Models from data: analysis of industrial processes and telecommunication systems

Olli Simula*, Jaakko Hollmén and Esa Alhoniemi

Helsinki University of Technology, Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland, http://www.cis.hut.fi

Abstract

Modeling of systems is often based on the knowledge of the physical phenomena in the system. When physical knowledge is not available, or when modeling is too dif£cult due to nonlinearities, operational measurement data from the process may be used to build laws concerning the behavior of such a system. In this paper, we review some data-driven methods that have been successfully applied in industrial settings and in telecommunications. In particular, we present a neural network model — Self-Organizing Map — that has been applied in various projects.

Keywords: data analysis, neural networks, Self-Organizing Map, industrial processes, telecommunications

1 Introduction

Analytic system models have traditionally been used to characterize complex industrial systems and processes. The modeling is based on the knowledge of the physical phenomena and on the assumptions of the system behavior. However, many complex engineering systems, e.g., industrial processes, telecommunication systems and networks, and economical systems, are dif£cult or even impossible to model accurately. This is due to nonlinearities and complex or hidden dependencies between numerous factors affecting the system. Usually huge amounts of operational data are available in system databases. Therefore, the data-driven approach is a natural alternative in analysis, characterization, and understanding of these complex systems.

In data-driven methods, various types of data available in databases can be used. The data include, e.g., process measurements, system parameters, and other kind of stored information characterizing the system behavior. Data-driven approach provides means to analyze the system without an explicit physical model.

Adaptive algorithms based on computational intelligence often form the basis of various data-driven methods. Arti£cial Neural Networks (ANNs) represent this kind of computational algorithms. Their most important property is the capability to learn from examples and the ability to generalize. The Self-Organizing Map (SOM) [1, 2] is one of the most popular neural network models. Especially, due to its unsupervised learning capability it has proven to be very powerful in analysis of complex engineering systems.

This paper is organized in the following way. In Section 2, the framework of data-driven methods is brie¤y discussed. In Sections 2.1, 2.2, and 2.3, a neural network algorithm, the Self-Organizing Map is introduced. In Section 3, some industrial applications are presented. The section begins with data-driven methods for analysis of a pulp digester behavior (Section 3.1) and paper machine measurement signals (Section 3.2). It continues with a user pro£ling application within the context of mobile phone networks (Section 3.3) and mobile network monitoring (Section 3.4). Section 4 summarizes the paper.

2 Framework

As already stated earlier, analytic models are often dif£cult to build and therefore we turn the attention to data driven methods. Whereas there exists a large variety of methods in this framework [3, 4, 5, 6], we will concentrate on a sample that matches our research interests. The methods belong to the class of unsupervised learning, which can be used to £nd interesting structures of the data without extensive prior knowledge. In the following Sections (2.1, 2.2, 2.3), we will introduce an unsupervised neural network — Self-Organizing Map — that has been widely used in various applications for that purpose.

*e-mail: Olli.Simula@hut.fi (corresponding author)

2.1 Self-Organizing Map

The Self-Organizing Map (SOM) [1, 2] is a neural network model that can be ef£ciently used in analysis and visualization of high-dimensional data. It is the most popular neural network model based on unsupervised, competitive learning. The SOM has been used in wide-ranging applications (for a survey of SOM papers, see [7]), also, it has been applied for the analysis of industrial processes [8, 9, 10].

2.2 SOM in the data analysis process

Data analysis using the SOM is an iterative multistage process. In the following, the phases of this process are brie¤y described.

Data acquisition simply means obtaining the data for the analysis. It can be carried out, for example, by making a database query.

Data preprocessing means cleansing the data for further analysis. Some operations that are typical for this stage are data selection, segmentation, removal of errors, noise reduction, and symbolic to numeric transformations. It is impossible to depict any general procedure how the preprocessing should be carried out, because it is completely dependent on both the application and the data.

Feature extraction step transforms the cleansed data into feature variables – or brie¤y just features. For example, in some cases spectral features can be much more informative than the "raw" time domain representations of signals. It is important that the features describe all the phenomena in the data in most suitable way from the analysis point of view. Naturally, if there is no knowledge how to carry out the feature extraction task or if it is otherwise reasonable to use the original data as such, this step can be ignored.

