A Method Based on RBF-DDA Neural Networks for Improving Novelty Detection in Time Series

A. L. I. Oliveira and **F. B. L. Neto**
Polytechnic School, Pernambuco University
Rua Benfica, 455, Madalena, Recife – PE, Brazil
ZIP: 50.750-410, e-mail: {alio,fbln}@cin.ufpe.br

S. R. L. Meira
Center of Informatics, Federal University of Pernambuco
P.O. Box 7851, Cid. Universitaria, Recife – PE, Brazil
ZIP: 50.732-970, e-mail: srlm@cin.ufpe.br

Abstract

Novelty detection in time series is an important problem with application in different domains such as machine failure detection, fraud detection and auditing. An approach to this problem uses time series forecasting by neural networks. However, time series forecasting is a difficult problem, thus, the use of this technique for time series novelty detection is sometimes criticized. Alternatively, a number of different classification-based techniques have been recently proposed for this problem. The idea of these methods is to learn to classify time series windows as *normal* or *novelty*. Unfortunately, in many cases of interest there are only normal data available for training. Several of the classification-based techniques tackle this problem by adding random *negative samples* to the training set. In some cases the performance of the novelty detection method depends on the number of random negative samples added and selection of this number can be a problem. In this work, we present a method for novelty detection in time series based on RBF neural networks. The proposed method does not need negative samples and is based on the dynamic decay adjustment (DDA) algorithm for RBF networks training. We have carried out a number of experiments using four real-world time series, whose results have shown that the performance of the method proposed in this work is much better than that of a method that needs negative samples.

Introduction

Novelty detection – the process of finding novel patterns in data sets – is very important in several domains such as computer vision, machine fault detection, network security and fraud detection (Singh & Markou 2003; González, Dasgupta, & Kozma 2002; Gonzalez & Dasgupta 2002). A novelty detection system can be regarded as a classifier with two possible outcomes, one for *normal* and the other for *novelty* patterns. However, in most cases, there is only normal data available to train the classifier (Singh & Markou 2003; Gonzalez & Dasgupta 2002). Hence, novelty detection systems must be properly designed to overcome this problem.

The behavior of many systems can be modeled by time series. Thus, analysis of this kind of data is very important. Recently, the problem of detecting novelties in time series has received great attention, with a number of different techniques being proposed and studied, including techniques based on time series forecasting with neural networks (Koskivaara 2000; Oliveira *et al.* 2003), artificial immune system (Gonzalez & Dasgupta 2002), wavelets (Shahabi, Tian, & Zhao 2000) and Markov models (Keogh, Lonardi, & Chiu 2002). These techniques have been applied in areas such as machine failure detection (Gonzalez & Dasgupta 2002) and auditing (Koskivaara 2000; Oliveira *et al.* 2003).

Forecasting-based time series novelty detection has been criticized because of the not so good performance (Keogh, Lonardi, & Chiu 2002; González, Dasgupta, & Kozma 2002). Alternatively, a number of classification-based approaches have been recently proposed for novelty detection in time series (Gonzalez & Dasgupta 2002; Shahabi, Tian, & Zhao 2000; Keogh, Lonardi, & Chiu 2002; Oliveira, Neto, & Meira 2003). Some of these methods use artificially generated *negative samples* to represent novelty. These samples are added to the training set. After training, the classifier will be able to classify time series windows as *normal* or *novelty*. The problem with this approach is that the number of negative samples added has an important influence on classification performance (Oliveira, Neto, & Meira 2003).

In this work we propose a novel method for novelty detection in time series that does not need negative samples. The method uses RBF neural networks to classify time series windows as normal or novelty. It is based on the dynamic decay adjustment algorithm, originally proposed for training RBF networks for classification (Berthold & Diamond 1995). We present results of experiments with four real-world time series whose objective was to compare the proposed method with methods based on added negative samples.

