Feature-based Classification of Time-series Data *

ALEX NANOPOULOS ROB ALCOCK YANNIS MANOLOPOULOS
Data Engineering Lab, Department of Informatics
Aristotle University
Thessaloniki 54006
GREECE
{alex,rob,manolopo}@delab.csd.auth.gr

Abstract: In this paper we propose the use of statistical features for time-series classification. The classification is performed with a multi-layer perceptron (MLP) neural network. The proposed method is examined in the context of Control Chart Pattern data, which are time series used in Statistical Process Control. Experimental results verify the efficiency of the feature-based classification method, compared to previous methods which classify time series based on the values of each time point. Moreover, the results show the robustness of the proposed method against noise and time-series length.

Key-words: Data mining, time series, classification, statistical features

1 Introduction

Data mining is the process of pattern identification in large databases [1]. The main objectives of data mining are *prediction* and *description*. Data mining methods belong to several categories. *Regression* maps data to prediction values. *Generalization* produces a simple description from complex data and *association* finds dependencies among data. *Clustering* identifies a set of types with which data can be categorized, whereas *classification* maps data to a set of a predefined types. Data mining has been mainly applied to relational data. Non-relational data present important challenges to data mining due to their size and dimensionality.

Time-series data are supported by many database systems. A time series is a sequence of real numbers representing the values of a variable over time. They have found applications in temporal [2] and scientific databases, as well as in data warehouses containing a variety of data types, from stock market prices to electro-cardiograms. Mining time-series data can reveal important patterns, such as similarities [3], trends [4] or periodicity [5]. Since time-series data tend to grow rapidly over time, they present several performance issues to data mining algorithms.

Time-series classification involves the learning of a function that maps a series into a class from a set of predefined classes. Therefore, it requires the existence of a set of well-defined categories. The class definition is not always an easy task (e.g. in stock-market data) and may be obtained from a domain expert. Time-series classification finds applications in electrocardiogram (ECG) data, weather measurements, in statistical process control (SPC). More particularly, SPC uses time series for the representation of several critical parameters of a system. These time series are called control chart

*Research performed under the European Union's TMR Chorochronos project, contract number ERBFMRX-CT-96-0056 (DG12BDCN).

patterns (CCPs) and they are used for the detection of abnormalities in the system's condition.

In [6], a control chart classification approach is followed based on neural network classifiers. CCPs are synthetically generated, belonging to one of six predefined types, and are presented to a feed-forward back-propagation neural network, with every value having one input neuron. The proposed classifier achieves up to 97.1% accuracy. This approach presents the drawback of sensitivity to noise and time-series length. As will be shown in the following sections, when the amount of noise in the series is increased, the performance of classification is affected due to the implication of each distinct time point. Also, increased time-series length has an impact on both the classification performance and response times.

In this paper, we present a time-series classification method based on feature-extraction. We propose the use of several statistical, first and second order features and evaluate their benefits to classification performance. Since features carry information for the time series which is not based on individual time points, they are less sensitive to noise. Additionally, the number of features is much less compared to the length of the time series. Therefore training times are reduced drastically.

Related work in time-series data mining has mainly focused on fast similarity search. In [3], time series are represented by the first few coefficients of their Discrete Fourier Transform, which are then stored in a spatial data structure. This work was generalized by [7] for subsequence matching. In [8], two series are similar if enough of their portions can be included by the same orthogonal envelope. More recent work on similarity search include [9] and [10] where similarity is defined in terms of a set of linear transformations. In [11], normalization of time series is used to make similarity insensitive to scaling and shifting. Yi et al. [12] present algorithms for time-series similarity using time warping as a distance function. Other work related to time-series similarity include [13, 14, 15].

The rest of the paper is organized as follows: Section 2 gives a detailed description of the CCP classification problem. Previous work on CCP classification is presented in Section 3. Section 4 presents the proposed method and Section 5 contains experimental results that show the performance of all methods. Finally, Section 6 concludes the paper and gives some directions for further work.

