Novelty Detection in Time Series Through Self-Organizing Networks: An Empirical Evaluation of Two Different Paradigms

Leonardo Aguayo
Nokia Institute of Development and Technology (INDT)
SCS Bl. 1, Camargo Corrêa Bld., 6th floor, CEP 70397-900, Brasília, Brazil
leonardo.aguayo@indt.org.br

Guilherme A. Barreto
Department of Teleinformatics Engineering, Federal University of Ceará
Av. Mister Hull, S/N, Center of Technology, Campus of Pici
CP 6005, CEP 60455-970, Fortaleza, Ceará, Brazil
guilherme@deti.ufc.br

Abstract

This paper addresses the issue of novelty or anomaly detection in time series data. The problem may be interpreted as a spatio-temporal classification procedure where current time series observation is labeled as normal or novel/abnormal according to a decision rule. In this work, the construction of the decision rules is formulated by means of two different self-organizing neural network (SONN) paradigms: one builds decision thresholds from quantization errors and the other one from prediction errors. Simulations with synthetic and real-world data show the feasibility of the two approaches.

1. Introduction

Novelty - or anomaly - detection is concerned with the difficult problem of finding samples which appear to be inconsistent with a previously modeled subset of data. Typical application areas requiring novelty detection procedures are equipment fault detection and diagnosis, fraud detection, database cleaning, computer network security, among others. In general, such novelty detection strategies are devised by means of static pattern recognition techniques, where the time dimension plays no role at all in the decision rule.

However, several real-world applications provide data in a time-ordered fashion, usually in the form of successive measurements on the magnitude of one or several variables of interest, giving rise to time series data. In financial market, stock time series may present patterns that can guide an investor in his/her investment decisions in short- or long-term horizons; biomedical measurements such as ECG or EEG are a valuable source of information for reliable diagnosis; measurements from chemical or mechanical processes are used to control complex manufacturing systems. Indeed, classic authors [4] elect the "analysis of effects of unusual intervention events to a system" as one a relevant practical problem to be addressed by time-series analysis.

Novelty detection in time series data is particularly challenging due to the usual presence of specific features, such as trend and seasonality, that mask the character of novelty that may be present in data. Non-stationary processes, such as regime-switching time series, also impose additional limitations on time series modeling. Furthermore, some types of time series may have have relatively few samples, restricting the amount of data available to extract information about its underlying behavior. Finally, time-critical applications, such as fault detection and surveillance, require online detection of anomalies/novelties.

Traditional approaches to detect novelty, such as statistical parametric modeling and hypothesis testing [12] can be successfully used to model static patterns, as these techniques assume some degree of stationarity of the data. Linear stationary processes can be handled by standard Box-Jenkins ARMA time series models but nonlinear or non-stationary dynamic patterns - such as chaotic or regime-switching time series - require a more powerful approach in terms of learning and computational capabilities.

At this point the use of artificial neural networks (ANNs) have shown to be useful due to their capability to act as general purpose nonlinear system identifier, generalizing the acquired knowledge to unknown data. Most of the ANN-based methods rely on supervised ANN models, such as

1522-4899/08 $25.00 © 2008 IEEE
DOI 10.1109/SBRN.2008.21

Multi-Layer Perceptron (MLP) and Radial Basis Functions (RBF) architectures [13, 5].

A major limitation of such models is their inability to handle training data with unbalanced class sample sizes. Usually, normal data samples abounds, while anomalous data samples may not be always available or may be costly to collect. A plausible solution relies on building a model for the normal data samples only. Any incoming data sample that deviates from this model is considered anomalous (or novel). This approach is usually implemented by means of vector quantization algorithms (e.g. K-means) so that classification of an incoming data sample as normal/abnormal is based on the magnitude of the quantization error produced by that sample.

In recent years, it has been observed an increasing number of applications of neural network based vector quantization algorithms - in particular the Self-Organizing Map (SOM) [8] - to novelty detection tasks [15, 3, 10, 16], most of them dealing with static (spatial) data only. However, since the early 1990's, the SOM algorithm itself and temporal variants of it have been proposed with the aim of performing clustering (or vector quantization) on time series data (see [2] for a review).