The Proposed Method

The novelty detection method proposed in this paper works by classifying time series windows as normal or novelty. The system requires fixed length windows, with window size w. A window is formed by w consecutive datapoints extracted from the time series under analysis. The first training pattern will have the first w datapoints from the time series as its attributes values. To obtain the second pattern we start with the second datapoint and use the next w dat-

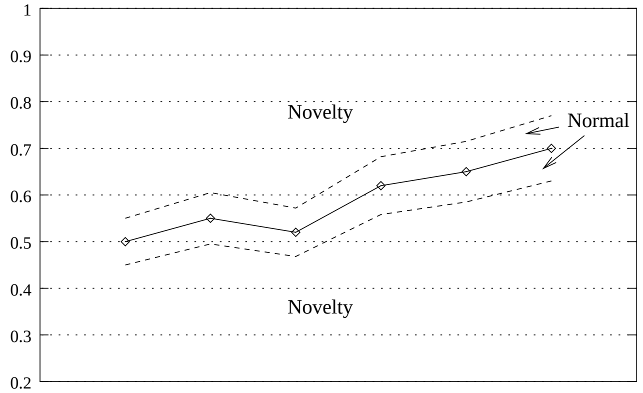


Figure 1: Definition of normal and novelty regions.

apoints. The remaining patterns are obtained by sliding the window by one and taking the next w datapoints. So, if we have a time series with l datapoints and use window size w, we will have $l - w + 1$ patterns. These patterns will later be separated to obtain training and test sets.

Given a window from the time series, the idea is to define an *envelope* around it as shown in figure 1. Any time series window with all values inside the envelope is considered normal. Windows with points outside the envelope are considered novelty. We use a threshold p_1 to define the envelope. Normal patterns are defined by establishing the maximum percent deviation p_1 above and below each datapoint of a given original pattern.

Our method uses radial basis functions (RBFs) in order to define the region of normality shown in figure 1. In this work we use radial Gaussians, the commonest type of RBF. A RBF unit works by computing the Euclidian distance to an individual reference vector $\overrightarrow{r_i}$, giving an output

$$R_i(\overrightarrow{x}) = \exp\left(-\frac{||\overrightarrow{x} - \overrightarrow{r_i}||^2}{\sigma_i^2}\right) \qquad (1)$$

where $\overrightarrow{x}$ is the input vector and $||\overrightarrow{x} - \overrightarrow{r_i}||$ is the Euclidian distance between the input vector $\overrightarrow{x}$ and the Gaussian center $\overrightarrow{r_i}$. The Gaussian center will correspond to a normal time series window. The standard deviation σ_i should be adjusted to make the RBF unit produce a low output for values outside the envelope. We use the dynamic decay adjustment – a constructive training algorithm for RBF networks (Berthold & Diamond 1995) – to adjust Gaussian standard deviations properly.

The DDA algorithm is a constructive algorithm used to build and train RBF networks for classification tasks (Berthold & Diamond 1995). It does not need a validation set and so, all training data can be more effectively used for training. RBF-DDA has often achieved classification accuracy comparable to multi-layer perceptron networks (MLPs) but training is significantly faster. RBF-DDA neural networks have a single hidden layer with RBF units fully connected to the input layer, as shown in figures 2 and 3. The number of units in the input layer depends on the dimensionality of the problem. Units in the hidden layer are added by the training algorithm as needed. The number of units in the output layer corresponds to the number of classes in the problem. The centers and standard deviations of RBF units are also determined by the training algorithm. The winner-takes-all rule is used for classification.

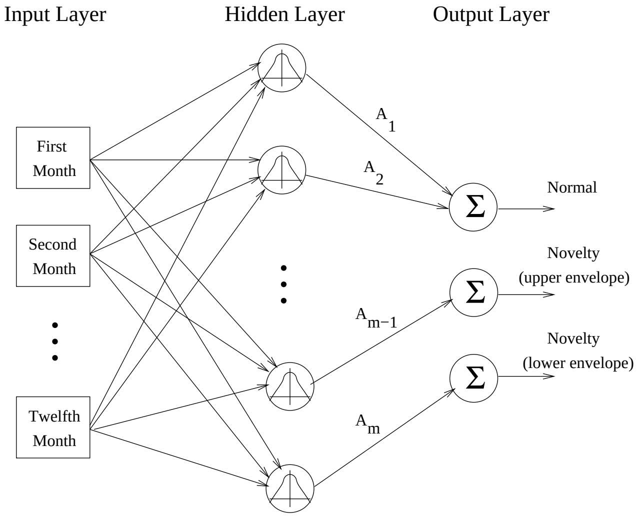


Figure 2: RBF network with three output units for classification-based novelty detection.

