# Chapter 2

# The Self-Organizing Map as a Tool in Knowledge Engineering

Johan Himberg, Jussi Ahola, Esa Alhoniemi,
Juha Vesanto, and Olli Simula
*Helsinki University of Technology*

## Abstract

The Self-Organizing Map (SOM) is one of the most popular neural network methods. It is a powerful tool in visualization and analysis of high-dimensional data in various engineering applications. The SOM maps the data on a two-dimensional grid which may be used as a base for various kinds of visual approaches for clustering, correlation and novelty detection. In this chapter, we present novel methods that enhance the SOM based visualization in correlation hunting and novelty detection. These methods are applied to two industrial case studies: analysis of hot rolling of steel and continuous pulp process. A research software for fast development of SOM based tools is briefly described.

*Keywords* : explorative data analysis, self-organizing map, visualization of multidimensional data, correlation detection, clustering, novelty detection, process analysis, hot rolling of steel, continuous pulp process

## 2.1 Introduction

Traditionally, modeling and control of industrial processes is based on analytic system models. The models may be built using knowledge based on physical phenomena and assumptions of the system behavior. However, many practical systems, e.g., industrial processes, are so complex that global models cannot be defined. In such cases, system modeling must be based on experimental data obtained by various measurements.

Modern automation systems produce large amounts of measurement data. However, interpretation of this data and the correlations between measurements and other system parameters is often difficult. In many practical situations, even minor knowledge about the characteristic behavior of the system might be useful. For this purpose, easy visualization of the data is of great help. The measurements need to be converted into some simple and comprehensive display which would reduce the dimensionality of measurements and simultaneously preserve the most important metric relationships between the data.

Artificial neural networks have successfully been used to build system models directly based on process data. They provide means to analyze the system or process without explicit physical model. The Self-Organizing Map (SOM) [12] is one of the most popular neural network models. Due to its unsupervised learning and topology preserving properties it has proven to be especially suitable in analysis of complex systems. The SOM algorithm implements a nonlinear topology preserving mapping from a high-dimensional input data space onto a two-dimensional network or grid of neurons. The network roughly approximates the probability density function of the data and, thus, inherently clusters the data. Various visualization alternatives of the SOM are useful, e.g., in searching for correlations between measurements and in investigating the cluster structure of the data.

SOM based data exploration has been applied in various engineering applications such as pattern recognition, text and image analysis, financial data analysis, process monitoring and modeling as well as control and fault diagnosis [15; 19]. In addition, the SOM has been used in analysis and monitoring of telecommunications systems. Applications include equalizer structures for discrete-signal detection and adaptive resource allocation in telecommunications networks.

The ordered signal mapping property of the SOM algorithm has proven to be powerful in analysis of complex industrial systems and processes. The SOM allows easy visualization of system parameters and their correlations, cluster structure of the data, monitoring of operation state, and novelty detection. The SOM based approach has, for instance, been utilized to determine the reasons for situations where the output quality of an industrial process is not satisfactory. The case studies presented in this chapter include analysis of pulping and steel rolling processes.

The SOM can be used in many different ways for data visualization and exploration. In this chapter, we present SOM based tools for data exploration in practical industrial applications. Novel methods to enhance the SOM based visualization in correlation detection, cluster analysis, and operation monitoring as well as novelty detection will be discussed.

## 2.2 Data analysis using the Self-Organizing Map

Our approach to data analysis is explorative and will concentrate on visualization based approaches. The main idea is to provide an overall picture of the data and create tools that help the analyst to *see* what the data are like and get ideas for further, perhaps more quantitative descriptions of the data.

### 2.2.1 *The Self-Organizing Map*

A SOM is formed of units located on a regular low-dimensional grid (usually 1D or 2D to enable visualization). The lattice of the grid can be hexagonal or rectangular. The former is often used because it is more pleasing to the eye.

