered: a bilateral contract framework and a pool [1].

Price series exhibit greater complexity than demand series, given specific characteristics existing in price series. In most competitive electricity markets, price series present the following features: high frequency, nonconstant mean and variance, daily and weekly seasonality, calendar effect on weekend and public holidays, high volatility, and high percentage of unusual prices [2]. Therefore, price-forecasting tools are essential for all market participants for their survival under competitive environment [3]. In the short term, a producer needs price forecasts to optimally self-schedule and to derive its bidding strategy in the pool. Retailers and consumers need price forecasts for the same reasons as producers.

In the technical literature, several techniques to forecast short-term electricity prices have been reported, namely hard and soft computing techniques.

The hard computing techniques include autoregressive integrated moving average (ARIMA) [4], wavelet-ARIMA [5], and mixed model [6] approaches. Usually, an exact model of the system is required, and the solution is found using algorithms that consider the physical phenomena that govern the process. Although these approaches can be very accurate, they require a lot of information, and the computational cost is very high.

The soft computing techniques include neural networks (NN) [7], fuzzy neural networks (FNN) [8], weighted nearest neighbors (WNN) [9], adaptive wavelet neural network (AWNN) [10], hybrid intelligent system (HIS) [11], and cascaded neuro-evolutionary algorithm (CNEA) [12]. A combination of neural networks with wavelet transform (NNWT) has also been recently proposed [13], presenting a good trade-off between forecasting accuracy and computation time. Usually, an input-output mapping is learned from historical examples; thus, there is no need to model the system. Hence, these approaches can be much more efficient computationally and as accurate as the first ones, if the correct inputs are considered [14].

In this paper, a novel hybrid approach is proposed for short-term electricity prices forecasting. The proposed approach is based on the combination of wavelet transform (WT), particle swarm optimization (PSO), and adaptive-network-based fuzzy inference system (ANFIS). Our hybrid WPA approach is examined on the electricity market of mainland Spain, commonly used as the test case in several price forecasting papers [4]–[13]. It has been concluded that the Spanish market has a hard nonlinear behavior and time variant functional relationship [5], [8]. So, this market is a real-world case study with sufficient complexity. The proposed approach is compared with ARIMA, mixed-model, NN, wavelet-ARIMA, WNN, FNN, HIS, AWNN, NNWT, and CNEA approaches, to demonstrate its effectiveness regarding forecasting accuracy and computation time.

The contributions of this paper are threefold:

1) to propose a novel hybrid approach for short-term electricity prices forecasting;

2) to improve forecasting accuracy, taking into account the results of previous publications;

3) to reduce modeling complexity, achieving an acceptable computation time.

This paper is organized as follows. Section II presents the proposed approach to forecast electricity prices. Section III provides the different criteria used to evaluate the forecasting accuracy. Section IV provides the results from a case study based on the electricity market of mainland Spain. Section V outlines the conclusions.

II. PROPOSED APPROACH

The proposed approach is based on the combination of WT, PSO, and ANFIS. The WT is used to decompose the usually ill-behaved price series into a set of better-behaved constitutive
series. Then, the future values of these constitutive series are forecasted using ANFIS. The PSO is used to improve the performance of ANFIS, tuning the membership functions required to achieve a lower error. Finally, the ANFIS forecasts allow, through the inverse WT, reconstructing the future behavior of the price series and therefore to forecast prices.

A. Wavelet Transform

The WT is a transform that decomposes a signal into different frequency bands. The WT can be divided in two categories: continuous wavelet transform (CWT) and discrete wavelet transform (DWT). The CWT $W(a, b)$ of signal $f(x)$ with respect to a mother wavelet $\phi(x)$ is given by [15]

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \phi \left( \frac{x - b}{a} \right) dx$$  \hspace{1cm} (1)

where the scale parameter $a$ controls the spread of the wavelet and translation parameter $b$ determines its central position. DWT is more efficient and just as accurate as the CWT [16]. DWT is defined as

$$W(m, n) = 2^{-(m/2)} \sum_{t=0}^{T-1} f(t) \phi \left( \frac{t - n 2^m}{2^m} \right)$$  \hspace{1cm} (2)

where $T$ is the length of the signal $f(t)$. The scaling and translation parameters are functions of the integer variables $m$ and $n$ ($a = 2^m, b = n 2^m$); $t$ is the discrete time index.