2 Problem Description

There exist several examples of categorized time-series data. For instance, the *Shuttle* data[1] set consists of the output from 109 sensors from the first eight hours of Space Shuttle mission STS-067. Sensors and their output can be categorized with respect to the phenomena they measure. Other examples include electrocardiogram (ECG) data, which contain a number of human ventricular events.

In this work, we focus on a specific type of categorized time series, the CCP. Statistical Process Control procedures use such CCPs for the assessment of the status of industrial systems. In [6], CCPs are categorized into six types and the corresponding equations are provided for the synthetic generation of CCPs for each type. The types that CCPs can exhibit are: normal, cyclic, linear trends (increasing or decreasing) and shifts (upward or downward). Equations 1 - 4 describe the way to generate patterns from each type and Table 1 explains each symbol used in equations. Figure 1, illustrate examples of produced CCPs from each type.

CCP data present a context where classification can be performed effectively, since it contains well described classes. However, non stationary (in mean or variance) time series may impact the effectiveness of classification. Such series can be generated using the GARGH (Generalized Autoregressive Conditional

[1] Available at http://www.ics.uci.edu/ mlearn/MLRepository.html

Heteroskedasticity) model, as described in [16]. In this case, it is expected that none of the existing methods will perform effectively. In this paper we are interested in a relative comparison of time-series classification methods. Therefore, we leave the examination of the classification of these type of data as a direction of future work.

$$y(t) = \mu + r(t)\sigma \quad [Normal] \tag{1}$$

$$y(t) = \mu + r(t)\sigma + a\sin(2\pi t/T) \quad [Cyclic] \tag{2}$$

$$y(t) = \mu + r(t)\sigma \pm gt \quad [Linear \ \ trend] \tag{3}$$

$$y(t) = \mu + r(t)\sigma \pm ks \quad [Shift] \tag{4}$$

$y(t)$	time-series value
t	time
μ	mean value (taken as 30)
σ	standard deviation
α	amplitude of cyclic variations (taken in the range 5 to 15)
g	magnitude of gradient trend (taken in the range 0.2 to 0.5)
k	determines shift position (k is 0 before shift position, 1 after)
r	normally distributed random number between 0 and 3
s	shift magnitude (taken in the range 7.5 to 20)
T	period of cycle (taken in the range 10 to 15)

