Market data analysis and short-term price forecasting in the Iran electricity market with pay-as-bid payment mechanism

N. Bigdeli\textsuperscript{a}, K. Afshar\textsuperscript{a,}\textsuperscript{*}, N. Amjady\textsuperscript{b}

\textsuperscript{a} EE Department, IKIU, Qazvin, Iran
\textsuperscript{b} EE Department, Semnan University, Semnan, Iran

\textbf{Abstract}

Market data analysis and short-term price forecasting in Iran electricity market as a market with pay-as-bid payment mechanism has been considered in this paper. The data analysis procedure includes both correlation and predictability analysis of the most important load and price indices. The employed data are the experimental time series from Iran electricity market in its real size and is long enough to make it possible to take properties such as non-stationarity of market into account. For predictability analysis, the bifurcation diagrams and recurrence plots of the data have been investigated. The results of these analyses indicate existence of deterministic chaos in addition to non-stationarity property of the system which implies short-term predictability. In the next step, two artificial neural networks have been developed for forecasting the two price indices in Iran’s electricity market. The models' input sets are selected regarding four aspects: the correlation properties of the available data, the critiques of Iran's electricity market, a proper convergence rate in case of sudden variations in the market price behavior, and the omission of cumulative forecasting errors. The simulation results based on experimental data from Iran electricity market are representative of good performance of the developed neural networks in coping with and forecasting of the market behavior, even in the case of severe volatility in the market price indices.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

The electric power industry in many countries all over the world is moving from a centralized operational approach to a competitive one. The understanding of electric power supply as a public service is being replaced by the notion that a competitive market is a more appropriate mechanism to supply energy to consumers with high reliability and low cost [1].

An electricity market usually includes two instruments to facilitate trading among power producers and consumers: the pool, which is an e-commerce marketplace, and a framework to enable physical bilateral contracts. Financial contracts to hedge against risk of price volatility are possible and advisable, but they do not affect the physical operation of the system [1].

In the pool, generating companies (GENCOs) submit generation bids and their corresponding bidding prices, and consumers do the same with consumption bids. The market operator (MO) uses a market-clearing tool to clear the market. This tool is normally based on single-round auctions [2].

In the electricity markets, the pricing mechanism is an important issue. It can be either uniform pricing (UP) or pay-as-bid (PAB) pricing. In the electricity markets, the marginal bid block sets the market-clearing price (MCP). Under the UP structure, MCP is paid to every winning block. However, in the PAB structure, every winning block gets its bid price as its income [3, 4]. Therefore, proper bidding strategy is critical in profit maximizing in electricity markets and so the bid blocks should be generated regarding the market price indices. Usually, there is limited information about market indices. Therefore, producers and consumers rely on price forecast information to prepare their corresponding bidding strategies [5, 6]. If a producer has a good forecast of next-day market-clearing prices it can develop a strategy to maximize its own benefit [7] and establish a pool bidding technique to achieve its maximum benefit. Similarly, once a good next-day price forecast is available, a consumer can derive a plan to maximize its purchased electricity from the pool. This requirement is more highlighted as the exchanged load in the market experiences hourly, daily and seasonal oscillations which are related to calendar, climate and some other reasons. This property and the required balance between generation and load cause volatility and even spikes in electricity price [2].

In UP markets, if no congestion occurs in the transmission network, there is one important price index to forecast: MCP [8].
In the literature, there are many efforts to develop methods for day-ahead forecasting of MCP. For example, time series models [1], jump diffusion/mean reversion models [9], artificial neural networks [6,10–13], GARCH model [14], wavelet transform [15] and ARIMA models [16] have been proposed for this purpose. Among these methods, artificial neural network is a good candidate due to its high ability in modeling hidden non-linear relationships between inputs and outputs of a system [17]. However, if congestion exists in the network, some other price indices such as zonal market-clearing price (ZMCP) or locational marginal price (LMP) should be considered instead of MCP [2].

The PAB markets experience a different situation. While in the literature, there are fewer efforts for price forecasting in this area, there are different price indices which may be important to forecast for a proper bidding strategy. The most important factors are hourly weighted average price (WAP) and hourly minimum/maximum accepted price. The hourly WAP in a PAB electricity market is defined as the average of the accepted price stages in the market, where each price has been weighted by the amount of its underlying accepted energy in the market, for each hour. The hourly minimum/maximum accepted price is the lowest/highest value of the accepted price stages in the market, for each hour. In an un-congested market, WAP is used for profit computation and risk analysis for bidding at a special price. Minimum/maximum accepted price, on the other hand, is related to the foremost/upmost price blocks in the bidding curves.