When dealing with time series data, SOM-based approaches usually converts the time series into a non-temporal representation (e.g. spectral features computed through Fourier transform) and use it as input to the SOM [17]. It is also possible to use tapped delay lines at the input of the SOM, again converting the time series into a spatial representation [6].

In this paper, however, we are also interested in evaluate the performance of a different self-organizing neural network approach, which does not use the SOM algorithm as a vector quantization algorithm, but rather it provides a multiple (local) model formulation, where a bank of several models are simultaneously fitted to the input time series in other to find the best estimation (prediction) of the current time series observation. If the best predictor provides a too high prediction error, then a novelty may be occurring. For this purpose, the multiple model strategy is implemented through the Operator Map (OPM) network [9], a generalization of the SOM network in which the usual static prototype-based neurons are replaced with dynamic time series models, such as the linear autoregressive model or the Kalman filter.

The remainder of the document is divided as follows. In Section 2 we briefly describe the self-organizing algorithms used in this work to perform novelty detection in time series. Section 3 presents in detail the methodology based on the analysis of both quantization and prediction errors; Section 5 contains the numerical results and comments on the performance of all the simulated algorithms. Finally, Section 4 resumes the key points, conclusions and future work.

2 Vector Quantization of Time Series Data with Self-Organizing Networks

There are many approaches to time series clustering or vector quantization [11], but we limit the scope of our description to prototype-based algorithms. In what concerns the task, we assume that the algorithms are trained on-line as the data is collected. The input vectors are built through a fixed-length window, sliding over the time series of interest. Thus, at time step n, the input vector is given by

$$\mathbf{x}_n^+ = [x_n \; x_{n-1} \; \cdots \; x_{n-p+1}]^T, \tag{1}$$

where $p \geq 1$ is the memory-depth parameter. The superscript T denotes the transpose of a vector. Weight updating is allowed only during a fixed number of steps, T_{max}. Once the network is trained, decision thresholds are computed based on either the quantization or prediction errors - the former approach is used when the SOM algorithm is applied, and the latter when the OPM network is used.

2.1 The SOM algorithm

SOM training is carried out using the vector $\mathbf{x}_n^+$ as input. Thus, the winning neuron, i_n^*, is given by

$$i_n^* = \arg\min_{\forall i} \|\mathbf{x}_n^+ - \mathbf{w}_n^i\|, \quad i = 1, \ldots, Q, \tag{2}$$

where $\mathbf{w}_n^i$ are the weights of the neuron i, Q is the number of neurons (see Figure 1) and n denotes the current iteration of the algorithm. Accordingly, the weight vectors are updated by the following learning rule:

$$\mathbf{w}_{n+1}^i = \mathbf{w}_n^i + \eta_n \mathcal{H}_n(i^*, i) \left(\mathbf{x}_n^+ - \mathbf{w}_n^i\right), \tag{3}$$

where $\mathcal{H}_n(i^*, i)$ is a gaussian function which control the degree of change imposed to the weight vectors of those neurons in the neighborhood of the winning neuron:

$$\mathcal{H}_n(i^*, i) = \exp\left(-\frac{\|\mathbf{r}_n^i - \mathbf{r}_n^{i^*}\|^2}{\sigma_n^2}\right), \tag{4}$$

where σ_n defines the radius of the neighborhood function at iteration n, and $\mathbf{r}_n^i$ and $\mathbf{r}_n^{i^*}$ are the respective coordinates of neurons i and i^* at the output array. The learning rate, $0 < \eta_n < 1$, should decay in time to guarantee convergence of the weight vectors to stable states. In this paper, we use $\eta_n = \eta_0 \left(\eta_T/\eta_0\right)^{t/T_{max}}$, where η_0 is the initial value of η, and η_T is its final value after T_{max} training iterations. The variable σ_n should decay in time in a similar fashion.