Table 1: Parameters for synthetic generation of CCPs

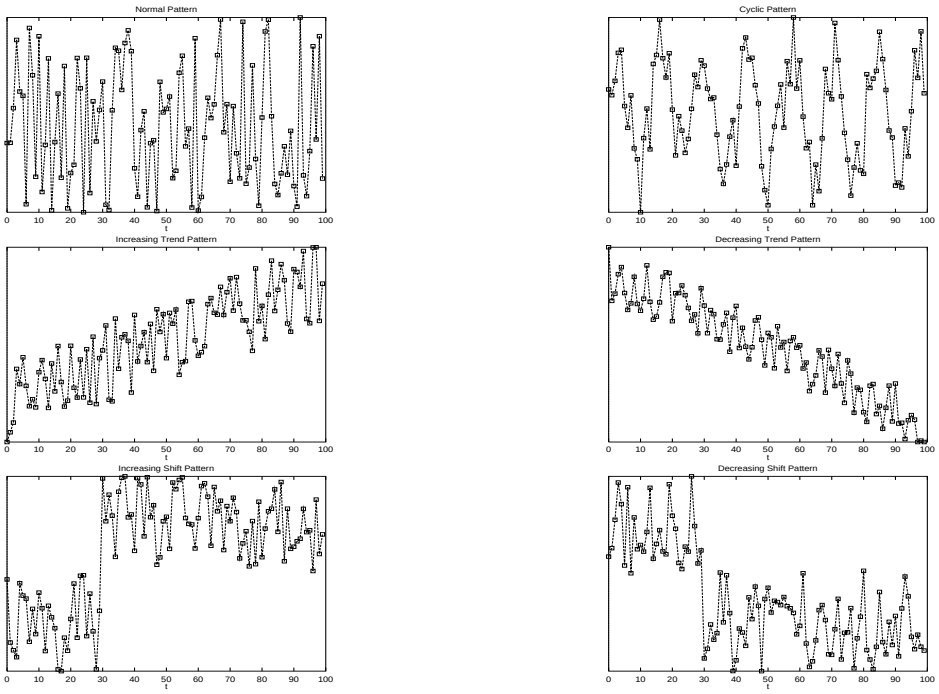

Figure 1: Examples from each type of control chart patterns.

There exist several types of classification methods which can be categorized into linear and non-linear. Linear classifiers include decision trees [17] and rule-induction methods. They have the advantage of

simple interpretation. Due to the nature of their learning function, which is based on data attributes, they are appropriate for relational data. Time-series data lack the existence of attributes. Non-linear classification methods include neural networks, adaptive spline methods and projection pursuit regression [18].

One of the most common types of neural network classifier is the Multi-layer Perceptron (MLP) neural network. The MLP is based on the known structure of the human brain. It contains layers of neurons with weighted connections between these neurons. The output of each neuron is the sum of its inputs passed through an activation function, such as the sigmoidal function. Neurons are organized into layers: input, hidden (one or more layers) and output. The main advantages of the MLP are that it learns from examples and is able to form non-linear boundaries between classes. It finds, by adapting its weights, a relationship between input data and output types. Thus, the user does not need to have so much domain knowledge about the problem. In this work we focus on MLPs with one hidden layer. Although, several other structures may achieve improved results, we are interested on the relative comparison of the presented methods. The following two sections describe the use of the MLP neural network for CCP classification.

3 Previous Work on Time-series Classification

Pham and Oztemel [6] worked on the classification of CCPs. They generated 83 training patterns from each of the six types, with each pattern having a length of 60 points. A MLP neural network was employed to classify the patterns. The approach employed was to have one input neuron for every value in the time series. Thus, the network had 60 inputs and 6 outputs (one for each class). For testing, 1002 unseen patterns were generated. The MLP achieved an accuracy of 96% on this test data. Next, experiments were carried out using three neural networks and combining their outputs using a decision-making module. The resulting composite system was able to classify patterns with an accuracy of 97.1%.

In later work, Pham and Oztemel [19] proposed the use of the Learning Vector Quantization (LVQ) neural network. As in the previous work, all the points in the time series were used as inputs to the network. The LVQ network gave a worse performance than the MLP network, achieving an accuracy of 92.3% on test data. A modified version of LVQ, called LVQ-X, was proposed in the paper and a higher accuracy of 97.7% was obtained.

Keogh and Pazzani [20] proposed a time-series representation based on piece-wise linear segmentation [21]. This representation provides data compression and noise reduction. In the paper, a distance measure is defined which corresponds to how far two segments are from being parallel. This distance measure is insensitive to translation, linear trends and discontinuities. The classification task proposed distinguishes time series which belong to a single well-defined class from those that do not exhibit any structure. The algorithm merges all time series into a single sequence. This is done using an agglomerative clustering approach by merging each time two series belong to the same class. Due to the merging approach, this method was tested in [20] against time series belonging to a single class.

Recently, in [22], there is proposed a method for classifying CCP data. The paper discusses the use of artificial neural networks for recognition of the shape from CCP data. Synergistic, distributed and distributed synergistic neural networks are proposed and focus is given on learning efficiently non-linear characteristics and overlapping ranges of values of the data set describing CCPs.

Finally, feature selection was used in [23] for classifying speech signals. Several statistical features, along with others specifically designed for speech data, are proposed and their effectiveness is verified

with experimental results. The later work motivated ours on using features for classifying CCPs with neural network classifiers.