Each unit $i$ of the SOM is represented by an $n$-dimensional prototype vector $\mathbf{m}_i = [m_{i1}, \ldots, m_{in}]$, where $n$ is equal to the dimension of the input space. On each training step, a data sample $\mathbf{x}$ is selected and the prototype vector $\mathbf{m}_c$ closest to it, the winner unit, is found from the map. The prototype vectors of the winner unit and its neighbors on the grid are moved towards the sample vector:

$$\mathbf{m}_i := \mathbf{m}_i + \alpha(t)h_{ci}(t)(\mathbf{x} - \mathbf{m}_i), \qquad (1)$$

where $\alpha(t)$ is the learning rate and $h_{ci}(t)$ is a neighborhood kernel centered on the winner unit $c$. Both learning rate and neighborhood kernel radius decrease monotonically with time. During the iterative training, the SOM behaves like a flexible net that folds onto the "cloud" formed by input data.

### 2.2.2 *Data analysis scheme*

Using the SOM in data analysis is only one part of a multi-staged process. The map — as any method — is a fruitful tool only if the input data really describe the essential phenomena and is not governed by completely erroneous data. The phases of a basic explorative data analysis process using the SOM can be sketched as follows:

*Data acquisition* may be real time measurement collection (on-line) or database query (off-line) which is usually the case when an exploratory analysis is made.

*Data preprocessing, selection and segmentation* are usually elaborate tasks involving a lot of *a priori* knowledge. Erroneous raw data have to be removed. Proper data scaling and representational transformations (e.g., symbolic to numerical values) have to be considered. Clearly inhomogeneous data sets may have to be divided to disjoint subsets according to some criteria in order to avoid problems which would come up if a global model was applied.

*Feature extraction* is the phase where preprocessed and segmented data are transformed into feature data vectors. It is important to realize that the objective in our case is to interpret the data and extract knowledge from it and from relations in it — not to make black-box classification or regression. Therefore, the feature variables have to describe the important phenomena in the data in such a way that they are clear in the analysis. It is evident that this and the previous stages cannot be properly done without knowledge of application domain.

*Training* of the SOM is performed according to the Sec. 2.2.1. The training parameters need to be determined. Fortunately, the basic SOM algorithm seems to be rather robust in this sense, and by following certain basic guidelines (see, e.g., [12]) satisfactory results are usually obtained. However, one delicate issue is the scaling of the feature variables. The variables with large relative variance tend to dominate the map organization. In order to equalize the contribution of individual variables in the map organization, they are usually normalized to be equivariant. The distance measure used in the SOM training has to be chosen in such a way that applying it to data makes sense. Usually, the Euclidean distance is used. The variable normalization and the distance measure are, of course, data dependent issues and related to the feature extraction phase.

*Visualization and interpretation* are the key issues for using SOM in data analysis. These include correlation detection, cluster analysis and novelty detection. The scope of this chapter is on the visualization and interpretation phase which are described in the next section. We remind that the data analysis is usually not a flow-through process, but requires iteration, especially between feature extraction and interpretation phases. An integrated software environment is clearly needed. We describe our research software in Sec. 2.4.

## 2.3 Visualization

The SOM provides a low-dimensional map of the data space. The aim of visualization is both to understand the mapped area and to enable investigation of new data samples with respect to it.

To understand what the SOM really shows, it is important to understand that it actually performs two tasks: vector quantization and vector projection. Vector quantization creates from the original data a smaller, but still representative, data set to be worked with. The set of prototype vectors reflects the properties of the data space. The projection performed by the SOM is nonlinear and restricted to a regular grid (the map grid). The SOM tries to preserve the topology of the data space rather than relative distances.

In contrast, there are several other ways of projecting multidimensional data to lower dimensions. A well-known method is based on Principal Component Analysis (PCA): the eigenvectors with the largest eigenvalues are calculated from the data set, and the data samples are projected on the subspace spanned by these vectors. This is a fast linear operation, but gives misleading results if the ignored directions have significant information.

A different approach is to project the data so that relative distances between samples are as close to the original as possible according to some cost function. Different cost functions lead to different nonlinear algorithms, e.g., Sammon's projection [18] or the Curvilinear Component Analysis (CCA) [4]. Large data sets cause often problems for these, usually iterative, projection methods as the procedure becomes computationally heavy. One possibility is to reduce the computational task by first quantizing the data using some suitable method, e.g., k-means and then applying the projection method. Some recent solutions include [17]. Of course, the SOM can be seen as doing something similar, except that only topology — not distances — is preserved.