2.2 The Operator Map Model

Neurons in the OPM network are regarded as mathematical *operators*, denoted generically by $\mathcal{G}(\cdot)$, representing a non-specific filtering operation over temporal patterns. Such operators usually contain adjustable parameters,

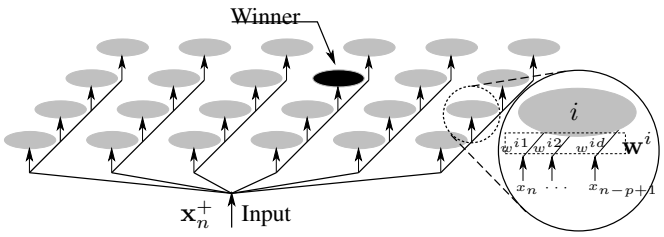

Figure 1. Sketch of a 2D-SOM: winner neuron i^* highlighted.

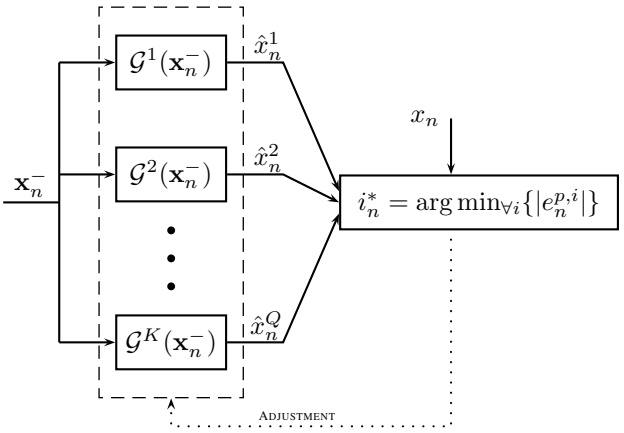

Figure 2. Sketch of the OPM network.

which can be tuned in an adaptive, self-organized fashion. In this architecture, a given operator may become sensitive to a certain dynamical range of the input time series. More specifically, let us assume that at discrete time step n a given time series can be described by the following global model

$$x_n = H\left(\mathbf{x}_n^-\right) + \varepsilon(t) \tag{5}$$

where $\mathbf{x}_n^- = [x_{n-1}\ x_{n-2}\ \cdots\ x_{t-p}]^T$ is a vector comprised of p last samples of a time series, $H(\cdot)$ is an unknown mapping, and $\varepsilon(t)$ is a random sample from a gaussian white noise process with zero mean and variance σ_ε^2.

Let us also assume that the global model $H(\cdot)$ can be approximated with arbitrary accuracy by a set of Q local linear models $\mathcal{G}^i$, $i = 1, \ldots, Q$ associated with the neurons in the OPM model. Since our target application is anomaly detection in time series, we are interested in providing a good estimate of the current state x_n of the system being monitored, given $\mathbf{x}_n^-$ and the local models $\mathcal{G}^i(\cdot)$.

Let $\hat{x}_n^i$ be the estimate of the current state of the system computed by neuron i. Then,

$$e_n^{p,i} = x_n - \hat{x}_n^i, \tag{6}$$

is the *prediction error* due to neuron i. If the system is working normally, then one should expect a small value for the prediction error. Otherwise, something anomalous may be occurring. A common choice for the local filter $\mathcal{G}^i$ is the linear autoregressive (AR) model. In this case, the estimate due to neuron i of the current value of the time series is given by:

$$\hat{x}_n^i = [\mathbf{w}_n^i]^T \mathbf{x}_n^- = \sum_{j=1}^{p} w_n^{ij} x_{n-j}^- \tag{7}$$

where $\mathbf{w}_n^i = [w_n^{1i}\ w_n^{2i}\ \cdots\ w_n^{pi}]^T$ is the coefficient (weight) vector associated to neuron i. The winning neuron i_n^* is the one providing the best estimation of x_n. In other words, the winning filter at time n is the one with the smallest absolute value for the prediction error:

$$i_n^* = \arg\min_{\forall i}\{|x_n - \hat{x}_n^i|\} = \arg\min_{\forall i}\{|e_n^{p,i}|\} \tag{8}$$