The approach of taking the actual values of the time series as classifier inputs [6] presents the following drawbacks:

- It is sensitive to the amount of noise in the time series. Since each input contributes to the classification task, noise can affect the accuracy of the method more severely.
- It is sensitive to the length of the time series. When the time series becomes large, it is impractical to have a neural network with such a large number of inputs.
- It requires that all chart patterns are of the same length (due to the constant number of input neurons). Using features instead, patterns of different lengths could be used, as long as the same number of features is extracted.

4 Proposed Method

In this section, we propose the extraction of several features from time series and their use for classification. The classification is performed based on the features for each series and not on the actual values. Many different features have been employed in image processing and pattern recognition to classify objects. There are many features in image processing which can be categorized in those who describe intensity an those who describe shape. The extraction of appropriate features has been recognized as an important problem.

Evidently, the nature and the number of features needed depends on their discriminatory quality. The required characteristics for the selected features are ease of computation, insensitivity to transformations and resulting performance. For time-series data, and more particularly for CCP data, we propose the use of *statistical features*, which are commonly used in image processing.

The proposed statistical features were selected because they are simple and easy to implement. The invocation of more elaborated features, like *moments* of several orders, is expected to improve the efficiency of the method. Also, the invocation of an automated feature selection method could help to improve the results of the method. Nevertheless, statistical features are a starting point for the evaluation of the method, and the improved methods of feature selection will be addressed by future work.

Features can be divided into first and second order features. First-order features are based on the actual values of the series whereas, second-order features are based on the differences of nearby values.

The most common statistical features are the mean value μ, standard deviation σ, skewness $SKEW$ and kurtosis $KURT$. Skewness and kurtosis contain information on the shape of the distribution of the time-series values. More precisely, $SKEW$ characterizes the degree of asymmetry of values around the mean value. KURT measures the relative peakness or flatness of the value distribution relative to a normal distribution. The equations for these features are:

$$\mu = \frac{\sum_{t=1}^{n} y(t)}{n} \tag{5}$$

$$\sigma = \sqrt{\frac{\sum_{t=1}^{n} (y(t) - \mu)^2}{n}} \tag{6}$$

$$SKEW = \frac{\sum_{t=1}^{n}(y(t) - \mu)^3}{n\sigma^3} \tag{7}$$

$$KURT = \frac{\sum_{t=1}^{n}(y(t) - \mu)^4}{n\sigma^4} - 3 \tag{8}$$

There exist several second-order features in image processing. The most common ones are called co-occurrence features. For this work, new second-order features are used which are more suitable for time-series classification. The features are simple to calculate and are similar to the first-order features employed. For the calculation of the second-order features used in this work, it is first necessary to calculate a transformed time series $y'(t)$ based on the original $y(t)$ and a user-specified value D:

$$y'(t) = y(t + D) - y(t), \quad 1 \leq t \leq n - D \tag{9}$$

D is the distance between the points in the time series being compared. From $y'(t)$, the four statistical features are calculated using equations 5 - 8. These features will be referred to as μ_2, σ_2, $SKEW_2$ and $KURT_2$. $y'(t)$ contains information about the shape of the original series $y(t)$ and also acts as a noise filter. Second-order features can be employed with first-order features to give a better representation of y(t).

