Improving Time Series Forecasting by Discovering Frequent Episodes in Sequences

Francisco Martínez-Álvarez[1], Alicia Troncoso[1], and José C. Riquelme[2]

[1] Area of Computer Science, Pablo de Olavide University of Seville, Spain
{fmaralv,ali}@upo.es
[2] Department of Computer Science, University of Seville, Spain
riquelme@lsi.us.es

Abstract. This work aims to improve an existing time series forecasting algorithm –LBF– by the application of frequent episodes techniques as a complementary step to the model. When real-world time series are forecasted, there exist many samples whose values may be specially unexpected. By the combination of frequent episodes and the LBF algorithm, the new procedure does not make better predictions over these outliers but, on the contrary, it is able to predict the apparition of such atypical samples with a great accuracy. In short, this work shows how to detect the occurrence of anomalous samples in time series improving, thus, the general forecasting scheme. Moreover, this hybrid approach has been successfully tested on electricity-related time series.

Keywords: Time series, forecasting, outliers.

1 Introduction

This work provides a new methodology to forecast time series and, in addition, to predict the apparition of outliers. The analysis of temporal data and the forecast of future values of time series are among the most important problems that data analysts face in many fields, ranging from finance and economics, to production operations management or telecommunications. A *forecast* is a prediction of some future events.

The proposed approach is specifically framed in electricity prices time series forecasting, which is a difficult task due to the nonconstant mean and variance and significant outliers typically present in these series.

Thus, the combination of two different techniques are proposed to fulfill this goal. The first one is a general-purpose forecasting algorithm introduced in [9], called LBF. The authors obtained a previous labeling of the elements forming the time series by means of clustering techniques. The forecasting process was performed by using just the information provided by the clustering. Thus, the values of the elements in datasets were discretized and, as a result, the sequence of real values was transformed in a sequence of discrete values or labels. These labels were used to predict the future behavior of the time series, avoiding the

N. Adams et al. (Eds.): IDA 2009, LNCS 5772, pp. 357–368, 2009.

use of the real values until the last step of the process. The results returned by the algorithm, however, were not labels but the real values.

The algorithm introduced in [10] is inserted in the general scheme of forecasting with the aim of dealing with the presence of outliers. Concretely, they proposed an algorithm called Q-epiMiner that was, in fact, an improvement of the well-known serial episodes [8]. The main achievement of the Q-epiMiner algorithm was to characterize sequences of similar behavior over all occurrences, as well as providing a tree structure to organize these sequences.

Therefore, the discovery of frequent episodes is used in order to determine possible candidates to be outliers when using the LBF algorithm to forecast time series. The general outline of the new proposed approach is shown in Fig. 1.

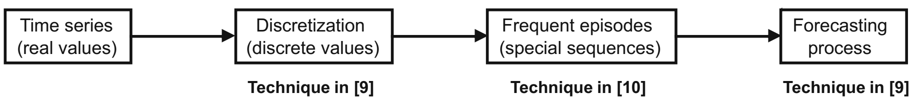

Fig. 1. General outline of the proposed methodology

The rest of the paper is organized as follows. The latest works related to time series forecasting and discovery of outliers are gathered in Section 2. Section 3 presents a brief explanation of the two existing and used algorithms in which the new approach is based on. Section 4 introduces the proposed methodology, showing how the two existing techniques are combined in order to improve the forecasting process. Section 5 shows the results obtained for the electric energy market of Spain for the year 2006, including measures of the quality of them. Finally, Section 6 summarizes the main conclusions achieved and provides clues for future work.

2 Related Work

Regarding to time series forecasting, two mixed models were proposed to obtain the forecasting of the prices for two different prediction horizons in [3]. The first one forecasted electricity prices for each of the 24 hours of the next day using ARIMA models, while the second model computed the predictions by using Bayesian information criteria.

A modification of the nearest neighbors methodology was proposed in [13]. To be precise, the approach presented a simple technique to forecast next-day electricity market prices based on the weighted nearest neighbors methodology.