where $|u|$ denotes the absolute value of the scalar u. The quantity $e_n^{p,i^*} = x_n - \hat{x}_n^{i^*}$ is the prediction error produced by the current winning neuron. The learning rule for the weight vector of neuron i is a LMS-like equation, slightly modified by the inclusion of a neighborhood function:

$$\mathbf{w}_{n+1}^i = \mathbf{w}_n^i + \eta_n \mathcal{H}_n(i^*, i)\, e_n^{p,i}\, \mathbf{x}_n^- \tag{9}$$

$$= \mathbf{w}_n^i + \eta_n \mathcal{H}_n(i^*, i)\left(x_n - \hat{x}_n^i\right)\mathbf{x}_n^-, \tag{10}$$

where $\mathcal{H}_n(i^*, i)$ is the neighborhood function as defined in Eq. (4). A successfully trained OPM network should fit Q local autoregressive models to a given nonstationary time series. Note that an OPM with one single neuron (i.e. $Q = 1$) is equivalent to a linear AR model.

3 Novelty Detection Methodologies

This section describes two variations of the same basic procedure: in simple words, take the quantization or prediction errors obtained at the training phase of the ANN algorithm and use them to compute decision thresholds, which are used to classify test samples as NORMAL or NOVEL. Figure 3 presents a box diagram with the steps followed in this study.

3.1 Quantization Error Based Approach

It has become common practice [15, 3, 1] to use the quantization error of the winner neuron

$$e_n^{q,i^*} = \|\mathbf{x}_n^+ - \mathbf{w}_n^{i^*}\|, \tag{11}$$

as a measure of the degree of proximity of $\mathbf{x}_n^+$ to a statistical representation of normal behavior encoded in the weight vectors of the SOM.

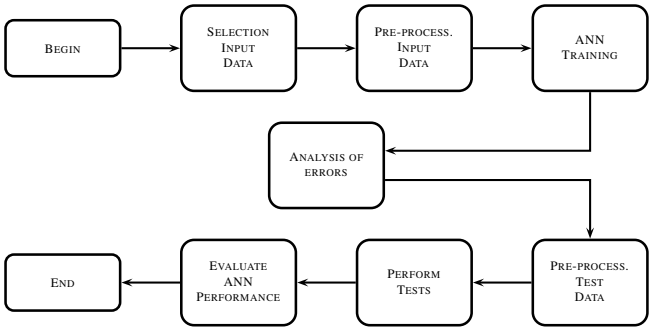

Figure 3. Block diagram of the chosen methodology for novelty detection.

Once the standard SOM has been trained, we present the training data vectors once again to this network. From the resulting set of quantization errors $\{e_n^{q,i^*}\}_{n=1}^N$, computed for all training vectors, we compute decision thresholds for the anomaly detection tests. For a successfully trained network, the sample distribution of these quantization errors should reflect the "known" or "normal" behavior of the input variable whose time series model is being constructed.

Several procedures to compute decision thresholds have been developed in recent years, most of them based on well-established statistical techniques [7], but we apply the method recently proposed in [3]. For a given significance level α, we are interested in an interval within which we can certainly find a percentage $100(1-\alpha)$ (e.g. $\alpha = 0.05$) of normal values of the quantization error. Hence, we compute the lower and upper limits of this interval as follows:

- **Lower Limit** (τ^-): This is the $100\frac{\alpha}{2}$th percentile[1] of the distribution of quantization errors associated with the training data vectors.
- **Upper Limit** (τ^+): This is the $100(1-\frac{\alpha}{2})$th percentile of the distribution of quantization errors associated with the training data vectors.

Once the decision interval $[\tau^-,\ \tau^+]$ has been computed, any anomalous behavior of incoming data samples can be detected by means of the following simple rule:

$$\begin{array}{rl} \text{IF} & e_n^{q,i^*} \in [\tau^-,\ \tau^+] \\ \text{THEN} & \mathbf{x}_n^+ \text{ is } \textbf{NORMAL} \\ \text{ELSE} & \mathbf{x}_n^+ \text{ is } \textbf{ABNORMAL} \text{ (or } \textbf{NOVEL}\text{).} \end{array} \quad (12)$$

[1]The percentile of a distribution of values is a number N_α such that a percentage $100(1-\alpha)$ of the sample values are less than or equal to N_α.