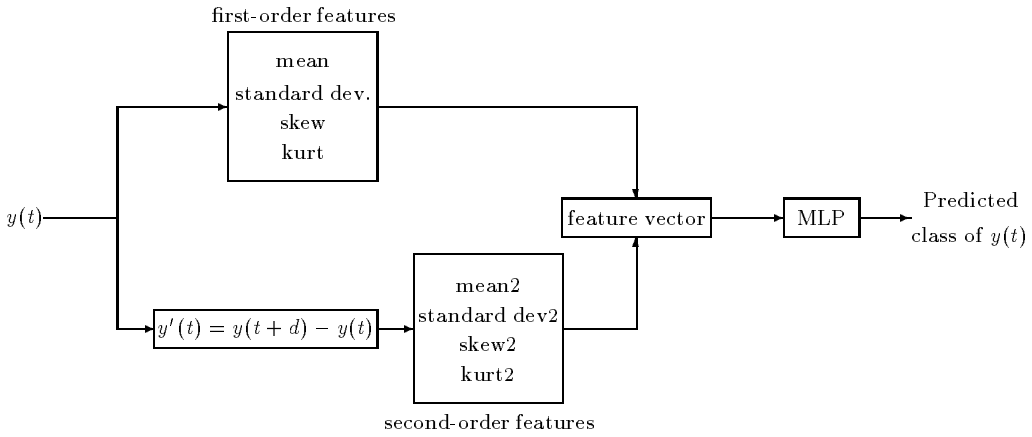

Figure 2: Proposed method

The proposed method is presented in Figure 2. First-order features are calculated from $y(t)$ using equations 5 - 8. Then, from $y'(t)$, we calculate the second-order features. First and second-order features form the feature vector. Feature vectors of each time series are presented to a MLP neural network for classification. Since the length of the feature vector is eight, the MLP has much less input neurons compared to the method described in the previous section, particularly when the time-series length increases.

5 Experimental Results

This section presents the results obtained from the comparison of the method proposed by [6], which will be referred to as *VB* (value based), with the proposed one, which will be referred to as *FB* (feature based). A training set was generated for both methods, consisting of 180 time series. These series were generated using equations of Section 3, where each class has the same number of instances. The neural

network had three layers, one for input, one hidden and one for output. The number of hidden neurons was equal to half the sum of the input and output neurons. The time series of the training set were presented to the neural network in a round-robin manner, i.e. one of the six types each time. The evaluation of the trained neural network was performed with 6,000 time series generated using again the equations of Section 3. Both train and test data are following the same distribution, i.e. they were generated with the same values for the corresponding parameters. The performance measure was the classification accuracy for the evaluation set. Each measure was repeated five times and the overall result was taken as the average. The averaging was performed on the results of the several experiments in order to derive more reliable results. Notice that this averaging should not be confused with the use of a structure with multiple neural networks which are combined using an average of the independent results. Of course, such neural network types are expected to improve the performance of classification, but, as explained, the objective of this paper is the relative comparison of existing methods.

First, we measured the impact of the neural network parameters *momentum(α)* and *learning rate(η)*. Parameter η represents how fast the weights at each neuron are changed with respect to the error. Parameter α shows how much the weights are changed compared to the change in the previous iteration. We tested three values for η: $0.001, 0.01, 0.1$, which represent a low, medium and high value, respectively. Though 0.1 may be not be considered large, for larger values of η, the neural network did not manage to be trained. For α, we tested values $0.1, 0.5$ and 0.9, representing a low, medium and high value, respectively. Table 2 presents the results for time series generated having length 60 and standard deviation equal to 2.0. Note that the standard deviation is a measure of the noise the time series contains. The stopping criterion for the neural network training was either 10,000 iterations or an RMS value equal to 0.05.

VB	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$
$\eta = 0.001$	0.986	0.981	0.974
$\eta = 0.01$	0.166	0.166	0.955
$\eta = 0.1$	0.167	0.167	0.167

FB	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$
$\eta = 0.001$	0.969	0.967	0.967
$\eta = 0.01$	0.935	0.964	0.970
$\eta = 0.1$	0.167	0.167	0.167

Table 2: **a.**Accuracy of VB method **b.**Accuracy of FB method

Table 2 illustrates that the VB method achieves high accuracy for a small η value (0.001). The accuracy of FB is also high, but slightly less than that of VB. For medium and high η values, the VM method did not perform well since the neural network could not been satisfactory trained. This is due to the fact that the training for the VB method is based on each value of the time series. The FB method performs more robustly since it achieved high accuracy for medium η values. Parameter α did not have a significant impact except in the case when η was set to 0.01 because for a high α value, the VB method managed to produce high accuracy.