Li et al. proposed a forecasting system immersed in a grid environment in [6]. In this paper, a fuzzy inference system –adopted due to its transparency and interpretability– and time series methods were proposed for day-ahead electricity price forecasting.

The authors in [12] proposed an artificial neural network-based approach to forecast the energy price in the Spanish market. Thus, a novel training method was presented and applied to the multilayer perceptron in order to improve the forecasting process.

The apparition of outliers in time series has also been widely discussed in the literature. Thus, the authors in [1] proposed a technique to detect outliers in data whose generation was difficult to model. They assumed that the correlation among data close in time is higher than those farther apart.

Another method for detecting outliers was proposed in [7], in which the authors considered two different sources of outliers –additive and innovation– in autorregressive moving-average time series. Concretely, they proposed the application of two different procedures associated to each source simultaneous and coherently.

The occurrence of spike prices (prices significantly higher than the expected value, i. e., outliers) is an usual feature associated with electricity prices-related time series. With the aim of dealing with this peculiarity, the authors in [14] proposed a data mining framework based on both support vector machines and probability classifiers.

The work described in [5] searched for patterns in electricity prices data in order to verify how the outliers may modify the behavior of such prices. To fulfill this goal, they used Box and Jenkins models, Discrete Fourier Transform series smoothing and GARCH approaches.

Also, the work in [4] discussed the use of the fractal theory to forecast the electricity price time series. For this purpose, a forecasting model based on improved fractal was built and solved to forecast short–term electricity price time series.

3 Fundamentals

The proposed approach is a combination of two existing techniques. Thus, this section provides the mathematical fundamentals underlying to both LBF (Subsection 3.1) and Q-epiMiner algorithms (Subsection 3.2). A more detailed explanation can be found in [9] and [10], respectively.

3.1 Time Series Forecasting: The LBF Algorithm

The LBF algorithm was initially presented in [9]. Given the hourly prices recorded in the past, up to day d, the forecasting problem aims to predict the 24 hourly prices corresponding to day d+1.

Let $P_i \in R^{24}$ be a vector composed of the 24 hourly energy prices corresponding to a certain day i

$$P_i = [p_1, p_2, \ldots, p_{24}]\,. \tag{1}$$

Let L_i be the label of the prices of the day i obtained as a previous step to the forecasting by using a clustering technique. Let S_W^i the subsequence of labels of the prices of the W consecutive days, from day i backward, as follows:

$$S_W^i = [L_{i-W+1}, L_{i-W+2}, \ldots, L_{i-1}, L_i] \tag{2}$$

where the length of the window, W, is a parameter to be determined.

The LBF algorithm first searches for the subsequences of labels which are exactly equals to S_W^d in the data base, providing the equal subsequences set, ES, defined by the equation,

$$ES = \left\{ \text{set of indexes } j \text{ such that } S_W^j = S_W^d \right\} \tag{3}$$

In case of not finding any subsequence in data base equal to S_W^d, the procedure searches for the subsequences of labels which are exactly equals to S_{W-1}^d. That is, the length of the window composed of the subsequence of labels is decreased.

According to the LBF approach, the 24 hourly prices of day $d+1$ are predicted by averaged the prices of the days succeeding those in ES. That is,

$$P_{d+1} = \frac{1}{\text{size}(ES)} \cdot \sum_{j \in ES} P_{j+1} \tag{4}$$

where size(ES) is the number of elements belonging to the set ES. Afterwards, LBF algorithm outputs need to be de-normalized to generate the desired forecasted values.

When the horizon of prediction is greater than one day, the following tasks have to be carried out. First of all, the real values of the predicted sample are linked to the whole dataset. Second, the clustering process is repeated with the enlarged dataset and, finally, the window size is re-calculated and the prediction step is performed.

3.2 Frequent Episodes in Sequences: The Q-epiMiner

The algorithm introduced in [10] analyzes data events or sequences in order to find episodes. Formally, a sequence is an ordered list of events, where an event is identified by the pair $ev =< date, eventType >$. The ordered occurrence of events is called serial episode and represented by:

$$E = [ev_1, ev_2, ..., ev_n] = [< d_1, t_1 >, < d_2, t_2 >, ..., < d_n, t_n >] \tag{5}$$

where n is the number of events forming an episode.