3.2 Prediction Error Based Approach

In order to use the OPM for anomaly detection purposes we also defined a decision interval $[\tau^-, \tau^+]$, but now for the distribution of prediction errors produced by the training data. Computation of the lower/upper limits of this interval follows the same logic of the technique presented in previous section, except for the fact that now we use the distribution of the prediction errors of the winning neurons:

- **Lower Limit** (τ^-): This is the $100\frac{\alpha}{2}$th percentile of the distribution of prediction errors $\{e_n^{p,i^*}\}$.
- **Upper Limit** (τ^+): This is the $100(1-\frac{\alpha}{2})$th percentile of the distribution of prediction errors $\{e_n^{p,i^*}\}$.

The decision rule for incoming data samples is then written as follows:

$$\begin{array}{rl} \text{IF} & e_n^{p,i^*} \in [\tau^-,\ \tau^+], \\ \text{THEN} & \mathbf{x}_n^+ \text{ is } \textbf{NORMAL} \\ \text{ELSE} & \mathbf{x}_n^+ \text{ is } \textbf{ABNORMAL} \text{ (or } \textbf{NOVEL}\text{).} \end{array} \quad (13)$$

4 Computer Simulations

The feasibility the proposed method is evaluated using input signals derived from four different dynamic systems, three of them are realizations of chaotic series. The first one is composed by the x component of Lorenz equations

$$\dot{x} = \sigma_L(y-x),\ \dot{y} = x(\alpha_L - z) - y,\ \dot{z} = xy - \epsilon_L z, \quad (14)$$

which exhibits chaotic dynamics for $\sigma_L = 10$, $\alpha_L = 28$ and $\epsilon_L = 8/3$. The second and third signals come from two different Mackey-Glass series, with distinct τ delays:

$$\dot{x} = Rx(t) + \frac{Px(t-\tau)}{(1+x(t-\tau)^{10})}, \quad (15)$$

with $P = 0.2$, $R = -0.1$ and $\tau = 17$ or $\tau = 35$. The fourth signal is an autoregressive process AR(2):

$$x_n = 1.9x_{n-1} - 0.99x_{n-2} + N_n, \quad (16)$$

with N_n is a random sample from a gaussian white noise process with zero mean and variance $\sigma_n^2 = 10^{-3}$. Typical realizations of each signal are shown at the top of Figure 4.

The novelty detection experiment was designed to perform the on-line detection of an anomalous signal, after training the networks with a sequence considered **NORMAL**. The role of NORMAL signal was assigned to the Lorenz series, which is then used to train the SOM and OPM networks and to compute the decision thresholds. The three remaining signals (two Mackey-Glass series and the

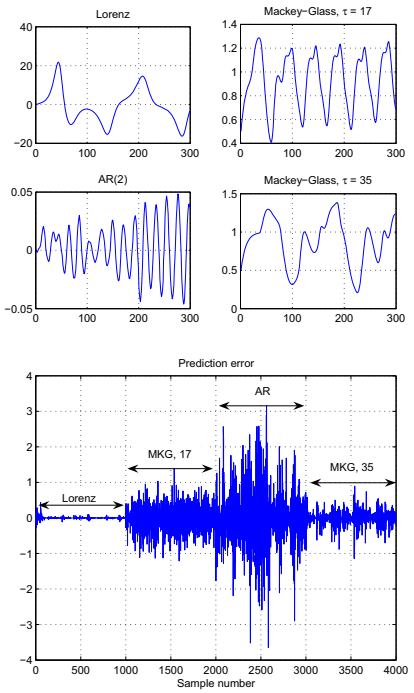

Figure 4. Realizations of the time series used in the simulations and prediction error e_n^{p,i^*}.

AR signal) are used for testing purposes. Each individual signal is comprised of 1000 samples. The SOM and OPM networks had a one-dimensional topology and were trained with the following parameters: $Q = 30$, $T_{max} = 1000$, $\eta_0 = 0.5$, $\eta_T = 0.001$, $\sigma_0 = Q/2$, $\sigma_T = 0.001$ and $p = 10$.