In this paper, we focus on pool-based PAB electricity markets. Two artificial neural networks are proposed for forecasting hourly WAP and hourly minimum accepted price (MAP) in Iran’s electricity market as a PAB market. The maximum accepted price has been ignored here. The reason is that in Iran’s electricity market the cap of affordable prices is predetermined by the market regulator. Due to the congestion status of the market, the maximum accepted price is constant [18]. The available data for development of these models are the corresponding price time series (WAP and MAP) and the forecast of hourly required load (RL). The term hourly required load is the total energy which should be exchanged in the market for each hour. It contains the effect of some phenomena such as climate properties (such as temperature and humidity) and the calendar properties as well. Because of the special rules in Iran’s electricity market, the prices should be forecasted 3 days earlier to send the bidding curves to the market on time. Since the market is cleared and announced 1 day ago, in choosing inputs, the WAP time series up to 2 days ago are practically available. However, data of RL is available up to desired hour. More details about Iran’s electricity market can be found [19].
In this paper, our primary systematic approach includes the predictability analysis of the available experimental data as well as data correlation analysis for the selection of proper models' inputs. The former includes the analysis of the bifurcation diagrams and recurrence plots. The results are representative of existence of deterministic chaos in addition to non-stationarity in the system. These properties imply just short-term predictability of the system. From the latter, the models' input sets are selected regarding four aspects: the correlation properties of the available data, the critiques of Iran’s electricity market, a proper convergence rate in case of sudden variations in the market price behavior, and the omission of cumulative forecasting errors. Afterwards, the neural network models have been derived based on the experimental data from Iran’s electricity market for about 625 days. The performance of developed networks is evaluated via experimental testing sets that contain sudden changes in \( WAP \) and \( MAP \) ranges as well as variations in shape, average and variance of hourly \( WAP \) and \( MAP \) values. The results are representative of good performance of developed neural networks in forecasting market behavior even in the case of severe price variations.

The rest of the paper is organized as follows: in Section 2, the properties of the available data and their correlation analysis results are stated. The developed artificial neural networks and their evaluation results are brought in Section 3, and the paper is concluded.
Fig. 4. (a) Autocorrelation function plot of WAP, (b) the cross-correlation of WAP with RL, (c) autocorrelation function plot of MAP, and (d) the cross-correlation of MAP with RL versus time lag.

in Section 4. Two appendices, i.e. Appendices A and B contain the mathematical background of data analysis and the fundamentals of forecasting with artificial neural networks (ANNs), respectively.

2. The available data and correlation analysis

2.1. The available data and its general properties

Seeking for predictability and proper inputs for our models, in this section the experimental data from Iran’s electricity market are closely examined. The available data are the $RL$ in MWh as well as $WAP$ and $MAP$ in US$ for 844 days from 24 September 2005 up to 15 December 2007. The mentioned data have been shown in Fig. 1, where the first plot shows $RL$ and the second and third plots show the $WAP$ and $MAP$ versus time in hours, respectively. To get a better view, this figure has been zoomed out for some 2500 h periods in Fig. 2. From the first graph in these two figures, a quasi-periodic behavior (with period 24 h) is observed for daily variations in $RL$ whose variance is almost constant and its average changes continuously. The patterns in the second and third graphs are completely different as the periodicity disappears in $WAP$ for example in summer (Fig. 2(b)) and in other days the daily pattern is different in different days. For $MAP$, abrupt irregular changes are also observed in addition to other various patterns.

From one point of view, $RL$ can be considered as system parameter and $WAP$ and $MAP$ as the system operating points. Based on this interpretation, the dot plots of $WAP$ and $MAP$ values versus $RL$ have been shown in Fig. 3(a) and (b), respectively. From Fig. 3(a), it is observed that for low values of $RL$ the $WAP$ value bifurcates. This bifurcation diagram may, however, be due to a chaotic or stochastic nature of a non-linear process [20–22]. Since we are interested in predictability, it is important for us to distinguish between these two types of processes. In order to get a better view about the behavior and characteristics of the underlying system, the recurrence plots (RPs) have been considered as our tool which will be referred to them later in this section. In the following, at first, the correlation analysis results of the underlying time series, i.e. $WAP$ and $MAP$ values, versus $RL$ will be presented. Afterwards, the related results about analysis of our system with RP will be brought.

2.2. Correlation analysis results

The mathematical formulation of cross-correlation has been brought in Appendix A, Part I. The plots in Fig. 4(a) and (b) show the autocorrelation function plot of WAP, and the cross-correlation of $WAP$ with $RL$, respectively. From Fig. 4(a), it is seen that $WAP$ time series is highly correlated with its lagged values of 24k to 24k + 5 where k is a non-negative integer. This correlation slowly decays as k increases and may be taken as a hallmark of predictability. Fig. 4(b) represents similar behavior but with lower correlation between $WAP$ and $RL$. From this figure, it is concluded that $RL$ time series should be also accounted for prediction of $WAP$.