Thus, the algorithm is able to handle with three time constraints, provided as input data: the minimum time span between two events or gap_{min}, the maximum time span between two events or gap_{max} and the maximum time span between the beginning and the end of an episode or $windowSize$. In this way, an episode has to simultaneously satisfy three constraints: the gap_{min} has to be greater or equal to a given threshold, the gap_{max} has to be lesser or equal to a given threshold and the $windowSize$ has to be lesser or equal to a given threshold. The thresholds are set depending on the requirements of the application.

The rules used in the algorithm are computed where the antecedent is a serial episode and the consequent contains only one event type. Then, a list of the positions in the data sequence L_E of a particular episode is built. Concretely, the list contains the time stamps associated with the events comprised in the episode and they are sorted by increasing values.

The next step consists in evaluating the whole sequence of events and the L_E. From this evaluation, the algorithm provides a set of tuples $< ev, L_{ev} >$, where L_{ev} is the list of locations of the event occurrences and ev is an event of the episode E.

Finally, the standard prefix-based strategy [2] is used for the overall enumeration since it fits well with both episodes extraction and the use of the sorted lists. In other words, an episode is used as a prefix and expanded in order to obtain new episodes.

4 Methodology

This section explains the methodology proposed to improve the forecasting process provided by the LBF algorithm. The discovery of frequent episodes is included in the aforementioned algorithm as a crucial step for outliers detection.

Thus, the proposed methodology is divided into two phases clearly differentiated. First of all, the LBF algorithm is trained with the datasets under study. Second, the predictions with the highest error rates made during the training are analyzed by means of frequent episodes techniques. From this analysis, the days likely to be outliers will be determined and not considered in the prediction process.

4.1 Combining the LBF Algorithm with Frequent Episodes

The value of two parameters have to be determined in the LBF process, K and W. With regard to the number of clusters (K), the proposed approach acts exactly equal to what was proposed in [9] (see Section II.C). However, it is important to remark that the length of the window (W) is slightly different calculated from how it was proposed in the original paper. In practice, W is calculated by means of cross-validation.

Concretely, the $n-$fold cross-validation is used in this work to obtain the optimal value of W. In $n-$fold cross–validation, the original sample is partitioned into n subsets. Of the n subsets, a single subset is retained as the validation data for testing the model, and the remaining $n-1$ subsets are used as training data. The cross-validation process is then repeated n times (the folds), with each of the n subsets used exactly once as the validation data. The n results from the folds are then averaged (even if some authors prefer a combination of them) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

Twelve folds have been created in this work ($n = 12$) for all datasets, where each fold represents a month. Consequently, the training set consists of one year.

The 12−fold cross–validation is then evaluated. The forecasting errors are calculated in every fold by varying the length of W. These monthly errors are denoted by $e_{month}\{W = j\}$ for $j = 1 \ldots W_{max}$, where $W_{max} = 10$ –no longer sequences were found in datasets. Then, the average errors are calculated for each window size as follows,

$$\mathbf{e}_i = \frac{\sum_{i=1}^{n} e_{month}\{W = i\}}{n} \tag{6}$$

where $n = 12$ and $month = \{Jan, \ldots, Dec\}$.

The W selected is the one that minimizes the average error corresponding to the 24 folds (months) evaluated.

$$W = \arg\min\{\mathbf{e}_i\} \text{ with } i = 1, ..., W_{max} \tag{7}$$

This modification has a simple justification. With the former methodology (see Section II.D in [9]), only one test set was evaluated and, consequently, the number of episodes found may be limited and not conclusive. However, with the application of $n-$fold cross-validation, the number of sequences is increased n times providing thus n training sets instead of one.

It is now –just after the training step and before the prediction process– that the discovery of frequent episodes plays an important role. Hence, the apparition of anomalous days is intended to be predicted. Once the $n-$ fold cross–validation is applied, the number of sequences generated can be calculated as:

$$\#S(W, \overline{FL}) = n\overline{FL} - W + 1 \tag{8}$$

where n is the number of folds, $\overline{FL}$ the average fold length and W the length of the sequence (or window) considered.