In Figure 4 (bottom) one can see the prediction errors $\{e_n^{p,i^*}\}$ collected from the winning neuron i^* for the OPM network. It is possible to notice that (i) the low prediction errors produced when the for the first $k = 1000$ samples, revealing the good capability of the OPM to produce a correct model of normal behavior, and (ii) that when signals of different dynamics are presented, the resulting prediction errors are much higher.

It is also illustrative to observe the cumulative distribution function (CDF) of the prediction errors for the OPM network. Figure 5 depicts the CDFs for $e_n^{i^*}$ obtained from all the different testing sequences, where it is possible to verify that **ABNORMAL** behavior results in distributions with higher variance.

Concerning the performance of the standard SOM for novelty detection in time series, two experiments were performed. First, an experiment to detect noisy samples was performed adding to the AR(2) series a sequence of "spikes" repeated at each 100 samples, simulating a disturbance on the measurement. Figure 6 shows the noisy AR(2) series and the corresponding quantization error generated

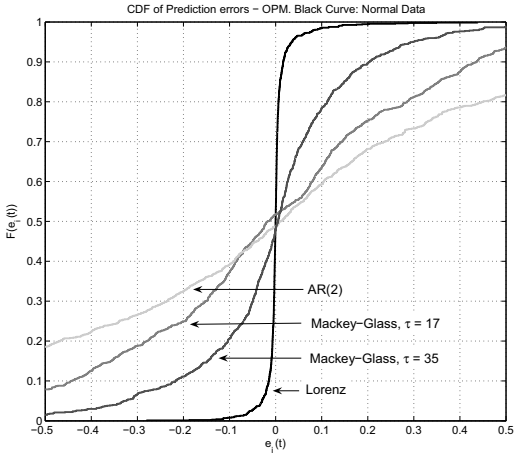

Figure 5. CDFs for the prediction errors.

by the standard SOM network. The time instants where the spikes were added shows clearly higher quantization error. The thresholds $[\tau^-,\ \tau^+]$ can be adjusted to minimize typical performance metrics of binary classifiers, such as false positive and false negative rates (or any weighted combination of them).

The second experiment involves a real-world signal representing the current on a solenoid of a valve aimed to control the injection of combustion in the space shuttle (Figure 7). It is possible to notice the irregular behavior of the valve at the two rightmost cycles. The corresponding quantization error signal is shown at Figure 8, including the thresholds $[\tau^-,\ \tau^+]$ corresponding to the 5% and 95% percentiles of e_n^{q,i^*}. Again, the method performed satisfactorily, with clear detection of abnormal states.

5 Conclusion

This work described two self-organizing paradigms for detecting abnormal samples in time series by non-parametric analysis of either the quantization or prediction errors available after training selected ANNs. The methodology based on the prediction errors of the OPM network is novel. The methods were evaluated in novelty detection tasks for univariate time series, but its extension to multivariate time series is straightforward. Decision thresholds used to classify data as normal or abnormal may be further optimized to minimize typical performance metrics of binary classifiers, such as false positive and false negative rates (or any weighted combination of them). Future work on the subject includes the combination of the method with special pre-processing of time series: before feed then into the ANNs, the idea is to use using different memory kernels for the SOM networks, such as the Gamma memory [14].

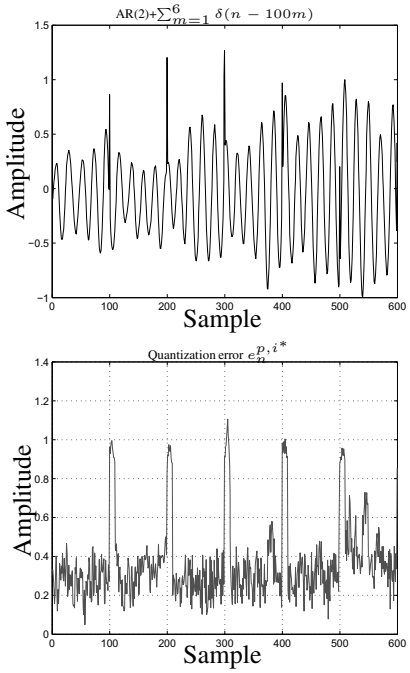

Figure 6. AR(2) series with added noise and respective quantization error e_n^{q,i^*}.