Fig. 4(c) and (d) shows the autocorrelation function plot of $MAP$, and the cross-correlation of $MAP$ with $RL$, respectively. Considering these graphs, although the overall correlation properties of $MAP$ are relatively similar to those of $WAP$ but one point is highlighted. The autocorrelation function of $MAP$ is rather decreasing as k increases. This point is representative of lower self-predictability of this time
series. On the other hand, the cross-correlation of MAP with RL is relatively more and decays slowly. This means that the role of RL in prediction of MAP is more important. Some additional points should be also mentioned. In calculation of cross/auto-correlation functions (Eq. (A.1)), a linear system is implicitly assumed. In other words, in correlation computations, no non-linear relationship between different time series is taken into account. Besides, due to use of expectation $E\{\cdot\}$ in this equation, no conclusion about the seasonality and/or non-stationarity of the system may be derived. These points for a complex system such as electricity market are of importance. In order to take a sense about these phenomena, consider the plots in Fig. 1, where in durations 5000–9000 and 14,000–18,000, the RL shows similar behavior, while the WAP and MAP behaviors are completely different.

2.3. RP analysis results

The fundamentals of recurrence plots have been presented in Appendix A, Part II. Recurrence plots are used as a tool for analyzing experimental time-series data. They are especially useful for finding hidden correlations in complex data sets and to determine the stationarity of a time series. With RP, one can graphically detect hidden patterns and structural changes in data or see similarities in patterns across the time series under study. The RPs of the RL and the WAP data sets have been shown in Fig. 5. These graphs have been generated with the CRP toolbox [23] of MATLAB [24]. To plot these figures, the parameter $d$ has been derived as 3 (from false nearest method), parameter $\tau$ has been derived as 5 and 6 (for the first minimum of mutual information function of the data), and $\varepsilon$ has been chosen 0.7 and 0.35 (for normalized data) for Fig. 5(a) and (c), respectively. The plots in Fig. 5(a) and (c) illustrate the RP for whole data. In Fig. 5(b) and (d), a smaller region of the plots in Fig. 5(a) and (c) has been zoomed out to imply a better sense about the system properties.

From Fig. 5(a), it is observed that the RL time series exhibits seasonal non-stationary behavior. In Fig. 5(b), relatively regular structure with almost long diagonals is also representative of a deterministic quasi-periodic behavior in this time series. Variable slope of these diagonals are due to non-linear dynamics of the time series.

Now, consider the graphs in Fig. 5(c) where the RPs of WAP time series has been shown. In Fig. 5(c), presence of the dark quarters is representative of seasonality and non-stationarity in the hourly WAP time series. In this figure, presence of relatively long diagonals in the leftmost down and rightmost up corners, are representative of quasi-periodic stable dynamics in these regions. In the four dark middle parts, the system behavior seems drastically more complex. In these regions, the existence of white ribbons is representative of short-term non-stationarity in the WAP dynamics. Also, short irregular diagonals are indicative of unstable periodic orbits and even chaos in the WAP behavior. This behavior implies just short-term predictability of this time series.

After the analysis of RPs, now consider the CRP and JRP graphs of RL and WAP which have been shown in Fig. 6. In Fig. 6(a), the
existence of short diagonals and disappearance of the main diagonal represents short-term correlation of these two time series. As another point, note that the existence of different patterns in this figure is also representative of non-stationary correlation between RL and WAP. As a result, the predictability of WAP with respect to RL will be different in various time periods. Therefore, in order to synthesize any model for WAP forecasting, one should consider as fewer time lagged data (WAP and RL) as possible in the model input set. The JRP graph in Fig. 6(b) represents similar results. However, the appearance of the main diagonal in this figure is due to the 24 h quasi-periodic nature of these signals. For MAP, frequent abrupt fluctuations are also observed, which are due to generating units' bidding strategies. These fluctuations make its behavior more complex and less predictable (see Fig. 1(c)). The RP analysis results have not been brought here due to space limitations.

3. Design and evaluation of artificial neural network models

The analysis of the RPs in the previous section are representative of the short-term predictability of WAP and MAP time series and existence of short-term correlation between these time series and RL. Also, it was shown that the cross- and auto-correlation functions of these time series could be employed to determine the effective lagged values to forecast WAP and MAP for short future. Based on these achievements, in this section, we try to synthesize proper models for the forecasting these time series. Due to the high performance of neural networks in modeling non-linear complex systems [25–29], they have been considered as the modeling tool here. In Appendix B, the basics of prediction with neural networks have been shortly reviewed. In this section, the trained networks and their performance analysis results are presented.