### 2.3.1 *Basic methods for SOM visualization*

The SOM grid provides a basis for various visualizations. Variable values or other features may be shown with respect to the grid.

#### (a) Unified distance matrix

The unified distance matrix (u-matrix) [7; 22] is a simple and effective tool to show the possible cluster structure on a SOM grid visualization. It shows the distances between neighboring units using a gray scale representation on the map grid. This gives an impression of "mountains" (long distances) which divide the map into "fields" (dense parts, i.e., clusters). See Fig. 2.1(c).

#### (b) Component planes

The SOM is often "sliced" into component planes in order to see how the values of a certain variable (component) varies on different locations of the map [20]. Each plane represents the value of one variable (component) of the prototype vector in each node of the SOM using, e.g., gray scale representation. One can now see the general behavior of the variable values in different parts of the SOM. See Fig. 2.1(c).

The component planes play an important role in the correlation detection: by comparing these planes even partially correlating variables may be detected by visual inspection — a simple enhancement to this is described in the next section. This kind of comparison could be done using scatter plots as well, but this would require a quadratic amount of displays with respect to the number of variables: each variable against each other variable. When using the component planes the number of displays grows linearly. Furthermore, the vector quantization performed by the SOM removes noise. The component planes can also be easily compared with the cluster representation of the u-matrix.

### (c) Hits

When investigating new data with the SOM the question is, which part of the map best corresponds to the data? Traditionally, this has been answered by finding the nearest prototype vector (the best matching unit, BMU) for each investigated data sample and then indicating it from the SOM. See Fig. 2.1(c). For multiple data vectors, one can count the number of times that each unit has been the BMU, and thus, a data histogram is obtained. By comparing different histograms, one can evaluate the similarity of different data sets in terms of the map. Similar histograms imply similar data sets.

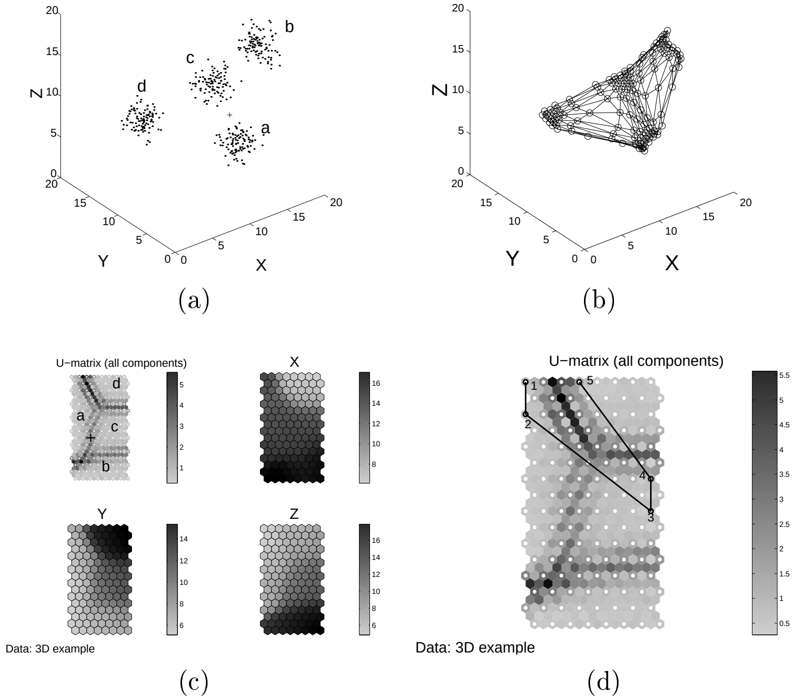
### (d) Trajectories

If the data have been acquired from a process, one may be interested in visualizing the evolution of the process state in time. The BMU of the current feature vector may be regarded as the operating point on the map which in turn can be regarded as a projection of the multidimensional state space. Trajectory (Fig. 2.1(d)) is a line connecting a sequence of these operating points [10; 21] that shows the change of the process in time. A software tool related to this issue is presented in Sec. 2.4.3.