Note that the maximum number of dissimilar sequences that can be generated is bounded by:

$$N_{max}(K, W) = K^W \tag{9}$$

which will be a number typically much higher than the sequences found in the training sets ($N_{max} >> \#S(W, \overline{FL})$).

The episodes aimed to be found are those which generate a prediction error greater than the average error in the cross–validation process. For this reason, a set of events that satisfies $e_j > \min\{\mathbf{e}_i\}$, for $i = 1, ..., W_{max}$ and $j = 1, ..., n\overline{FL}$, is constructed. This set, CS, gathers all the candidates events to be preceded by an episode precursor of outliers.

Nevertheless, not all these candidates have the same probability to be outliers since the associated errors range from values near to the mean error (these candidates should be finally discarded by the approach) to values significantly high. For this reason, each candidate is co-labeled by using clustering techniques, concretely, the K-means algorithm. The decision on how many clusters have to be created is always an open question and many indices might be used. However,

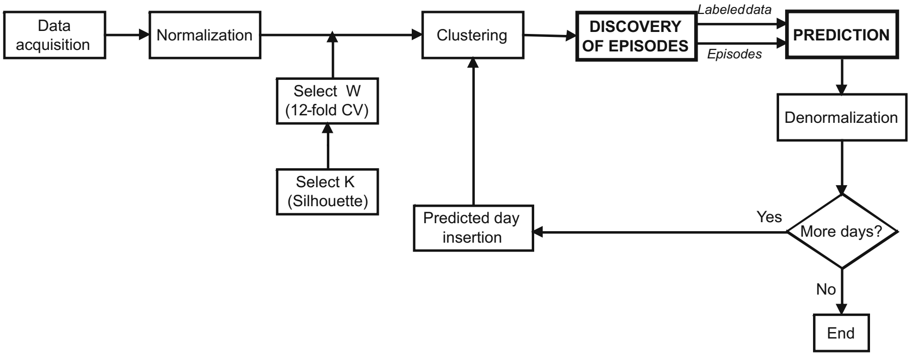

Fig. 2. Illustration of the proposed methodology

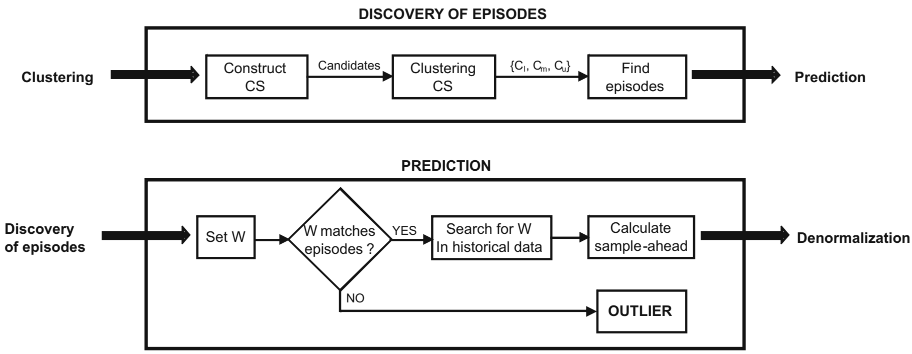

Fig. 3. Detail of both *discovery of episodes* and *prediction* steps

it is worthless to have a large number of clusters and for this reason only three conceptual classes will be created: lower (C_l), central (C_c) and upper (C_u) errors. An a priori reasoning reveals that those candidates belonging to C_u must be more probable to be preceded by sequences generators of outliers than the candidates in C_l or C_c. Results corresponding to each cluster of data will be separately analyzed in Section 5.

The next step consists in computing the episodes (concrete sequences of labels) occurred before the candidates in order to determine the apparition of an outlier. Hence, the approach has to decide which sequences preceding the candidates are the episodes causing errors greater than the expected average error. Then, the sequences that only appear before the candidates are considered frequent episodes and therefore preceding outliers. Fig. 2 illustrates the whole process of prediction when the frequent episodes are included in the LBF algorithm. In addition, the steps corresponding to *discovery of episodes* and *prediction* are further detailed in Fig. 3.