Training of the SOM is the next step of the analysis. If the variables or features are not measured in the same units, i.e., they do not have a common scale, they should be normalized. Usually, the individual variables are normalized to zero mean and unit variance. Then, the SOM is trained using the algorithm that is described in detail in the next section.

Visualization and interpretation of the SOM is the last step of the analysis. During the training, the SOM has formed a compact representation of the data that re¤ects its properties. Thus, the SOM can be further used for tasks like correlation detection [11], cluster analysis [12] and novelty detection [13]. A review of different visualization methods for SOM can be found in [14].

2.3 SOM algorithm

The Self-Organizing Map is usually a two-dimensional regular lattice of map units (neurons). Each unit, denoted here by i, is represented by a prototype vector $\mathbf{m}_i$. All units have been connected with adjacent neurons, which are called its neighbors.

First, the prototype vectors are initialized with, for example, random values. Then, the training takes place: the values of the prototype vectors are adapted so that they re¤ect the properties of the input data.

Training a Self-Organizing Map is divided to two steps, which are applied alternately, typically thousands of times. First, one data vector $\mathbf{x}$ from the training data set is randomly selected and the corresponding best-matching (winner) unit (BMU) c is determined. The prototype vector of BMU, $\mathbf{m}_c$, is the one that is nearest to the data sample. In other words, it minimizes the Euclidean distance between $\mathbf{x}$ and $\mathbf{m}_c$:

$$c = \arg\min_i \|\mathbf{x} - \mathbf{m}_i\|. \quad (1)$$

In the next step, the prototype vectors of the winner and its neighbors are moved towards the data vector. It should be noted that the neighborhood is de£ned in terms of the lattice structure, not according to the distances between data samples and prototype vectors in the input space. The update step can be performed by applying

$$\mathbf{m}_i(t+1) := \mathbf{m}_i(t) + \alpha(t) h_c(t,i)[\mathbf{x}(t) - \mathbf{m}_i(t)], \quad (2)$$

where the last term in the square brackets is proportional to the gradient of the squared Euclidean distance $d(\mathbf{x}, \mathbf{m}_i) = \|\mathbf{x} - \mathbf{m}_k\|^2$. The learning rate $\alpha(t) \in [0,1]$ must be a decreasing function of time and the neighborhood function $h_c(t,i)$ is a non-increasing function around the winner unit. A good candidate is a Gaussian centered on the winner unit:

$$h_c(t,i) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_c\|^2}{2\sigma(t)^2}\right), \quad (3)$$

where $\mathbf{r}_c$ depicts the coordinates of the winner unit c, and $\mathbf{r}_i$ denotes coordinates of an arbitrary unit i. During learning, the learning rate and the width of the neighborhood function are decreased, typically in a linear fashion. The map then tends to converge to a stationary distribution, which approximates the density of the data.

One step of the training algorithm of the SOM is illustrated in Figure 1. The size of the SOM is 16 units, which have been arranged into a two-dimensional grid of 4 by 4 units. Data sample is marked with a cross; the black circles are the values of the prototype vectors before, and the gray circles after updating them towards the data sample. This kind of an update step is repeated iteratively during the training process.

There exists two freely available software packages that include implementation of the SOM: the SOM-PAK [15] and SOM Toolbox for Matlab [16].

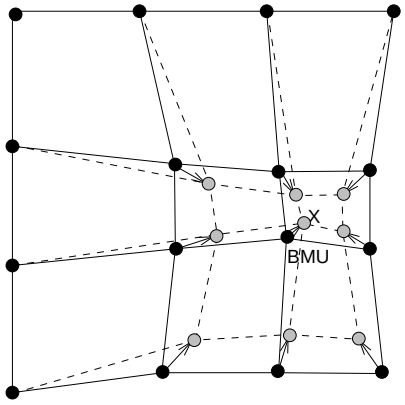

Figure 1: An illustration of the SOM training

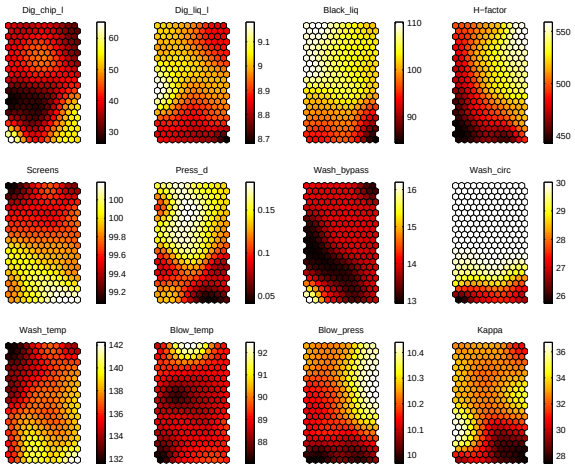

Figure 2: Component planes of the SOM depicting the operational states of a pulp digester