3.1. WAP modeling and evaluation

A multi-layer perceptron (MLP) feed forward neural network with three hidden layers is proposed here for forecasting hourly WAP values in terms of its past values as well as hourly RL values. Based on the analysis results of the last section as well as critical rules of Iran’s electricity market (described by part in Section 1), the inputs of the developed network have been selected as the following 23 inputs:

- The forecast of RL for the desired hour and its five previous hours \((l_0, l_1, l_2, l_3, l_4, l_5)\).
- The RL of the same hour of 4 previous days \((l_{-24}, l_{-48}, l_{-72}, l_{-96})\).
- WAP for the same hour and its five previous hours of 2 days beforehand \((p_{-48}, p_{-49}, p_{-50}, p_{-51}, p_{-52}, p_{-53})\).
- WAP for the same hour and its five previous hours of 3 days beforehand \((p_{-72}, p_{-73}, p_{-74}, p_{-75}, p_{-76}, p_{-77})\).
- WAP for the same hour of 4 days beforehand \((p_{-96})\).

In order to find the most proper network input set, several combinations of inputs were evaluated. Amongst, the selected input set, in one hand, considers the correlation properties of the available data and, on the other hand, implies a proper convergence rate in the case of sudden variations in the market behavior. The other advantage of the selected input set and so the trained network is that the predicted WAP for each hour is just related to the real data and not to the previous forecasted values of WAP. This means that the error of forecasting is neither propagated nor cumulative.

The best network architecture has been also found via evaluation of different structures. These structures include networks with different numbers of hidden layers, different numbers of neurons in each layer and different types of transfer functions. We converged to a configuration consisting of three hidden layers and number of neurons as: 23 for input layer, 3, 5 and 8 for hidden layers and 1 for output layer. All of the input data were normalized between zero and 1 by dividing the load terms by its maximum value, i.e. 34,000 MWh, and the price terms by maximum possible price in the market. Based on this normalization, the transfer function for input and hidden layer neurons has been selected as a log sigmoid transfer function. The transfer function used for the output layer is a pure linear transfer function. For training the network, the neural network toolbox of MATLAB was selected due to its flexibility and simplicity [24] and the network was trained by the Levenberg–Marquardt algorithm [28]. The available data are the data for 20,256 h (844 days) from which the first 15,000 data (related to the first 625 days) have been used for training. The remainder is the data for 219 days which have been employed as our testing set. The consumed time for training this network (on a Pentium 4, 2 GHz, Core2 CPU Laptop with 1 GB RAM) is about 51.4 s, while the consumed time for computation of the network output for the above-mentioned testing set is about 0.025 s.

In Fig. 7, the results of applying the testing data set to the trained network in comparison with the actual outputs are shown. Fig. 7(a)

![Fig. 6. (a) CRP and (b) JRP of WAP time series versus RL time series.](image)
shows the prediction results in comparison with the actual values of WAP for all the testing period, and it has been magnified in Fig. 7(b)–(d) for different time intervals to give a better view. As seen, the testing set has been chosen such that it contains sudden changes in the WAP range. The evaluation indices introduced in Eqs. (B.1)–(B.4) have been calculated for both the training and testing data sets and the results have been brought in Table 1. For MAE and RMSE, both absolute and percentage values are shown in the table. The percentage values have been derived by dividing the errors to the actual values and then averaging them. As seen, very low values of the evaluation indices are once again representative of the very good performance of the developed model to predict the WAP behavior for a wide range of time and variations.

In order to show the effectiveness of the trained network, the performance of the trained network has been compared with some other ANNs with different input sets and structures in Table 2. In this table, Modified MAPE, defined in Eq. (B.4), has been considered as the performance evaluation index. Five input sets have been considered for ANN training, where Set 1 is the above-mentioned 23-input set. Sets 2–5 are some slightly changed sets with respect to Set 1 as follows:

- Set 2 is the same as Set 1 except that less RL terms have been considered in Set 2. \( l_{0}, l_{-1}, l_{-2}, l_{-3}, l_{-4}, l_{-5} \) have been omitted from this input set. Therefore, number of inputs in Set 2 will be 21.
- Set 3 is the same as Set 1 except that it considers more RL terms. \( l_{-6}, l_{-7}, l_{-8} \) have been added to Set 3. Therefore, number of inputs in Set 3 will be 26.
- Set 4 is the same as Set 1 except that it considers less WAP terms. \( p_{-.51}, p_{-.52}, p_{-.53}, p_{-.75}, p_{-.76}, \) and \( p_{-.77} \) have been omitted from Set 4. Therefore, number of inputs in this case will be 17.
- Set 5 is the same as Set 1 except that it considers more WAP terms. \( p_{-.54}, p_{-.55}, p_{-.56}, p_{-.78}, p_{-.79}, \) and \( p_{-.80} \) have has been added to Set 5. Therefore, number of inputs in this case will be 29.