### (e) Combining different projections

To get an idea of the shape of the map in the data space, the prototype vectors of the SOM can be projected to a low dimension using some vector projection method which tries to preserve distances between projected points. A common practice is to use Sammon's projection and to show the topological relations of the map by connecting points that corresponds to neighboring units. The SOM may be considered unreliable if the topological structure is completely twisted or folded. In Fig. 2.1(b) one can see that this has not happened in our artificial example but the map is well ordered in this case.

In order to clarify the connections between visualizations, they may be linked together using color which is a dominant visual hint for grouping objects. This idea has been applied to carrying information from the SOM representation to a geographical map in [1; 3; 8]. We have applied this idea simply to link different presentations of the same data together, e.g., the SOM grid and a scatter plot or Sammon's projection [5; 23]. See Fig. 2.2. Similar linking idea to PCA has been earlier presented by Aristide [3].



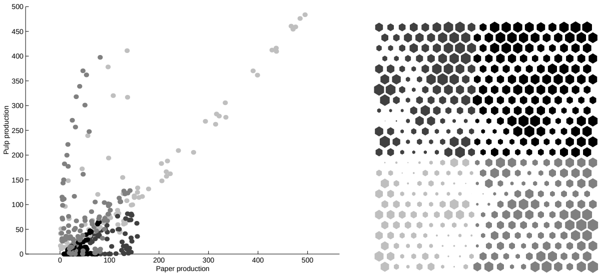
(a) (b)

(c) (d)

Fig. 2.1 Figure (a) shows a simple three-dimensional (variables X,Y and Z) artificial data set with four clear clusters (a,b,c,d). Figure (b) shows the prototype vectors ("o") of a SOM trained with the data in (a). The topological connections are shown as lines connecting the neighboring prototype vectors. Figure (c) shows the u-matrix and component planes. In the u-matrix dark gray represents long inter-unit distances and light gray short ones. The clusters — that can be seen as light "fields" between the dark "mountains" — have been labeled for convenience. The +-sign shows the BMU for the sample marked by +-sign in (a) (located between clusters a and c). The component planes show how the variables X,Y and Z vary along the map. Figure (d) shows a trajectory of five samples on the u-matrix.

### 2.3.2 *Correlation hunting*

Correlations between component pairs are revealed as similar patterns in identical positions of the component planes. The correlation detection can be made easier if the component planes are reorganized so that the possibly correlated ones are presented near each other [24]. See Fig. 2.3. Using component planes for correlation hunting in this way is easy, but also rather vague and sometimes even misleading. However, it is easy to select interesting component combinations for further investigation. A more detailed study of interesting combinations can be done using scatter plots which can be linked to the map units by color as has been regularly done in the case studies in Sec. 2.5.*



(a) (b)

Fig. 2.2 Correlations between vector components can be efficiently visualized using scatter plots. In Fig. (a) each dot corresponds to one map unit. The $x$- and $y$-coordinates of the dots have been taken from two components of the prototype vectors. To link the scatter plot to other visualizations, each dot is given a color according to the color coding of the map units shown in Fig. (b). In this grayscale figure only four shades of gray are used. In practice a full color palette is much more informative. In addition to color coding, Fig. (b) also uses size to indicate clusters on the map: small units correspond to cluster borders. It can be seen that for most units, especially those with light gray color coding, the two components are linearly correlated but that there are distinct exceptions.

### 2.3.3 *Novelty detection*

When investigating new data using SOM, the BMU of each data sample is found and indicated on the map (see Sec. 2.3.1). The problem with this simple approach is that it gives no information of the accuracy of the match. Typically, there are several units with almost as good match as the BMU. Alternatively, the data sample may actually be very far from the map — a novelty in terms of the map.

Instead of simply pointing out the BMU, the response of all map units to the data can be shown. The resulting response surface shows the relative goodness of each map unit in representing the data. The response can be,

*For technical reasons we can't use colors in this presentation. In order to sketch the idea, a gray level coding is used instead. A full color version can be found in [27].