4.2 Parameters of Quality

The parameters used in order to measure the accuracy of the approach are now introduced. Note that in subsequent equations, true positives or TP is the number of candidates that indeed were preceded by episodes that caused errors greater than the average; true negatives or TN is the number of sequences found before a candidate that was properly discarded; false positives or FP is the number of candidates whose preceding episodes were erroneously considered to be causing of errors greater than the average and, finally, false negatives or FN is the number days not considered candidates and eventually preceded by episodes causing errors greater than the average.

According to these definitions, the sensitivity is the probability to detect a frequent episode as precursor of outliers. Its formula is defined as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{10}$$

Other parameter is the specificity which is the ratio of outliers candidates properly discarded by the approach. The mathematical expression is:

$$Specificity = \frac{TN}{TN + FP} \tag{11}$$

The positive predictive value (PPV) is the probability that a detected outlier is indeed a real one. Its formula is:

$$PPV = \frac{TP}{TP + FP} \tag{12}$$

Finally, the negative predictive value (NPV) is the probability that a discarded candidate to be outlier was not indeed a real one. Its formula is:

$$NPV = \frac{TN}{TN + FN} \tag{13}$$

5 Results

In order to prove that the LBF works properly over different datasets, the authors in [9] considered several public electricity prices time series. The new approach is applied on the Spanish electricity price time series (OMEL) [11].

This section is structured as follows. First, the training of the LBF is presented, obtaining thus the adequate values for the parameters K and W. The results provided in this step are, then, analyzed by means of frequent episodes techniques intending to find those patterns in the historical data that perform the worst predictions. From this analysis, some days will become candidates to have an anomalous behavior and, consequently, have a higher error prediction. Finally, the validity of considering these days candidates to be outliers is discussed.

5.1 Discovering Frequent Episodes in Time Series

The forecasting process is applied for the year 2006, with a historical data of a length of one year and with a horizon of prediction of one month. Given this situation, every time a month is forecasted the training set changes. When January 2006 is forecasted, the training set comprises the whole year 2005. However, when February 2006 is forecasted the historical data ranges from February 2005 and January 2006, and so on.

The results of January 2006 are now described since the explanation of the remaining eleven months is analogous. As for the LBF, the number of clusters to be generated as well as the length of the window to be searched for are calculated according to the methodology presented in Section 4.1. Thus, this pair of parameters are equal to: $(K, W) \leftarrow (4, 5)$.

Consequently, the number of sequences generated during the training step is $\#S(W, \overline{FL}) = 361$. However, many sequences appeared repeatedly and the final number of different sequences were 43. As the maximum number of possible sequences is $N_{max}(K, W) = 1024$, the aforementioned number of sequences represent the 4.19% of the potential.

With regard to the Q-epiMiner, the parameters are set to $gap_{max} = 1$, $gap_{min} = 1$ and $windowSize = W$ in order to adapt its application to the particular problem tackled in this work. Also note, that the events are the labels generated during the LBF process, the date is the day associated with such label and the type of event is the curve of prices associated to this day.

The CS can be now constructed. For this purpose, the $\mathbf{e}_i$ from (5) have to be considered since the candidates are those days belonging to the training set that obtained an error greater than $\mathbf{e}_i$. The value of the mean error, calculated according to the methodology in Section 4.1 is $\mathbf{e} = 2.23\%$. From the 365 days comprising the training set, 131 had an error greater than 2.23% so the constructed CS contains 131 candidates.

The candidates have to be classified by means of K-means, with $K = 3$ as discussed in Section 4.1. The obtained cutoff values were 3.78% and 5.69% for dividing classes C_l–C_m and C_m–C_u, respectively. From these cutoffs, the candidates were classified as follows: $98 \in C_l$, $25 \in C_m$ and $8 \in C_u$.