In Table 2, the ANNs structures have been represented as ordered \( n \)-tuples including number of neurons in input, hidden and output layers, from left to right, respectively. The results are representative of the proper selection of both input set and ANN structure, as stated before.

### Table 2

<table>
<thead>
<tr>
<th>Structure</th>
<th>Input set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1 (proposed)</td>
<td>1.64 1.79 2.87 1.85 1.93</td>
</tr>
<tr>
<td>(of inputs, 3, 5, 8, 1)</td>
<td>1.88 1.89 2.86 1.89 1.88</td>
</tr>
<tr>
<td>(of inputs, 5, 5, 8, 1)</td>
<td>1.73 1.81 2.74 1.72 1.89</td>
</tr>
<tr>
<td>(of inputs, 3, 3, 8, 1)</td>
<td>1.79 1.86 2.9 1.69 1.89</td>
</tr>
<tr>
<td>(of inputs, 3, 7, 8, 1)</td>
<td>1.68 1.79 3 1.75 1.78</td>
</tr>
<tr>
<td>(of inputs, 3, 5, 5, 1)</td>
<td>1.69 1.71 2.83 1.78 1.75</td>
</tr>
<tr>
<td>(of inputs, 3, 5, 11, 1)</td>
<td>1.75 1.76 2.73 1.7 1.76</td>
</tr>
<tr>
<td>(of inputs, 3, 7, 1)</td>
<td>1.9 1.73 2.5 1.94 1.99</td>
</tr>
<tr>
<td>(of inputs, 5, 5, 8, 11)</td>
<td>1.71 1.8 2.91 1.74 1.85</td>
</tr>
<tr>
<td>(of inputs, 5, 5, 7, 12, 1)</td>
<td>1.8 1.79 2.38 1.72 1.82</td>
</tr>
</tbody>
</table>

### Table 1

<table>
<thead>
<tr>
<th>Data set</th>
<th>Index</th>
<th>MAE</th>
<th>RMSE</th>
<th>Modified MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Absolute (US$) %</td>
<td>Absolute (US$) %</td>
<td>%</td>
</tr>
<tr>
<td>Training set</td>
<td>0.79</td>
<td>1.17</td>
<td>0.95</td>
<td>1.41 1.37</td>
</tr>
<tr>
<td>Testing set</td>
<td>1.03</td>
<td>1.42</td>
<td>1.02</td>
<td>1.42 1.645</td>
</tr>
</tbody>
</table>

The same procedure as WAP neural network modeling has been repeated for MAP, too. The result is a MLP with two hidden layers and 18 inputs:

- The forecast of RL for the desired hour and its five previous hours \( (l_{0}, l_{-1}, l_{-2}, l_{-3}, l_{-4}, l_{-5}) \).
The RL of the same hour of 4 previous days \((l_{-24}, l_{-48}, l_{-72}, l_{-96})\).

- MAP for the same hour and its five previous hours of 3 days beforehand \((mp_{-48}, mp_{-49}, mp_{-50}, mp_{-51}, mp_{-52}, mp_{-53})\).
- MAP for the same hour of 3 and 4 days beforehand \((mp_{-72}, mp_{-96})\).

As seen, because of existing more fluctuations in MAP time series and its less autocorrelation, here, the number of underlying price inputs has been reduced with respect to the inputs of the WAP model. Once again the input set and so the trained network has the advantage that the predicted MAP for each hour is just related to the available actual data and not to its previous forecasted values which means the omission of cumulative forecasting error.

The configuration of the trained network consists of two hidden layers and number of neurons as: 18 for input layer, 3 and 7 for hidden layers, and 1 for output layer, respectively. All of the input data are normalized between zero and 1 by dividing the load terms by 34,000 MWh and the price terms by maximum possible price in the market. The transfer function for input and hidden layer neurons has been selected as log sigmoid and for the output layer is a pure linear transfer function. For training the network, the Levenberg–Marquardt algorithm has been employed. The available data are the data for 20,256 h (844 days) from which the first 10,008 data (related to the first 417 days) have been used for training. The remainder is the data for 427 days which have been employed as our testing set. The training set for MAP is different with that of WAP because it is desired that the testing set includes severe variations as well. The consumed time for training this network (on a Pentium 4, 2 GHz, Core2 CPU Laptop with 1 GB RAM) is about 5.64 s, while the consumed time for computation of the network output for the above-mentioned testing set is about 0.032 s.