A fast DWT algorithm based on the four filters was developed by Mallat [17]. Although the decimation process facilitates a faster computational process, this algorithm is not a time-invariant transform and may lose historical information in the process. The losses are especially undesirable for the forecast models, since complete and accurate historical data is essential in forecasting future data [18]. To keep the important historical information, as well as the time invariance for the transform, the redundant or nondecimated WT provided in MATLAB [19] can be applied, since the extra storage requirement is by no means excessive [20]. This algorithm is similar to that of Mallat, but omitting the decimation step.

Multiresolution via Mallat’s algorithm is a procedure to obtain “approximations” and “details” from a given signal. By successive decomposition of the approximations (Fig. 1), a multilevel decomposition process can be achieved where the original signal is broken down into lower resolution components.

A wavelet function of type Daubechies of order 4 (abbreviated as Db4) is used as the mother wavelet $\phi(t)$. This wavelet offers an appropriate trade-off between wavelength and smoothness, resulting in an appropriate behavior for short-term electricity prices forecasting. Similar wavelets have been considered by previous researchers [5], [15], [16]. Also, three decomposition levels are considered, as shown in Fig. 1, since it describes the price series in a more thorough and meaningful way than the others [21].

B. Particle Swarm Optimization

PSO is a heuristic approach first proposed by Kennedy and Eberhart in 1995 [22] as an evolutionary computational method. The PSO algorithm is based on the biological and sociological behavior of animals searching for their food [23].

Empirical evidence has been accumulated to show that the algorithm is a useful tool for optimization [24]. PSO has been applied to many optimization problems in engineering, for instance in [25].

Consider an optimization problem of $D$ variables. A swarm of $N$ particles is initialized in which each particle is assigned a random position in the $D$-dimensional hyperspace. Let $x$ denote a particle’s position and $v$ denote the particle’s flight velocity over a solution space.

The best previous position of a particle is $P_{best}$. The index of the best particle among all particles in the swarm is $G_{best}$. Velocity and position of a particle are updated by the following update rules:

$$v_i(t) = \omega v_i(t - 1) + \rho_1(x_{p_{best}} - x_i(t)) + \rho_2(x_{G_{best}} - x_i(t))$$  \hspace{1cm} (3)

$$x_i(t) = x_i(t - 1) + v_i(t)$$  \hspace{1cm} (4)

where $\omega$ is an inertia weight; $\rho_1$ and $\rho_2$ are random variables defined as $\rho_1 = r_1 C_1$ and $\rho_2 = r_2 C_2$, with $r_1, r_2 \sim U(0,1)$; and $C_1$ and $C_2$ are positive acceleration constants.

Fig. 2 illustrates the search mechanism of a PSO technique using the velocity update rule (3) and the position update rule (4).

In this work, constants $C_1$ and $C_2$ are both set at 2.0, following the typical practice in [26]. An inertia correction function called “inertia weight approach (IWA)” is also used in this work [26]. During the IWA, the inertia weight $\omega$ is modified according to the following equation:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{\text{Iter}_{\max}} \text{Iter}$$  \hspace{1cm} (5)

where $\omega_{max}$ and $\omega_{min}$ are the initial and final inertia weights, $\text{Iter}_{\max}$ is the maximum number of iteration, and $\text{Iter}$ is the current number of iteration.
ANFIS is a class of adaptive multilayer feedforward networks, applied to nonlinear forecasting where past samples are used to forecast the sample ahead. ANFIS incorporates the self-learning ability of NN with the linguistic expression function of fuzzy inference [27].