3 Applications

In the following, we illustrate the use of data-driven methods in analysis of both pulp mill and paper machine data. Also, we present two telecommunications applications: user pro£ling for fraud detection in mobile communications networks and mobile network monitoring.

In some other applications that are not discussed in detail here, we have used the SOM in the analysis of hot rolling processes in steel industry [17], in adaptive resource allocation in telecommunications [18] and in cluster analysis to extract and identify operational states of a paper machine [19].

3.1 Pulp digester fault analysis

In a pulp digester case study, the SOM was used in analysis of situations where the end product quality measurement value (kappa number) sank below the target value. Figure 2 contains one visualization that was used to analyze one situation of this kind. In this £gure, the component planes of a 22 by 12 units SOM trained using 12 digester measurements are shown. Each map unit corresponds to one process "state" prototype. Correspondingly, each component plane shows distribution of prototype vector values of a single variable on the map. Especially in the right bottom corner the value of the kappa number is low: the problematic states are located in that part of the map. It is immediately possible to make some visual observations of dependencies between measurements. When the kappa number ('Kappa') is low, values of black liquor ¤ow ('Black_liq') and pressure difference between bottom of the digester and blow line ('Press_d') are low. Also, values of the digester wood chip level ('Dig_chip_l') and temperature ratio between black liquor extraction screens ('Screens') as well as wash temperature ('Wash_temp') are high.

In the study, the SOM was successfully used with other data visualization techniques to reveal the reasons for the digester faults. The £nal explanation for the end quality variations was the fact that the normal production rate was too high. The digester was very sensitive to changes in the quality of the raw material and its processing before feeding it into the digester. Figure 2 shows how some most important digester measurements behave as a result of these variations. Complete results of the pulp digester analysis have been reported in [20].

3.2 Analysis of measurement signals of a paper machine

Often the measurement signals contain many data points, but actually quite little essential information. Time series segmentation methods can be used to £nd simpli£ed representations for the signals. In many cases, it is suf£cient to use a piecewise linear model: it can be used to £nd signal descriptions that are mixtures of impulses, sharp edges, ¤at zones and ramps. Figure 3 shows an example of segmentation of a process signal that consists of 200 points using a piecewise linear model that consists of 10 models.

The piecewise linear models are typically used as inputs to some pattern recognition method. In the following, we show an example where the piecewise linear representation is utilized in signal grouping [21]. A somewhat similar study was done in [22].

In Figure 4 (left), there are 18 signals that consist of 1000 points each. First, piecewise linear models that consisted of 50 segments (not shown) were computed for all signals. Then, dynamical time warping was used to compute the pairwise distances between signals. Then, based on the obtained distance matrix,

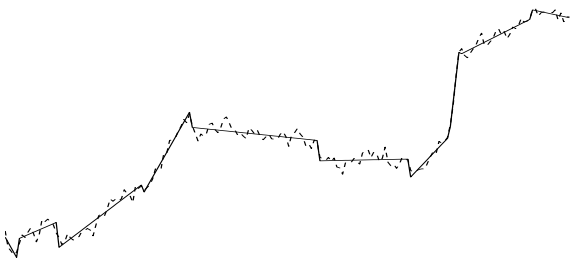

Figure 3: A signal (dashed line) with its piecewise linear approximation (solid line)

the signals were grouped using a clustering algorithm. The cluster hierarchy is shown in Figure 4.