In our novelty detection method, we generate, for each original time series window pattern, two additional patterns. The first corresponds to the upper envelope and the second, to the lower one. That is, the first additional pattern is obtained from the original by adding the percent deviation p_1 that defines the upper envelope. The second additional pattern is obtained by subtracting p_1 from the original pattern. Using this approach, an augmented training set three times larger than the original training set is generated. The upper and lower envelope patterns are added to the training set in order to help the DDA algorithm adjust the standard deviation σ_i of RBF units associated with normal patterns properly. In this way, after training, standard deviations are adjusted in a way that patterns in the novelty region produces low values for the normal output and high values for the novelty output.

We use the RBF architectures shown in figures 2 and 3. In the first case there are three outputs. The first output is associated with normal patterns, that is, original time series windows. The second and the third outputs are associated with upper and lower envelope patterns, respectively. The second architecture has only two outputs, one for normal patterns and the other for novelty patterns. In the training phase the novelty output will be associated with both upper and lower envelop patterns from the augmented training set.

After training, the network classifies new patterns as follows. If all outputs have values below 10^{-6}, the pattern is classified as novelty. If this is not the case, the winner-takes-all rule is used to classify patterns as either normal or novelty.

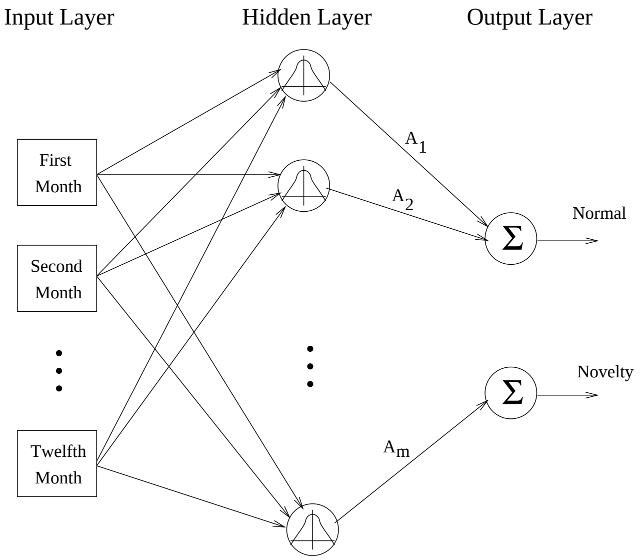


Figure 3: RBF network with two output units for classification-based novelty detection.

The Method Based on Negative Samples

The method proposed in this paper will be compared to another method recently proposed (Oliveira, Neto, & Meira 2003). This method is based on negative samples and on the use of RBF-DDA networks for classification of time series windows as normal or novelty. The method has achieved good performance, however, it has been shown that performance depends on the number of negative samples used in the training phase. This method is also based on the assumption that the time series available represent the normal behavior and so the training set will have only normal patterns. Hence, in order to train a classifier for the novelty detection task, random negative samples (*novelty random patterns*) are generated from the original patterns and added to the training set. These patterns are time series windows with datapoints in the novelty regions shown in figure 1.

This method requires an adequate number of random patterns for each window in the series, in order to represent adequately the novelty space. A number of *normal random patterns* are also added to training set. Normal random patterns are generated from original patterns and have all datapoints are inside the envelope (figure 1). In order to improve classification performance, random patterns should be added in a way that the resulting data set have equal numbers of normal and novelty patterns (Haykin 1998). The training set with the original patterns and the random normal and random novelty patterns is called augmented training set.

In many problems of interest, such as auditing, where novelty is related to possibility of fraud, we are mainly interested in testing network performance in detection of patterns whose deviation from normality is not too big. Thus, a second threshold p_2 is defined in the method in order to limit the novelty regions. For example, in the experiments presented below, we use $p_1 = 0.1$ and $p_2 = 0.5$, meaning that patterns whose attributes are at most 10% from the normal pattern are considered normal and patterns whose attributes deviates from a normal pattern from 10% to 50% are considered novelty or fraudulent patterns. After generating the augmented training set, a classifier is trained to discriminate normal and novelty windows in the time series. In this work we use the same classifier used in (Oliveira, Neto, & Meira 2003), that is, an RBF-DDA neural network whose architecture is shown in figure 3.