In Fig. 8, the results of applying the testing data set to the trained network in comparison with the actual outputs have been shown. In Fig. 8(a), the output of network has been compared with its actual values for all the testing data set, and it has been magnified in Fig. 8(b)–(d) for different time intervals to give a better view. Once again, the testing set has been chosen such that it contains sudden changes in MAP. As seen, in spite of variations in shape, average and variance of MAP the trained network works well. However, due to severe and irregular changes in MAP, the prediction error in Fig. 8(c) is larger than that of other figures such as Fig. 8(b) and (d). This unavoidable behavior has also been reflected in the evaluation indices (Eqs. (B.1)–(B.4)) calculated for the MAP training and testing sets listed in Table 3.

In order to show the properness of the trained network, the performance of the trained network has been compared with some

![Fig. 8](image)

**Table 3**

<table>
<thead>
<tr>
<th>Data set</th>
<th>Index</th>
<th>MAE Absolute (US$)</th>
<th>RMSE Absolute (US$)</th>
<th>Modified MAPE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>2.04</td>
<td>3.6</td>
<td>4.44</td>
<td>6.9</td>
</tr>
<tr>
<td>Testing set</td>
<td>3.12</td>
<td>4.82</td>
<td>7.7</td>
<td>11.90</td>
</tr>
</tbody>
</table>

**Table 4**

<table>
<thead>
<tr>
<th>Structure</th>
<th>Input set</th>
<th>Set 1 (proposed)</th>
<th>Set 2</th>
<th>Set 3</th>
<th>Set 4</th>
<th>Set 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(# of inputs, 3, 7, 1)</td>
<td>8.63</td>
<td>8.9</td>
<td>9.04</td>
<td>9.2</td>
<td>9.01</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 1, 7, 1)</td>
<td>10.52</td>
<td>10.56</td>
<td>10.47</td>
<td>10.71</td>
<td>10.51</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 5, 7, 1)</td>
<td>10.1</td>
<td>9.14</td>
<td>9.22</td>
<td>10.27</td>
<td>9.41</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 3, 5, 1)</td>
<td>8.68</td>
<td>8.87</td>
<td>8.78</td>
<td>9.07</td>
<td>12.28</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 3, 9, 1)</td>
<td>8.9</td>
<td>9.23</td>
<td>9.44</td>
<td>10.17</td>
<td>9.15</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 5, 1)</td>
<td>9.8</td>
<td>9.73</td>
<td>9.44</td>
<td>Diverges</td>
<td>9.65</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 3, 3, 1)</td>
<td>10.48</td>
<td>10.44</td>
<td>12.44</td>
<td>Diverges</td>
<td>10.55</td>
<td></td>
</tr>
<tr>
<td>(# of inputs, 3, 5, 1)</td>
<td>9.14</td>
<td>10.28</td>
<td>10.65</td>
<td>10.59</td>
<td>10.39</td>
<td></td>
</tr>
</tbody>
</table>
other ANNs. Input sets and structures of these ANNs are different. Five input sets are considered for these ANNs as follows:

- Set 1 is the above-mentioned 18-input set.
- Set 2 is the same as Set 1 except that it considers less RL terms. \( l_{-4} \) and \( l_{-5} \) have been omitted from Set 2. Therefore, number of inputs in this case will be 16.
- Set 3 is the same as Set 1 except that it considers more RL terms. \( l_{-6}, l_{-7} \) and \( l_{-8} \) have been added to Set 3. Therefore, number of inputs in this case will be 21.
- Set 4 is the same as Set 1 except that it considers less MAP terms. \( mp_{-51}, mp_{-52} \), and \( mp_{-53} \) have been omitted from Set 4. Therefore, number of inputs in this case will be 15.
- Set 5 is the same as Set 1 except that it considers more MAP terms. \( mp_{-54}, mp_{-55} \), and \( mp_{-56} \) have been added to Set 5. Therefore, number of inputs in this case will be 21.

\[ \text{Modified MAPE}, \text{ defined in Eq. (B.4), has been considered as the performance evaluation index and the comparison results have been shown in Table 4. Once again, the } n\text{-tuples in Table 4 are representative of the structures adopted for the ANNs, which contain number of neurons in input, hidden and output layers, from left to right, respectively. The results of Table 4 indicate proper selection of both input set and ANN structure in this case, too.} \]

4. Conclusions

In this paper, market data analysis and short-term price forecasting in the Iran electricity market with pay-as-bid payment mechanism has been considered. The data analysis procedure includes both correlation and predictability analysis of the most important load and price indices, i.e. RL, WAP, and MAP. For the predictability analysis, the WAP-RL and MAP-RL bifurcation diagrams as well as their individual and cross/joint recurrence plots have been studied. The employed data are the experimental time series from Iran’s electricity market in its real size and is long enough to make it possible to take the properties such as non-stationarity of the market into account. Based on these investigations, it is observed that the WAP time series performs deterministic chaos with non-stationary properties. These properties imply a short-term predictability of this time series. For the MAP time series the observed sudden irregular variations reduce its predictability to some extent.