Once the candidates are selected and classified, the sequences that generated them are evaluated. From the candidates in C_l, 7 different sequences were found ($\#S_l = 7$); from the candidates in C_m, 4 ($\#S_m = 4$) and from the candidates in C_u, 2 ($\#S_u = 2$). This fact involves that from the 43 sequences found in the training set, only 13 caused errors greater than the average.

Finally, the number of episodes causing outliers are determined. From the sequences C_l, just one appeared exclusively as a precursor of an outliers. With reference to sequences in C_m, three out of four. And both two sequences in C_u were exclusive.

Table 1 summarizes the results for the twelve months of the year 2006.

Table 1. Training parameters, candidates distribution and episodes found for the year 2006

	Training			Cutoff		Candidates			Sequences	
Month	K	W	**e**	C_l–C_m	C_m–C_u	$C_l(\#S_l)$	$C_m(\#S_m)$	$C_u(\#S_u)$	$\#S(W,\overline{FL})$	N_{max}
January	4	5	2.23%	3.78%	5.69%	98(7)	25(4)	8(2)	362	1024
February	4	5	4.07%	5.21%	6.87%	87(6)	31(7)	5(2)	362	1024
March	4	5	6.30%	7.03%	7.66%	73(5)	16(3)	8(1)	362	1024
April	4	5	2.79%	3.83%	5.01%	103(9)	30(6)	6(3)	362	1024
May	4	5	7.51%	7.97%	9.43%	65(4)	51(6)	10(4)	362	1024
June	6	4	4.02%	5.42%	6.38%	97(6)	38(5)	4(0)	360	1296
July	5	5	4.98%	5.67%	6.13%	180(8)	27(3)	12(5)	361	3125
August	6	4	5.35%	6.20%	6.94%	101(8)	26(5)	9(4)	360	1296
September	6	4	6.24%	7.30%	8.29%	110(8)	25(5)	5(0)	360	1296
October	6	4	6.38%	7.31%	7.88%	108(7)	23(4)	6(1)	360	1296
November	6	4	8.97%	11.68%	13.57%	120(9)	40(6)	6(3)	360	1296
December	5	5	6.51%	7.93%	8.97%	169(10)	38(9)	10(3)	361	3125

5.2 Quantifying the Improvements Achieved

How the prediction is improved by not considering the days pointed by the episodes precursors to outliers found is shown in this subsection. To evaluate the accuracy of the methodology, different criteria may be taken into consideration. However, two parameters –the mean relative error (MRE) and its standard deviation (σ_{MRE})– are used in order to make a comparison with the results in [9].

Table 2 shows the results of the forecasting process performed by the LBF and the results when the episodes causing outliers were discovered and removed from datasets. Note that the approach improves the forecasting in all the datasets considered but for in April. This fact is due to the absence of episodes found when this month was forecasted.

The greater is the average error, the better works this hybrid methodology since outliers are usually involved in high error rates. Equally remarkable is the reduction in the σ_{MRE} from 0.27 to 0.23. Last but not least, a statistical measure of the accuracy of the proposed methodology is provided. The parameters used are the ones described in Section 4.2 and collected in Table 3. Note that all parameters are referred to the whole year 2006, that is, the numbers gather the twelve sets –months– forecasted.

Note that the number of sequences initially considered was 178 ($\sum_{i=1}^{12}\{\#S_{l_i} + \#S_{m_i} + \#S_{u_i}\} = 178$). From these 178 sequences, 150 were sequences that appeared solely in the subset of candidates in which they were found. Consequently, the system considered 150 episodes to be causing of outliers. From all of them, 145 were indeed episodes that preceded a day with a forecasting error greater than the average during the training. The other five did not cause large errors. That is: $TP = 145$ and $FP = 5$. None of the $178 - 150 = 28$ sequences discarded generated predictions with a high error, so: $TN = 28$. Finally, during the forecasting process there appeared 8 sequences which were not initially considered by the model and that eventually were trigger of outliers.