Experiments

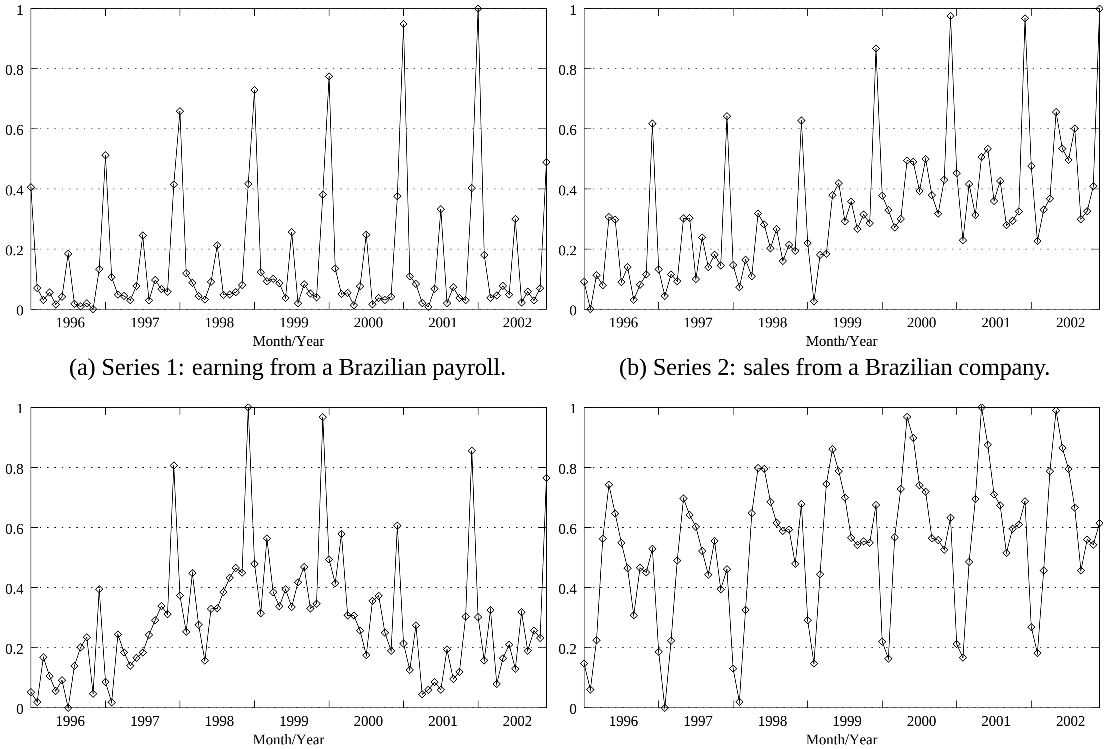
We have carried out some experiments using real-world time series in order to compare the performance of the method proposed in this work with the negative samples method. The time series are depicted in figure 4. All have 84 values corresponding to the months from January 1996 to December 2002. The first series was extracted from a real payroll. The remaining series are sales time series with values in the same period of the first series. Series 3 and 4 can be obtained at the URL http://www.census.gov/mrts/www/mrts.html. Each series had their values normalized between 0 and 1.

It is clear that these series are non-stationary. Hence, in order to use a focused TLFN (*Time Lagged Feedforward Networks*), such as a RBF network, it is important to preprocess the time series in order to work with their stationary versions (Haykin 1998). We have used the classic technique of differencing to obtain stationary versions of the time series (Chatfield 1989). For each original time series $\{x_1, \ldots, x_N\}$ a differenced time series $\{y_2, \ldots, y_N\}$ was formed by $y_t = x_t - x_{t-1}$. Note that the differenced time series does not have the first datapoint. In fact, it has been shown in previous works that differencing the time series has a great influence on classification-based time series novelty detection (Oliveira, Neto, & Meira 2003). Thus, in this work we consider only differenced versions of time series.

We have used a window size $w = 12$. This is a natural choice for the series analyzed, because they are seasonal series with period 12 months. However, for other kinds of series, a careful selection of w is important. The patterns are generated from each time series according to the procedure previously described. For the differenced versions of the time series considered in these experiments we will have 72 patterns with 12 attributes each. We have used the last 12 patterns as the test set and the remaining patterns as training set. It is important to emphasize that normalization is carried out only after the generation of the random patterns.

The method proposed in this work is trained with training sets augmented by a factor of three. This is because we need to add only two additional patterns per original pattern in order to represent the envelope, as described previously. The additional patterns are generated considering an envelope created with threshold $p_1 = 0.1$. Thus, we will have augmented training sets with 180 patterns for the time series considered here.