The ANFIS architecture is shown in Fig. 3. The ANFIS network is composed of five layers. Each layer contains several nodes described by the node function. Let $O_i^j$ denote the output of the $i$th node in layer $j$.

In layer 1, every node $i$ is an adaptive node with node function

$$O_i^1 = \mu_{A_i}(x) \quad i = 1, 2$$

or

$$O_i^1 = \mu_{B_{i-2}}(y) \quad i = 3, 4$$

where $x$ (or $y$) is the input to the $i$th node and $A_i$ (or $B_{i-2}$) is a linguistic label associated with this node. The membership functions for $A$ and $B$ are usually described by generalized bell functions, e.g.,

$$\mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x - p_i}{q_i}\right)^{2q_i}}$$

(8)

where $\{p_i, q_i, r_i\}$ is the parameter set.

Any continuous and piecewise differentiable functions, such as triangular-shaped membership functions, are also qualified candidates for node functions in this layer [28]. Parameters in this layer are referred to as premise parameters.

In layer 2, each node $\prod$ multiplies incoming signals and sends the product out

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2$$

(9)

Hence, each node output represents the firing strength of a rule.

In layer 3, each node $N$ computes the ratio of the $i$th rules’ firing strength to the sum of all rules’ firing strengths

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2$$

(10)

The outputs of this layer are called normalized firing strengths.

In layer 4, each node computes the contribution of the $i$th rule to the overall output

$$O_i^4 = \bar{w}_iz_i = \bar{w}_i(\alpha_i x + \beta_i y + \gamma_i), \quad i = 1, 2$$

(11)

where $\bar{w}_i$ is the output of layer 3 and $\{\alpha_i, \beta_i, \gamma_i\}$ is the parameter set. Parameters of this layer are referred to as consequent parameters.

In layer 5, the single node $\sum$ computes the final output as the summation of all incoming signals

$$O_i^5 = \sum_i \bar{w}_iz_i = \sum_i \frac{w_i\bar{z}_i}{w_1 + w_2}$$

(12)

Thus, an adaptive network is functionally equivalent to a Sugeno-type fuzzy inference system.

In this paper, ANFIS employs PSO method to adjust the parameters of the membership functions, as in [29]. The PSO techniques have the advantage of being less computationally expensive for a given size of network topology. The membership functions considered in this study are triangular-shaped.

D. Hybrid Approach

In this section, the algorithm used to implement the proposed approach is described step-by-step.

As depicted in Fig. 4, wavelet techniques are implemented in the first and last stages. The actual time-series (electricity price data) are first decomposed into a number of wavelet coefficient signals and one approximation signal. The decomposed signals are then fed into the ANFIS at the second stage to predict the future time-series patterns for each of the signals. Finally, the predicted signals are recomposed in the last stage to form the final predicted price series.

1) First Step: Form a matrix with a set of historical data on electricity prices, arranged in $C$ columns of a matrix thereof. Each column of the array has an associated profile of prices for a particular week whose prices are known beforehand. In this first step, the matrix has six columns, corresponding to the six previous weeks to the week whose prices are to be forecasted.

2) Second Step: Select a number of columns of the previous array so that the set of values derived from it represents the real input data. In this step, appropriate inputs are selected based on a correlation analysis. The candidate inputs with correlation coefficient greater than 0.8 are selected, corresponding to four of the six weeks with the highest correlation. Correlation analysis has also been recently used by some researchers for feature selection of price forecasting [30]–[32].
3) Third Step: Decompose the input data using the WT tool available in MATLAB. The operation mode of this process is to decompose the vector with the input data selected. The decomposition is made from the choice of basis functions (wavelet family of functions), and the number of levels wanted to split the series. The signal is divided into three levels, namely, a level of approximation (A) and details (D). Fig. 1 illustrates the decomposition process. The wavelet function used is the Db4 type, which offers a good approach and ability to use a relatively small number of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster. Subsequently, in the level of decomposition, the detail series (for high frequencies) obtained is analyzed, so that they make a selection of coefficients, making the code faster.