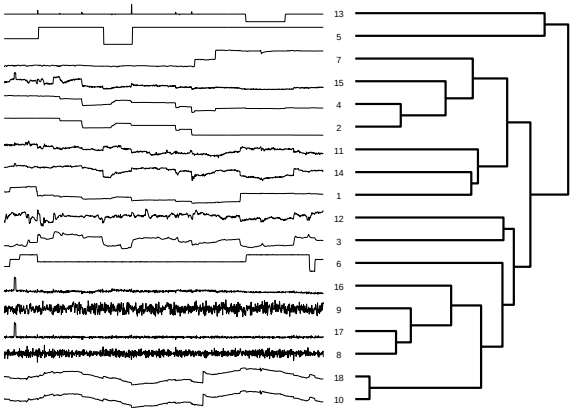

Figure 4: 18 signals (left) with their grouping tree (right). The height of each connection re¤ects the distance between two signals or signal groups

For example, grouping of signals can be utilized in analysis of process signals. Let us assume that there is a measurement with a sudden change whose origin is unknown. There may be dozens – or even hundreds – of measurements that could explain the change. The methodology above can ef£ciently be utilized to browse through all the signals and to bring out the ones with a similar waveform.

3.3 User pro£ling in telecommunications

The data-driven approach presented here may also be applied in other settings than in the analysis of industrial processes. To demonstrate this, we present a case study within telecommunications, where the purpose is to model the behavior of mobile phone users in order to detect fraudulent behavior based on the call data. This is closely related to anomaly detection within the industrial process framework. Anomalous states in the process would correspond to fraudulent behavior of mobile phone users, and process data would be analog to the logged call detail records resulting from making calls. Earlier work on process monitoring [9] can be seen as the groundwork leading to the problem of fraud detection, which may be seen as a *user* monitoring problem. User pro£ling and classi£cation for fraud detection is reported in [23, 24].

Fraud may be de£ned as the illegitimate use of a network and its services. Fraud detection aims at discovering fraudulent activity by whatever means necessary. A natural approach in the context of data analysis is to learn user pro£les from the call data and to use these user pro£les to infer on the fraudulent intentions of the users. In this fashion, the problem is largely solved in a similar manner as is done in the case of industrial process analysis.

A central issue in fraud detection is that we assume that the fraudulent intentions of the users are re¤ected in the call data. Conventionally, the calling activity is recorded for the purpose of billing in call detail records, which store call attributes like the identity of the subscriber, time of the call, duration of the call to mention a few. The user-speci£c call data forms a attributed series of transactions in time. Time is naturally in the center point for any modeling effort, since in user pro£ling one is interested in modeling the behavior, which is most naturally de£ned in a temporal context. By modeling the data, we model the fraudulent intentions of mobile phone users. Apart from fraud detection, user pro£ling efforts in telecommunications may be further motivated by the need to understand the behavior of customer to enable provision of matching services and to improve operations.

Two approaches to fraud detection may be used. In the differential approach, one may model the recent behavior in order to detect abrupt changes, which may be thought of as possible outbreaks for fraud. In the absolute approach, one may identify models of both normal and fraudulent behavior in order to determine the most probable mode of behavior. This latter approach requires the availability of samples from both classes. In the work reported here [24], we use samples from both classes — normal and fraudulent — but do not use the class information in the pro£ling part, but only in the visualization of the results. The data for one fraudulent user is illustrated in Figure 5. In this £gure, the time period on the horizontal axis is one week. The value of the call time series is one, if the user is making calls and zero if the phone is not used. The sampling time is one minute. The model identi£es dynamic transitions between the states of calling and no calling between time samples with a probabilistic model.

Since the model departs from the usual vector representation used in the data analysis context, we must extend the SOM algorithm by modifying the distance

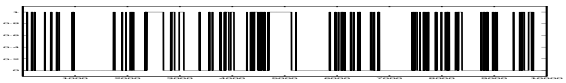

Figure 5: The data for one fraudulent user

measure between data samples and the map units. As explained in [2], we must de£ne an appropriate distance between the map units stored in the SOM, and also the corresponding update step, which is a gradient descent step of the error function in the neighborhood of the winner unit. For technical details and the derivation of this extension of the SOM algorithm, see [24].