For the negative samples approach, the normal and novelty regions were defined using thresholds $p_1 = 0.1$ and $p_2 = 0.5$. The generation of augmented training sets for this approach works by the addition of $n-1$ normal random



(a) Series 1: earning from a Brazilian payroll.

(b) Series 2: sales from a Brazilian company.

(c) Series 3: USA computer and software stores sales.

(d) Series 4: USA hardware stores sales.

Figure 4: Time series used in the experiments.

patterns and n random novelty patterns for each original pattern. We have used $n = 10$ and $n = 20$ in the experiments. With this, our training sets increase by factor of 20 and 40, respectively. Thus, for the series considered, experiments were carried out with augmented training sets having 1200 or 2400 patterns.

The series shown in figure 4 are supposed to represent the normal behavior. Nevertheless, we want to study the performance of the methods on novelty detection on test sets. Hence, we generate augmented test sets from the original test sets. For each series we have worked with two different augment test set sizes. The first is generated from the original test set by adding 9 normal random patterns and 10 random novelty patterns for each original pattern in the test set. In this way, the augmented test set gets increased by a factor of 20. The second alternative increases the test set by a factor of 200 by adding 99 normal random patterns and 100 random novelty patterns for each original pattern. Thus, in the former case we will have 240 patterns in the augmented test sets and in the later case we will have 2400 patterns.

For the method proposed in this work, we train the network for each series only one time, because RBF-DDA does not depend on weights initialization. We have tested each trained network ten times in order to take into account the variability of the random patterns added to form the test sets. On the other hand, for the negative samples approach we have trained each network with ten different versions of the training set generated from different seeds, to take into account the variability of the random patterns added to form them. The network was also tested ten times, for the same reason.

Results

Table 1 presents results obtained after training the networks for series 1, 2, 3, and 4. It contains the mean number of epochs used in training, the mean number of hidden units in the resulting network and the mean and standard deviation of the classification error on the test set. Table 2 presents the mean and standard deviation for the false alarm rate and the undetected novelty rate on the test set. In both tables results are presented for the proposed method using networks with three outputs; for the proposed method using networks with two outputs and for the negative samples method. A false alarms happens when the network classifies a normal pattern as novelty. An undetected novelty happens when a novelty pattern is misclassified. The negative samples method uses training sets with 1200 patterns while the method proposed in this work has 180 patterns on training sets. Tables 1 and 2 present the mean and standard deviation across ten executions. For each time series, we have used test sets with 240 and 2400 patterns in order to study the influence of test size on the performance of the methods.

The results presented in table 1 show that the proposed method produces classification error smaller than the negative samples approach for all time series considered. This is true for both architectures used in conjunction with the

Method	Epochs	Hid.	Class. error	
		units	mean	s.dev
Time series 1, test set with 240 patterns				
Proposed (3 out.)	3	150	1.17%	0.58%
Proposed (2 out.)	3	149	1.21%	0.57%
Neg. samples	4	623.2	11.00%	2.86%
Time series 1, test set with 2400 patterns				
Proposed (3 out.)	3	150	1.53%	0.18%
Proposed (2 out.)	3	149	1.56%	0.19%
Neg. samples	4	623.2	17.67%	2.86%
Time series 2, test set with 240 patterns				
Proposed (3 out.)	3	171	5.33%	1.48%
Proposed (2 out.)	3	170	5.50%	1.70%
Neg. samples	4	621.7	6.50%	1.86%
Time series 2, test set with 2400 patterns				
Proposed (3 out.)	3	171	6.22%	0.39%
Proposed (2 out.)	3	170	6.36%	0.44%
Neg. samples	4	621.7	12.78%	1.43%
Time series 3, test set with 240 patterns				
Proposed (3 out.)	3	181	2.92%	1.11%
Proposed (2 out.)	3	180	3.17%	1.23%
Neg. samples	4	644.3	8.79%	1.93%
Time series 3, test set with 2400 patterns				
Proposed (3 out.)	3	181	4.30%	0.43%
Proposed (2 out.)	3	180	4.61%	0.46%
Neg. samples	4	644.3	10.87%	0.87%
Time series 4, test set with 240 patterns				
Proposed (3 out.)	2	181	3.79%	1.08%
Proposed (2 out.)	2	180	3.88%	1.09%
Neg. samples	4	644.4	7.13%	1.32%
Time series 4, test set with 2400 patterns				
Proposed (3 out.)	2	181	3.98%	0.33%
Proposed (2 out.)	2	180	4.05%	0.33%
Neg. samples	4	644.4	14.10%	1.69%