Time-series data usually contain noise. Noise can be categorized as white, i.e. Gaussian noise, which is applied to whole time series, or pulsive, which represents abrupt variations of the values of the time series at few time instances. For CCP signals, as described in Section 3, we consider noise as a function of the standard deviation σ. Since high deviation from the mean value represents a time series with a less normal shape, σ can represent the amount of noise that the time series contains. Notice that noise is considered to corrupt the existing values of the time series and not to add new, noisy, values in the series. Figure 3a illustrates the accuracy of VB and FB methods versus σ. The length of the time series was set to 60, η was 0.001 and α 0.5. The stopping criterion for the neural network training was either 10,000 iterations or an RMS value equal to 0.05. As shown in the figure, both methods perform well for low σ values. As σ increases, performance of the VB method deteriorates. FB outperforms the VB

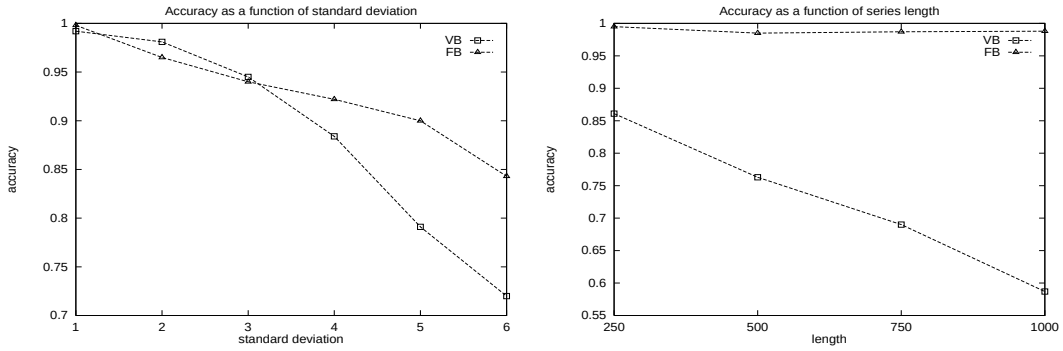

Figure 3: **a.**Accuracy versus σ **b.**Accuracy versus series length.

method due to the fact that it is not based on individual values and thus it is less affected by an increase in the standard deviation.

Next, we measured the impact of time-series length on the accuracy of classification methods. Evidently, the VB method is more sensitive to the time-series length because it takes as input the whole series. The length of the time series is application dependent. For instance, stock-market data tend to be very large. Figure 3b presents the accuracy of VB and FB methods with respect to time-series length. We tuned the parameters separately for each method. For the VB method, η was set to 0.001 and α to 0.8. With smaller values of α and larger values of η, the VB method tended to reach a local minimum. For the FB method, η was set to 0.01 and α to 0.5. The stopping criterion for the neural network training was either 5,000 iterations or an RMS value less than 0.05. As shown in Figure 3b, the VB method is sensitive to time-series length, whereas the FB method is not.

Besides accuracy, time-series length has significant impact on training time. Table 3 presents the time taken for the neural network to be trained for each method. The VB method requires significantly longer training times for larger time series because it involves many more computations for connections between input and hidden neurons. On the other hand, the training time for the FB method is not sensitive to time-series length because it involves a small constant number of features for the training.

	$length = 250$	$length = 500$	$length = 750$	$length = 1000$
VB	15 min	45 min	70 min	100 min
FB	1.3 min	1.25 min	1.22 min	1.24 min

Table 3: Training times versus time-series length

6 Conclusions

We have proposed a feature-based classification method for time-series data. Several first and second-order statistical features have been presented. These features are based on those utilized in image processing applications. The classification task is performed using a MLP neural network. The experimental results for control chart pattern data confirm the efficiency of the proposed method compared to a previous one, which classifies time series by using the value of the series at each time point. We show that the use of features results in better classification accuracy in the presence of noise and leads to faster and better classification for large time series.