Table 2. Forecasting for the year 2006 in OMEL time series

	LBF		LBF + episodes	
Month	MRE	σ_{MRE}	MRE	σ_{MRE}
January	7.26%	0.25	6.98%	0.21
February	4.93%	0.19	4.28%	0.16
March	5.88%	0.22	5.07%	0.19
April	3.62%	0.18	3.62%	0.18
May	8.11%	0.21	6.95%	0.19
June	3.76%	0.24	3.67%	0.24
July	4.30%	0.23	4.25%	0.23
August	5.37%	0.34	4.66%	0.27
September	6.41%	0.31	6.40%	0.30
October	7.89%	0.29	7.00%	0.22
November	8.30%	0.40	7.12%	0.29
December	8.02%	0.36	7.61%	0.31
Average	6.15%	0.27	5.63%	0.23

Table 3. Statistical analysis of the method

Parameters	Values
TP	145
TN	28
FP	5
FN	8
Sensitivity	94.77%
Specificity	84.85%
PPV	96.67%
NPV	77.77%

6 Conclusions

The combination of two techniques has been used in order to forecast time series. The initial approach –the LBF– was based on finding similar patterns in time series. However, its application to any kind of time series revealed that there were some samples that cannot be properly forecasted since they showed a stochastic behavior.

The use of frequent episodes techniques is thus applied, not for providing an accurate prediction for these samples, but for indicating that it is reasonably probable that an outlier occurs. The method has been successfully tested on twelve sets of the Spanish electricity price time series.

Future work is directed towards finding not only the days that are going to present an anomalous behavior, but the days whose prediction is going to be specially accurate. In addition, a relaxation for the rule that decides if a given sequence is an episode or not is intended to be created.

Acknowledgments

The financial support from the Spanish Ministry of Science and Technology, project TIN2007-68084-C-02, and from the Junta de Andalucía, project P07-TIC-02611, is acknowledged. The authors also want to acknowledge to Ph. D. Rigotti the source code provided for the statistical analysis of frequent episodes used in this work.

References

1. Basu, S., Meckesheimer, M.: Automatic outlier detection for time series: an application to sensor data. Knowledge and Information Systems 11(2), 137–154 (2007)
2. Esparza, J., Heljanko, K.: Unfoldings: A Partial-Order Approach to Model Checking. Springer, Heidelberg (2008)
3. García-Martos, C., Rodríguez, J., Sánchez, M.J.: Mixed models for short-run forecasting of electricity prices: application for the Spanish market. IEEE Transactions on Power Systems 22(2), 544–552 (2007)
4. Herui, C., Li, Y.: Short-term electricity price forecast based on improved fractal theory. In: Prooceedings of the eighth IEEE International Conference on Computer Engineering and Technology, pp. 347–351 (2009)
5. Jabłońska, M.: Analysis of outliers in electricity spot prices with example of New England and New Zealand markets. PhD thesis, Lappeenranta University, Finland (2008)
6. Li, G., Liu, C.C., Mattson, C., Lawarrée, J.: Day-ahead electricity price forecasting in a grid environment. IEEE Transactions on Power Systems 22(1), 266–274 (2007)
7. Louni, H.: Outlier detection in ARMA models. Journal of Time Series Analysis 29(6), 1057–1065 (2008)
8. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery 1, 259–289 (1997)
9. Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C., Ruiz, J.S.A.: LBF: A labeled-based forecasting algorithm and its application to electricity price time series. In: Prooceedings of the eighth IEEE International Conference on Data Mining, pp. 453–461 (2008)
10. Nanni, M., Rigotti, C.: Extracting trees of quantitative serial episodes. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 170–188. Springer, Heidelberg (2007)
11. Spanish Electricity Price Market Operator, `http://www.omel.es`
12. Pino, R., Parreno, J., Gómez, A., Priore, P.: Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. Engineering Applications of Artificial Intelligence 21(1), 53–62 (2008)
13. Troncoso, A., Riquelme, J.C., Riquelme, J.M., Martínez, J.L., Gómez, A.: Electricity market price forecasting based on weighted nearest neighbours techniques. IEEE Transactions on Power Systems 22(3), 1294–1301 (2007)
14. Zhao, J.H., Dong, Z.Y., Li, X., Wong, K.P.: A framework for electricity price spike analysis with advanced data mining methods. IEEE Transactions on Power Systems 22(1), 376–385 (2007)