In order to forecast the underlying market indices, i.e. WAP and MAP, two artificial neural networks have been developed. The correlation analysis results and the critiques of Iran’s electricity market have been considered to select proper input sets for the developed neural networks. The trained neural networks, have the advantage that the predicted values of WAP and MAP for each hour are just related to the available actual data and not to the previous forecasted values which results in the omission of cumulative forecasting error. The simulation results based on real data from Iran’s electricity market show that the trained networks not only work well in normal operation regimes in spite of variations in shape, average and variance of the market parameters, but also converge rapidly even in the case of severe volatility in the market price indices.

Appendix A

In this appendix, at first, the mathematical formulation of cross-correlation is shortly described and then the fundamentals of recurrence plots are presented.

A.1. Cross-correlation

One of the most common tools for determination of effective inputs for a forecasting model is the correlation analysis between the underlying time series. The cross-correlation between two time series \( X = \{x_i, i = 1, \ldots, N\} \) and \( Y = \{y_i, i = 1, \ldots, N\} \) can be defined as:

\[
C_{XY}(m) = E(x_i y_{i+m}) - E(X) E(Y^*), \quad m = 1, \ldots, N
\]  

where in Eq. (A.1), \( E(\cdot) \) and * stand for mathematical expectation and complex conjugate, respectively.

A.1.1. Recurrence plots (RPs)

RPs were firstly introduced by Eckman et al. [30] as a tool for analyzing experimental time-series data. For deriving an RP first of all all the dynamics of phase space trajectories of signal must be reconstructed via say “method of delays [31]”. From taken embedding theorem [32], the phase space trajectories of system can be reconstructed from a finite output time series \( u_t \) of system via definition of state vectors \( \vec{x}_i \) as follows:

\[
\vec{x}_i = [u_i, u_{i+\tau}, \ldots, u_{i+(d-1)\tau}]^T
\]  

(A.2)

In this equation, the time delay \( \tau \) is equal to the first minimum of the mutual information function [33] and the embedding dimension \( d \) is derived from standard methods such as false nearest neighbors’ method [31]. After reconstructing the phase space, RPs visualize the behavior of trajectories in the phase space [33,34] via a graphical representation of matrix:

\[
R_{ij} = \Theta(e - ||\vec{x}_i - \vec{x}_j||), \quad i, j = 1, \ldots, N
\]  

(A.3)

where \( \vec{x}_i \) stands for the point in the reconstructed phase space at time \( i \), and \( e \) is a predefined threshold and \( \Theta(\cdot) \) is the heaviside unit step function. \( ||V|| \) represents norm of vector \( V \), which is usually Euclidean norm. One assigns a “black” dot to the value one and a “white” dot to the value zero in drawing RPs. The two-dimensional graphical representation of \( R_{ij} \) then is called RP [35] and can be used to distinguish different dynamic systems. In this context, three main types of systems can be considered which are the periodic systems, stochastic random systems, and chaotic ones. The periodic motion is reflected by long and non-interrupted diagonals. The vertical distance between these lines corresponds to the period of the oscillation. The chaotic systems also lead to diagonals which are seemingly shorter. There are also certain vertical/horizontal distances, which are not as regular as in the case of the periodic motion. The RP of the uncorrelated stochastic signal consists of many single black points. The distribution of the points in these RPs looks rather different. Considering the three cases, we might conjecture that the shorter the diagonals in the RP, the less predictable the system [34].

In addition to RPs, which are dedicated to analysis of one time series of a system, there are extensions of them which are used for comparison or correlation analysis of two dynamics, systems, or time series. These extensions are called as cross recurrence plots (CRPs) and joint recurrence plots (JRP), which are briefly described here.