Future work may involve the examination of other features, like moments, and the implementation

of a more detailed feature extraction method. Additionally, other types of time-series, like the ones produced by the GARGH model [16], and real data sets can be examined. Moreover, other classification methods (like Bayesian classifiers), instead of neural networks, can be examined.

References:

[1] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press 1996.

[2] K. K. Al-Taha, R. T. Snodgrass, M. D. Soo, Bibliography on Spatiotemporal Databases, *SIGMOD Record* Vol.22, No.1, 1993, pp.59-67.

[3] R. Agrawal, C. Faloutsos, A. Swami. Efficient Similarity Search in Sequence Databases, *Proc. 4th International Conference Foundations of Data Organization and Algorithms (FODO'93)*, 1993.

[4] D. J. Berndt, J. Clifford, Finding Patterns in Time Series: A Dynamic Programming Approach, in [1].

[5] J. Han, G. Dong, Y. Yin, Efficient Mining of Partial Periodic Patterns in Time Series Databases, *Proc. of the 15th International Conference on Data Engineering*, 1999.

[6] D. T. Pham, E. Oztemel, Control Chart Pattern Recognition Using Neural Networks, *Journal of Systems Engineering*, Vol.2, 1992, pp.256-262.

[7] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Subsequence Matching in Time-series Databases, *Proc. of the 1994 ACM SIGMOD International Conference on Management of Data*, 1994.

[8] R. Agrawal, K.-I. Lin, H.S. Sawhney, K. Shim, Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series databases, *Proc. 21th International Conference on Very Large Data Bases (VLDB'95)*, 1995.

[9] D. Rafiei, On Similarity-Based Queries for Time Series Data, *Proc. of the 15th International Conference on Data Engineering*, 1999.

[10] D. Rafiei, A. Mendelzon, Similarity-Based Queries for Time Series Data, *Proc. ACM SIGMOD International Conference on Management of Data*, 1997.

[11] D. Q. Goldin, P.C. Kanellakis, On Similarity Queries for Time-series Data: Constraint Specification and Implementation, *Proc. First International Conference of Principles and Practice of Constraint Programming (CP'95)*, 1995.

[12] B. -K. Yi, H.V. Jagadishm, C. Faloutsos, Efficient Retrieval of Similar Time Sequences Under Time Warping, *Proc. of the Fourteenth International Conference on Data Engineering*, 1998.

[13] H. Shatkay, S. Zdonic, Approximate Queries and Representations for Large Data Sequences, *Proc. of the Twelfth International Conference on Data Engineering*, 1996.

[14] R. Agrawal, G. Psaila, E. L. Wimmers, M. Zait, Quering Shapes of Histories, *Proc. 21th International Conference on Very Large Data Bases (VLDB'95)*, 1995.

[15] P. Seshadri, M. Linvy, R. Ramakrishnan, Sequence Query Processing,*Proc. of the ACM SIGMOD International Conference on Management of Data*, 1994.

[16] P. Granses, P. Homeled, On Forecasting Exchange Rates using Neural Networks, *Applied Financial Economics*, 1998.

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[18] J. F. Elder, D. Pregibon, A Statistical Perspective on Knowledge Discovery in Databases, *Advances in Knowledge Discovery and Data Mining*, in [1].

[19] D. T. Pham, E. Oztemel, Control Chart Pattern Recognition Using Learning Vector Quantization Networks, *Int. J. Prod. Res*, Vol.32, 1994.

[20] E. J. Keogh, M. Pazzani, An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback, *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, 1998.

[21] T. Pavlidis, S. Horowitz, Segmentation of Plane Curves, *IEEE Transactions on Computers*, Vol.23, No.8, 1974.

[22] M.A. Wani, D.T. Pham, Efficient control chart pattern recognition through synergistic and distributed artificial neural networks, *Journal of Engineering Manufacture*, Vol.213, 1999, pp.157-169.

[23] F.T. Dellaert, T. Polzin, A. Waibel, Recognizing Speech, *Proc. of the 4th Intl. Conference on Spoken Language Recognition*, 1996.