4) Fourth Step: Get the signal from the Wavelet reorganized so that it can be submitted to the entrance of the ANFIS structure. The approach developed in this paper uses A3, along with D3 and D1, as inputs for the ANFIS.

5) Fifth Step: Train the ANFIS with the data from the implementation of the previous step. The training process allows the system to adjust its parameters as inputs/outputs submitted. The training process stops whenever the designated number of times is reached or the objective of training error is achieved. The number of epochs used was 18. This was achieved by trial and error. After defining the training data, the number of times and the type of membership functions, the system of neuro-fuzzy inference is optimized by adapting the parameters of membership functions. The PSO is used to train the parameters associated with the membership functions of fuzzy inference system.

6) Sixth Step: Create a vector with N-dimension, where N equals the number of membership functions. This vector contains the parameters of membership function and will be further optimized by PSO algorithm. The fitness function is defined as the mean squared error.

7) Seventh Step: Define the parameters associated with the PSO algorithm (Table I). Parameters are initialized randomly in first stage and then are being updated using PSO algorithm. In each iteration, one of the parameters of membership function is being updated. In other words, in the first iteration, for example, \( p_i \) is updated, then in the second iteration, \( q_i \) is updated. Then after updating all parameters again, the first parameter update is considered and so on [33]. These parameters are grouped in a vector that is being updated iteration to iteration. The PSO algorithm used to optimize parameters of membership functions is described below: 1) initialize the population positions and speeds. For each particle, the position and velocity vectors are randomly initialized with the same size as presented by the size of the problem; 2) assess the ability of the individual particle (\( P_{best} \)). If the value is better than the current value of the individual particle, \( P_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value. If the best of all the particles of individual values is better than the overall value of current \( G_{best} \), reset the current position of the particle and update the individual value.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>PARAMETERS OF PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Value</td>
</tr>
<tr>
<td>Number of particles</td>
<td>25</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>2000</td>
</tr>
<tr>
<td>Cognitive acceleration ( c_1 )</td>
<td>2.0</td>
</tr>
<tr>
<td>Social acceleration ( c_2 )</td>
<td>2.0</td>
</tr>
<tr>
<td>Initial inertia weight ( \omega_{\text{min}} )</td>
<td>0.9</td>
</tr>
<tr>
<td>Final inertia weight ( \omega_{\text{max}} )</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Fig. 4. Structure of the proposed price forecast strategy.
number or the result reached a minimum error set, then stop the iteration and collect the best solution.

8) Eighth Step: Extract the output of the ANFIS using the parameters found by the PSO.

9) Ninth Step: Use wavelet again to reconstruct the price series forecast given by ANFIS. The final output corresponds to the prediction of our hybrid WPA approach.

III. FORECASTING ACCURACY EVALUATION

To evaluate the accuracy in forecasting electricity prices, the mean absolute percentage error (MAPE) is considered, as in [4]–[13].

The MAPE criterion is defined as follows:

\[
\text{MAPE} = \frac{100}{N} \sum_{h=1}^{N} \frac{|\hat{p}_h - p_h|}{\bar{p}}
\]

\[
\bar{p} = \frac{1}{N} \sum_{h=1}^{N} p_h
\]

where \(\hat{p}_h\) and \(p_h\) are, respectively, the forecasted and actual electricity prices at hour \(h\), \(\bar{p}\) is the average price of the forecasting period, and \(N\) is the number of forecasted hours. For weekly MAPE, \(N\) equals to 168. The average price is used in (13) to avoid the adverse effect of prices close to zero [34].

A measure of the uncertainty of a model is the variability of what is still unexplained after fitting the model, which can be measured through the estimation of the variance of the error. The smaller this variance, the more precise is the prediction of prices [5].