CRP is a bivariate extension of RP and has been introduced to analyze the dependencies between two different systems by comparing their states. CRP can be considered as a generalization of the linear cross-correlation function. Suppose that we have two dynamical systems, each one represented by the embedded state trajectories \( \vec{x}_i \) and \( \vec{y}_j \) in a d-dimensional phase space. Analogous to the matrix RP (Eq. (A.3)), the corresponding cross-recurrence matrix is defined by:

\[
CR^{XY}(i, j) = \Theta(e - ||\vec{x}_i - \vec{y}_j||), \quad i = 1, \ldots, N, \quad j = 1, \ldots, M
\]  

(A.4)

where the length of the trajectories \( \vec{x}_i \) and \( \vec{y}_j \) is not required to be identical, and hence the matrix CR is not necessarily square. Note
that both of the systems are represented in the same $d$-dimensional phase space, because a CRP looks for those times when a state of the first system recurs to one of the other system. If the estimated embedding parameters from the both time series are not equal, the higher embedding parameter should be chosen. However, the data under consideration should not be from the same process. Therefore, the components of $\mathbf{x}_i$ and $\mathbf{y}_i$ are usually normalized before computing the cross recurrence matrix. The graphical representation of the matrix $CR$ is called CRP. In this graph, long diagonals are representative of similarity or correlation between the two systems. A measure based on the lengths of these diagonal lines can be used to find non-linear interrelations between the two systems, which cannot be detected by the common cross-correlation function. More similarity between the two time series results in the longer diagonals and higher density of dark dots around the main diagonal of the graph. Besides, a time dilatation or time compression of one of the trajectories causes a distortion of the diagonal lines or cause curvatures in them.

As stated above, CRP can be used for similarity analysis of two time series by examining the occurrence of similar states. JRP is another bivariate extension of RP which has been developed to compare different systems’ dynamics. This method considers the recurrences of each time series to its trajectory in its respective phase space separately and looks for the times when both of them recur simultaneously, i.e. when a joint recurrence occurs. JRP considers an extended phase space $\mathbf{R}^{d_x,d_y}$, where $d_x$ and $d_y$ are the phase space dimensions of the corresponding individual systems. This type of comparison when we have two physically different systems makes more sense. JRP of the two time series $\mathbf{x}_i$ and $\mathbf{y}_i$ embedded in $d_x$ and $d_y$ dimensional phase spaces is defined as:

$$JR^{x,y}(\mathbf{x},\mathbf{y}) = \theta(\epsilon_x - |\mathbf{x}_i - \mathbf{x}_j|)\theta(\epsilon_y - |\mathbf{y}_i - \mathbf{y}_j|), \quad i,j = 1, \ldots, N \quad (A.5)$$

where $\epsilon_x$ and $\epsilon_y$ are the corresponding thresholds for time series $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively. By this method not only the individual properties of each system is preserved, but also in the extended phase space, non-linear similarities of the two systems can be investigated.

### Appendix B

#### B.1. Forecasting with artificial neural networks (ANNs)

An ANN is composed of a number of interconnected neurons which are arranged in a few layers, called input, hidden and output layers. The output of each node is a weighted sum of its inputs added to a constant term called bias. Forecasting with neural networks involves two steps: training and testing. In training, a proper ANN is constructed via some different stages. At first, proper inputs should be selected. This stage is an important stage which is usually done through input-output linear/non-linear correlation analysis and probably the experience of the ANN developer about the underlying system. The selected inputs and the output of the network are then normalized to feed them to the network. The next stage is to choose the numbers of layers and nodes of each layer as well as the transfer function of each neuron. This stage is normally a trial and error stage which is repeated until the best performance of the network is achieved. Final stage in training includes learning of the chosen network. A learning process in the neural network constructs an input–output mapping function between the selected inputs and output. The learning process is done via adjusting the weights and biases at each iteration based on the minimization of an error measure. Thus, learning stage entails an optimization process. There are different learning algorithms in the literature [36]. One of most efficient learning algorithms is the Levenberg–Marquardt algorithm. This algorithm is actually a modified Gauss–Newton method that converges 10–100 times faster than the well-known back propagation algorithm. The details of the Levenberg–Marquardt algorithm can be found in Ref. [36].

The knowledge acquired by a neural network through its learning process is then tested by applying new data that it has never been seen before, called testing set. The performance of the trained network is then evaluated by comparison of the network output with its actual value via some statistical evaluation indices. Suppose that $A_i$ and $P_i$ are the actual and network outputs related to $i$th test sample, respectively, and $N$ is the number of samples in the testing set. Then some evaluation indices are defined as follows [29]:

\begin{equation}
RMSE = \sqrt{\frac{\sum_{i=1}^{N} (P_i - A_i)^2}{N}} \quad (B.1)
\end{equation}

\begin{equation}
MAE = \frac{\sum_{i=1}^{N} |P_i - A_i|}{N} \quad (B.2)
\end{equation}

Modified mean absolute percentage error (\textit{Modified MAE}) in order to evaluate this index, at first, the average of actual output values is computed as:

$$Av = \frac{1}{N} \sum_{i=1}^{N} A_i \quad (B.3)$$

then, the Modified\textit{ MAE} will be computed as [7]:

\begin{equation}
\text{Modified\textit{ MAE}} = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_i - A_i|}{Av} \times 100\% \quad (B.4)
\end{equation}

### References