Table 1: Classification performance of the novelty detection methods on test sets for each time series

Method	False alarm		Undetec. novelty	
	mean	s.dev	mean	s.dev
Time series 1, test set with 240 patterns				
Proposed (3 out.)	0.75%	0.47%	0.42%	0.44%
Proposed (2 out.)	0.79%	0.50%	0.42%	0.44%
Neg. samples	0.00%	0.00%	11.00%	2.86%
Time series 1, test set with 2400 patterns				
Proposed (3 out.)	0.95%	0.19%	0.58%	0.06%
Proposed (2 out.)	0.98%	0.19%	0.58%	0.06%
Neg. samples	0.00%	0.00%	17.67%	2.86%
Time series 2, test set with 240 patterns				
Proposed (3 out.)	4.58%	1.08%	0.75%	0.61%
Proposed (2 out.)	4.75%	1.27%	0.75%	0.61%
Neg. samples	0.00%	0.00%	6.50%	1.86%
Time series 2, test set with 2400 patterns				
Proposed (3 out.)	5.20%	0.42%	1.02%	0.19%
Proposed (2 out.)	5.34%	0.46%	1.02%	0.19%
Neg. samples	0.00%	0.00%	12.78%	1.43%
Time series 3, test set with 240 patterns				
Proposed (3 out.)	2.46%	0.89%	0.46%	0.36%
Proposed (2 out.)	2.71%	1.02%	0.46%	0.36%
Neg. samples	0.00%	0.00%	8.79%	1.93%
Time series 3, test set with 2400 patterns				
Proposed (3 out.)	3.48%	0.40%	0.82%	0.17%
Proposed (2 out.)	3.79%	0.42%	0.82%	0.17%
Neg. samples	0.00%	0.00%	10.87%	0.87%
Time series 4, test set with 240 patterns				
Proposed (3 out.)	2.29%	0.88%	1.50%	0.71%
Proposed (2 out.)	2.38%	0.90%	1.50%	0.71%
Neg. samples	0.00%	0.00%	7.13%	1.32%
Time series 4, test set with 2400 patterns				
Proposed (3 out.)	2.37%	0.28%	1.61%	0.23%
Proposed (2 out.)	2.44%	0.27%	1.61%	0.23%
Neg. samples	0.00%	0.00%	14.10%	1.69%

Table 2: False alarm and undetected novelty rates on test sets for each time series

proposed method (figures 2 and 3). The performance gain is variable across the series, however, it can be very high, for example, for series 1 with 2400 patterns on test sets, the classification error decreases from 17.67% (for the negative samples approach) to 1.53% (for the proposed method with three outputs). These results also show that the RBF architecture with three outputs produces slightly better results than the two outputs architecture. It can also be noted that the negative samples approach is much more sensible to the increase in test set size. For this approach, when test sets increase from 240 to 2400 patterns the classification error increases by a factor of almost two for some series.

The method proposed in this work and the negative samples method behave quite differently with respect to false alarm and undetected novelty rates, as shown in table 2. The negative samples method always produces 0% false alarm rate. All the misclassifications produced by this method happen with novelty patterns. In contrast, the proposed method produces more false alarms than undetected novelties.

We summarize the results from table 1 in table 3. This table presents the mean classification errors across the four time series for each method. It clearly shows that the proposed method performs much better and that its performance is less dependent on test set size. The relative increase in classification error when the test sets increases from 240 to 2400 patterns are: 21.5% for the proposed method with three outputs; 20.6% for the proposed method with two outputs; and 65.8% for the negative samples method.

Finally, we performed additional experiments with the negative samples approach, this time increasing the training set size. Recall that results presented in tables 1 and 2 where obtained with augmented training sets with 1200 patterns. Table 4 presents the mean classification errors on test sets for the four time series using augmented training sets with 2400 patterns. The test sets also have 2400 patterns. The results show that the classification errors on test sets decreases when the training sets get increased. Even so, the method proposed in this work still produces much better results as can be seen in tables 1 and 3. It has the additional advantage of using a fixed number of patterns in the training set,

Method	Mean Class. Error
Test sets with 240 patterns	
Proposed (3 outputs)	3.30%
Proposed (2 outputs)	3.44%
Negative samples	8.36%
Test sets with 2400 patterns	
Proposed (3 outputs)	4.01%
Proposed (2 outputs)	4.15%
Negative samples	13.86%

Table 3: Mean classification errors across the four time series for each method.

because it does not depend on negative samples.