Consistent with definition (13), weekly error variance can be estimated as

\[
\sigma^2_{e,\text{week}} = \frac{1}{168} \sum_{h=1}^{168} \left( \frac{|\hat{p}_h - p_h|}{\bar{p}} - (e_{\text{week}}) \right)^2
\]

\[
e_{\text{week}} = \frac{1}{168} \sum_{h=1}^{168} \frac{|\hat{p}_h - p_h|}{\bar{p}}.
\]

The weekly MAPE and error variance are used in the case study.

IV. RESULTS

The hybrid WPA approach is applied to forecast next-week (168 h) prices in the electricity market of mainland Spain. Price forecasting is computed using historical data of year 2002 for the Spanish market, available at [35]. It should be noted that the Spanish market is a duopoly with a dominant player, resulting in price changes related to the strategic behavior of the dominant player, which are hard to predict [5].

For the sake of simplicity and clear comparison, no exogenous variables are considered. Also, for the sake of a fair comparison, the same test weeks as in [4]–[13] are selected, which correspond to the four seasons of year 2002.

Different sets of lagged prices have been proposed as input features for price forecasting in the Spanish market. For the proposed price forecasting model, 168-h ahead predictions are computed, taking into account the hourly historical price data of the six weeks (or 42 days) previous to the week whose prices are to be forecasted. Thus, the forecaster output directly provides a vector of dimension equal to the length of the forecasting horizon (168-h ahead).

Numerical results with the hybrid WPA approach are shown in Figs. 5–8, respectively, for the winter, spring, summer, and fall weeks.
Table II shows a comparison between the hybrid WPA approach and ten other approaches (ARIMA, mixed-model, NN, wavelet-ARIMA, WNN, FNN, HIS, AWNN, NNWT, and CNEA), regarding the MAPE criterion. The proposed approach presents better forecasting accuracy: the MAPE for the Spanish market has an average value of 5.07%.

Improvement in the average MAPE of the proposed approach with respect to the ten previous approaches is 49.1%, 45.5%, 43.1%, 37.5%, 37.0%, 32.6%, 27.3%, 24.9%, 23.8%, and 4.7%, respectively.

In addition to the MAPE, stability of results is another important factor for the comparison of forecast approaches. Table III shows a comparison between the hybrid WPA approach and eight other approaches (ARIMA, NN, wavelet-ARIMA, FNN, AWNN, NNWT, HIS, and CNEA), regarding weekly error variances. Note that the average error variance is smaller for the hybrid WPA approach, indicating less uncertainty in the predictions.

Improvement in the average error variance of the proposed approach with respect to the eight previous approaches is 70.7%, 61.4%, 57.8%, 50.0%, 43.8%, 27.0%, 25.0%, and 25.0%, respectively. For the WNN and mixed-model, the error variance has not been presented in the respective references.

The four plots of Fig. 9 provide daily errors for the considered four weeks, using the NN, NNWT, and the hybrid WPA approach, respectively.
Note that the performance of the hybrid WPA approach is generally better than the performance of the NN and NNWT approaches. However, the superiority of the hybrid WPA approach is more apparent observing Tables II and III. Moreover, the proposed approach presents lower modeling complexity: the average computation time is less than 1 min using MATLAB on a PC with 1 GB of RAM and a 2.0-GHz-based processor. Instead, the computation time required by the recently proposed CNEA approach [12] is about 40 min. Hence, the proposed approach presents not only better forecasting accuracy but also lower modeling complexity, which is important for real-life applications.

V. Conclusion

A novel hybrid approach is proposed in this paper for short-term electricity prices forecasting. The proposed approach is based on the combination of wavelet transform, particle swarm optimization, and adaptive-network-based fuzzy inference system. The application of the proposed approach to electricity prices forecasting on the Spanish market is both novel and effective. The MAPE has an average value of 5.07%, while the average computation time is less than 1 min. Hence, the proposed approach presents better forecasting accuracy with an acceptable computation time, taking into account the results of previous publications. Selection of the best input features for price forecasting can be a matter for future research.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which greatly helped them to clarify and improve the contents of the paper.

REFERENCES

IEEE TRANSACTIONS ON POWER SYSTEMS