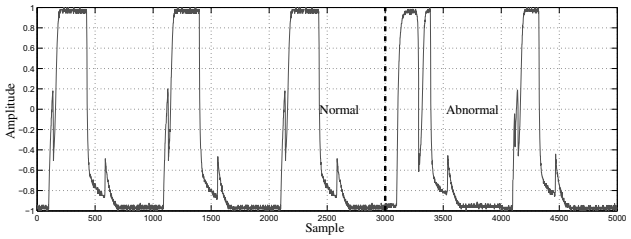

Figure 7. Energizing cycles of a space shuttle solenoid.

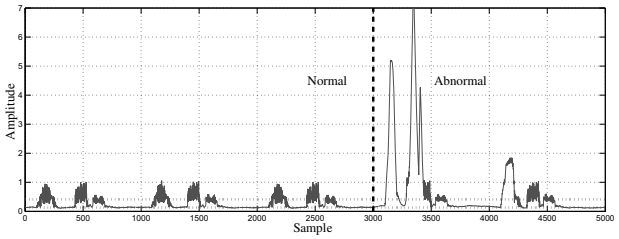

Figure 8. Quantization error e_n^{q,i^*} for the solenoid series.

References

[1] E. Alhoniemi, J. Hollmén, O. Simula, and J. Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering*, 6(1):3–14, 1999.

[2] G. A. Barreto and A. F. R. Araújo. Time in self-organizing maps: An overview of models. *International Journal of Computer Research*, 10(2):139–179, 2001.

[3] G. A. Barreto, J. C. M. Mota, L. G. M. Souza, R. A. Frota, and L. Aguayo. Condition monitoring of 3G cellular networks through competitive neural models. *IEEE Transactions on Neural Networks*, 16(5):1064–1075, 2005.

[4] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 3rd edition, 1994.

[5] C. L. Fancourt and J. C. Principe. On the use of neural networks in the generalized likelihood ratio test for detecting abrupt changes in signals. In *Proceedings of the 2000 IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)*, volume 3, pages 243–248, 2004.

[6] T. C. Fu, F. L. Chung, V. Ng, and R. Luk. Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, pages 27–37, 2001.

[7] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[8] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg, 2nd extended edition, 1997.

[9] J. Lampinen and E. Oja. Self-organizing maps for spatial and temporal AR models. In *Proceedings of the 6th Scandinavian Conference on Image Analysis (SCIA'89)*, pages 120–127, Helsinki, Finland, 1989.

[10] H.-J. Lee and S. Cho. SOM-based novelty detection using novel data. *Lecture Notes on Computer Science*, 3578:359–366, 2005.

[11] T. W. Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[12] M. Markou and S. Singh. Novelty detection: a review - part 1: Statistical approaches. *Signal Processing*, 83:2481–2497, 2003.

[13] M. Markou and S. Singh. Novelty detection: A review - part 2: Neural network based approaches. *Signal Processing*, 83:2499–2521, 2003.

[14] J. C. Principe, B. de Vries, and P. G. de Oliveira. The gamma filter-a new class of adaptive IIR filters with restricted feedback. *IEEE Transactions On Signal Processing*, 41(2):649–656, February 1993.

[15] S. T. Sarasamma and Q. A. Zhu. Min-max hyperellipsoidal clustering for anomaly detection in network security. *IEEE Transactions on Systems, Man and Cybernetics*, B-36(4):887–901, 2006.

[16] S. Singh and M. Markou. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):1041–4347, 2004.

[17] M. Wong, L. Jack, and A. Nandi. Modified self-organising map for automated novelty detection applied to vibration signal monitoring. *Mechanical Systems and Signal Processing*, 20(3):593–610, 2006.