Series	Class. Error	
	mean	s.dev
Time series 1	8.71%	1.76%
Time series 2	8.84%	1.36%
Time series 3	6.91%	1.02%
Time series 4	8.74%	2.09%
Mean	8.3%	0.93%

Table 4: Mean classification errors for the negative samples method with 2400 patterns on both training and test sets.

Conclusions

In this work we have presented a novel method for novelty detection in time series. This is an important problem and a number of techniques have been developed recently for it. Several of these techniques rely on negative samples in order to represent novelties in the training phase. On the other hand, our method does not uses negative samples. It is based on the DDA, a constructive algorithm for training RBF networks for classification tasks (Berthold & Diamond 1995). In this paper, the proposed method was compared to another method based on negative samples (Oliveira, Neto, & Meira 2003). The methods were compared experimentally using four real-world non-stationary time series and the experiments have shown that proposed method achieves much better performance. Experiments have shown that the performance of the negative samples method depends on the number of negative samples added to training sets. In contrast, the method proposed in this work has the additional advantage of using fixed training set size. For each time series window in the training set only two additional patterns are generated and added to the training set.

The methods have been compared using seasonal non-stationary time series which appear in many important problems, such as auditing (Koskivaara 2000; Oliveira *et al.* 2003). However, we are aware of the importance of assessing the performance of the method on other kinds of time series. Our future works include studying the impact of the window size on the method performance and adapting the method by using its ideas in conjunction with TDRBF, a neural network with more powerful temporal processing abilities that is also trained with the DDA algorithm (Berthold 1994). This method could be used to classify directly non-stationary time series. Finally, we include in our future works the application of the method proposed here and future extensions of it on real-world auditing problems, such as accountancy auditing (Koskivaara 2000) and payroll auditing (Oliveira *et al.* 2003).

References

Berthold, M. R., and Diamond, J. 1995. Boosting the performance of RBF networks with dynamic decay adjustment. In et al, G. T., ed., *Advances in Neural Information Processing*, volume 7.

Berthold, M. R. 1994. A time delay radial basis function network for phoneme recognition. In *Proc. of the IEEE International Conference on Neural Networks*, volume 7, 4470–4473.

Chatfield, C. 1989. *The Analysis of Time Series – An Introduction*. Chapman & Hall, fourth edition.

Gonzalez, F., and Dasgupta, D. 2002. Neuro-immune and self-organizing map approaches to anomaly detection: A comparison. In *Proc. of the 1st International Conference on Artificial Immune Systems*.

González, F.; Dasgupta, D.; and Kozma, R. 2002. Combining negative selection and classification techniques for anomaly detection. In *Proc. of IEEE Congress on Evolutionary Computation*, 705–710.

Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition.

Keogh, E.; Lonardi, S.; and Chiu, W. 2002. Finding surprising patterns in a time series database in linear time and space. In *Proc. ACM Knowledge Discovery and Data Mining - SIGKDD'02*, 550–556.

Koskivaara, E. 2000. Artificial neural network models for predicting patterns in auditing monthly balances. *Journal of Operational Research Society* 51(9):1060–1069.

Oliveira, A. L. I.; Azevedo, G.; Barros, A.; and Santos, A. L. M. 2003. A neural network based system for payroll audit support (in portuguese). In *Proceeding of the IV Brazilian National Artificial Intelligence Meeting*.

Oliveira, A. L. I.; Neto, F. B. L.; and Meira, S. R. L. 2003. Novelty detection for short time series with neural networks. In Abraham, A.; Köppen, M.; and Franke, K., eds., *Design and Application of Hybrid Intelligent Systems*, volume 104 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

Shahabi, C.; Tian, X.; and Zhao, W. 2000. TSA-tree: A wavelet-based approach to improve the efficiency of multi-level surprise and trend queries on time-series data. In *Proc. of 12th International Conference on Scientific and Statistical Database Management*.

Singh, S., and Markou, M. 2003. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering* 15.