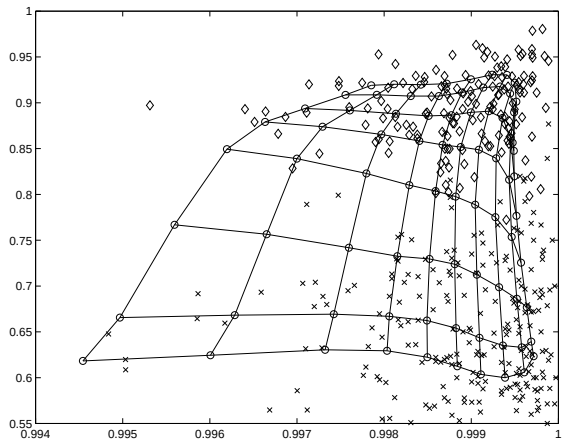

Figure 6: Self-Organizing Map of the mobile phone user pro£les

The user pro£les as illustrated in Figure 6 may be used to gain insight to the behavior of mobile phone users. The net-like structure is the map with the topologically ordered lattice, and the circles in the crossing represent the neurons, which store the parameters of the £rst-order Markov models to model the transitions between the states of the calling in time. The parameters calculated from the original time series for training data involving fraud users are shown with diamonds, normal users with cross marks. Parameters in the upper areas of the £gure correspond to long calls, while the parameters in the rightmost area correspond to infrequent calls. The fraudulent population seems to be characterized by longer calls than the normal population, whereas the frequency of the calling seems to roughly the same. One use of these models is to £ne-tune the models for better discrimination by taking advantage of the class information. Learning Vector Quantization has been used in this application [25].

3.4 Mobile network monitoring

SOM based methods have also been applied in monitoring the behavior of the Radio Access Network (RAN) of the third generation cellular systems [26]. The quality and capacity of the network are controled by hundreds of con£guration parameters. For an operator, the goal is to achieve the best possible Quality of Service (QoS) for various users. The network con£guration parameters should be optimized in order to obtain maximum capacity supporting different types of traf£c. Determining the optimal operation point for each cell in the network is the ultimate goal. In this application, the SOM has been used to monitor the operation of the RAN to £nd out the dependencies of the con£guration parameter settings of the network.

The radio network is a dynamic system which is very dif£cult to model exactly in all possible situations. The phenomena in the third generation RAN depend, e.g., on the type of service and on the number of users, their location, mobility etc. The important situations occur just on the acceptable limits of the QoS. Then, the operator should be able to predict the behavior of the network in order to control the system.

The SOM is used to classify the feature vectors describing the RAN into a £nite number of classes. Instead of mapping single static data points into one of the state clusters de£ned by the SOM statistical properties of the data within some time window were used. In this way, the essential information of the dynamic process was captured. The cluster properties are interpreted based on the knowledge of the physical network scenario.

Figure 7 shows the monitoring of some of the mobile cells of the network. Six different operational states of the mobile cells were de£ned corresponding to the clusters on the map. The dynamic behavior of a mobile cell can be followed by identifying the consecutive operational states on the clusters. This is visualized by a trajectory — a line drawn between consecutive BMUs in time – on the SOM. It can be seen from the £gure that some mobile cells are steadily staying in certain clusters while others are changing from one cluster to another. The three clusters on upper left part of the map correspond to situations with high error rate corresponding to undesired operation. For instance, the mobile cell 26 is all the time in the problematic state. Also the cells 20 and 21 are problematic. The cluster in the lower left corner represents small load and acceptable error rate. The cell 27 belongs to this group. The clusters on the right correspond to high error rate for short periods of time. For instance, the mobile cell 19 has this kind of behavior.

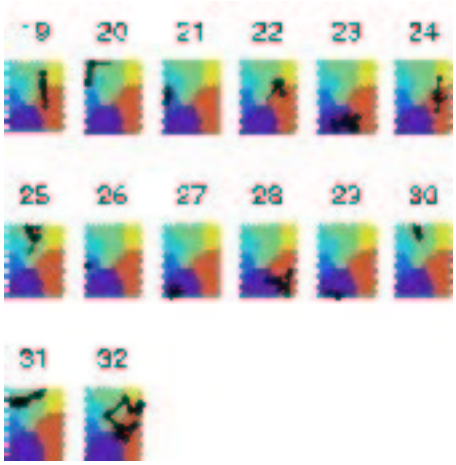

Figure 7: Monitoring cells in a mobile network

4 Summary and conclusions

Understanding of the process behavior in industrial setting and in telecommunications has been the main goal in investigating methods for building models from operational measurement data. This is especially helpful when systems are highly complex and when physical knowledge about process behavior is not available. We presented the SOM algorithm in some detail and some case studies demonstrating the effectiveness of the approach.

Acknowledgments

The work on fraud detection was funded by Siemens Corporate Technology in Munich, Germany, where the work was also carried out. The work on mobile network monitoring was carried out in cooperation with Nokia.

References

[1] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[2] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.

[3] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. Adaptive Computation and Machine Learning Series. MIT Press, 2001. To appear.

[4] Christopher Bishop. *Neural Networks in Pattern Recognition*. Oxford Press, 1996.

[5] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[6] Vladimir Cherkassky and Filip Mulier. *Learning from data: Concepts, Theory and Methods*. John Wiley & Sons, 1998.

[7] Samuel Kaski, Jari Kangas, and Teuvo Kohonen. Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1:102–350, 1998.

[8] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–84, 1996.

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

[10] Olli Simula, Juha Vesanto, Esa Alhoniemi, and Jaakko Hollmén. *Neuro-Fuzzy Tools and Techniques*, chapter Analysis and Modeling of Complex Systems Using the Self-Organizing Map, pages 3–22. Physica Verlag (Springer Verlag), 1999.

[11] Juha Vesanto and Jussi Ahola. Hunting for Correlations in Data Using the Self-Organizing Map. In Bothe et al. [27], pages 279–285.

[12] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, May 2000.

[13] Esa Alhoniemi, Johan Himberg, and Juha Vesanto. Probabilistic Measures for Responses of Self-Organizing Map Units. In Bothe et al. [27], pages 286–290.

[14] Juha Vesanto. SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.

[15] Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen, and Kari Torkkola. SOM_PAK: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996. `http://www.cis.hut.fi/research/som_pak/`.

[16] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. SOM Toolbox for Matlab 5. Report A57, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 2000. Software freely available via WWW at URL: `http://www.cis.hut.fi/projects/somtoolbox/`.

[17] L. Cser, A. S. Korhonen, P. Mäntylä, J. Ahola, and O. Simula. Improving the geometric quality parameters of hot rolling. In *Proceedings of International Conference on Quality Manufacturing*, 1999. Invited paper.

[18] Haitao Tang and Olli Simula. The optimal utilization of multi-service SCP. In Villy B. Iversen and Jorgen Norgaard, editors, *Intelligent Networks and New Technologies*, pages 175–188. Chapman & Hall, 1996.

[19] Maija Federley, Esa Alhoniemi, Mika Laitila, Mika Suojärvi, and Risto Ritala. State management for process monitoring, diagnostics and optimization. In *Control Systems 2000 Preprint*, pages 295–298, 2000.

[20] Esa Alhoniemi. Analysis of Pulping Data Using the Self-Organing Map. *Tappi Journal*, 83(7):66, July 2000. The paper is available in its entirety from TAPPI's web site at http://www.tappi.org/public/tappi_journal.asp.

[21] Esa Alhoniemi. Dynamic time warping in grouping and analysis of process data. In preparation.

[22] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping to massive datasets. In *Proceedings of 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases*, 1999.

[23] Jaakko Hollmén. *User Pro£ling and Classi£cation for Fraud Detection in Mobile Communications Networks*. PhD thesis, Helsinki University of Technology, 2000.

[24] Jaakko Hollmén, Volker Tresp, and Olli Simula. A self-organizing map algorithm for clustering probabilistic models. In *Proceedings of the Ninth International Conference on Arti£cial Neural Networks (ICANN'99)*, volume 2, pages 946–951. IEE, 1999.

[25] Jaakko Hollmén, Volker Tresp, and Olli Simula. Learning vector quantization algorithm for probabilistic models. In *Proceedings of EUSIPCO 2000 — X European Signal Processing Conference*, volume II, pages 721–724, 2000.

[26] Kimmo Raivio, Olli Simula, and Jaana Laiho. Neural analysis of mobile radio access network. In *IEEE International Conference on Data Mining*, November/December 2001. (to be published).

[27] H. Bothe, E. Oja, E. Massad, and C. Haefke, editors. *Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA '99)*. ICSC Academic Press, 1999.