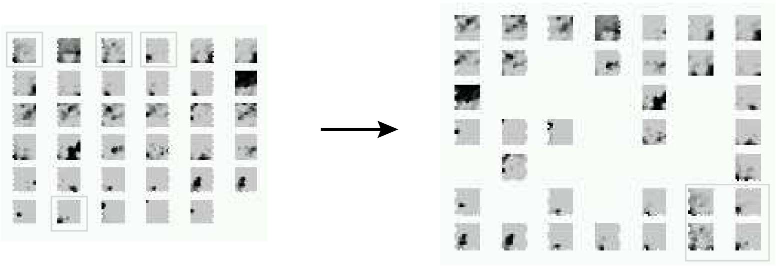


Fig. 2.3 Correlations between components can be hunted from the component planes visualization on the left. The task is easier if the planes are reorganized so that component planes which seem to have high correlation are placed near each other, as shown on the right. For example, this reorganization brings nicely together the four framed components.

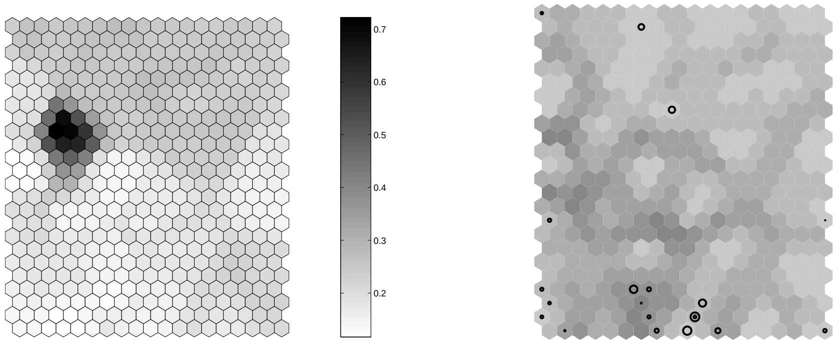
e.g., a function of the quantization error as follows:

$$g(\mathbf{x}, \mathbf{m}_i) = \frac{1}{1 + (q_i/a)^2}, \qquad (2)$$

where $q_i = ||\mathbf{x} - \mathbf{m}_i||$ is the quantization error, i.e., distance, between sample $\mathbf{x}$ and map unit $i$. The scaling factor $a$ is the average distance between each training data sample and its BMU. See Fig. 2.4(a). Perhaps a more interpretative response function results if the SOM is used as a basis for reduced kernel density estimate of the data. Then one can estimate the probability $P(i|\mathbf{x})$ of each map unit representing the data sample, see for example [2; 6].

In both cases above, the response surface is added onto the map afterwards, while the original SOM algorithm has a "crisp" winner-take-all activation function. There are related algorithms that have an intrinsic probabilistic background as the S-Map [11]. However, it seems that a kernel density estimation model added to the SOM gives results that are well comparable with these methods [2].

Another way to show the accuracy of the match is to use, e.g., the size of the sample marker. In Fig. 2.4(b), the fuzzy response function (Eq. 2) has been used to control the size of the sample markers (circles). Now, individual samples can be seen along with their BMUs (position) and accuracy (size).



(a) (b)

Fig. 2.4 Accuracy of matches. Figure (a) shows the response surface (Eq. 2) for one data sample. Figure (b) shows BMUs and corresponding accuracies of 20 samples. The background texture is averaged u-matrix of the SOM. Each circle represents one sample. The position of the circle indicates the BMU, and its size the accuracy of the match.

## 2.4 Software

To accomplish the explorative and iterative data analysis scheme, a flexible software environment is needed. It should include domain specific post- and preprocessing capabilities, SOM implementation and different visualizations. The possibility to rapidly customize the code is important. We have tried to achieve this in the SOM Toolbox.

### 2.4.1 *SOM Toolbox*

The MathWorks Inc.'s MATLAB [16] has been gaining popularity as the "language of scientific computing", and it employs a high-level programming language with strong support for matrix algebra, graphics and visualization. MATLAB suits for fast prototyping and customizing. The SOM Toolbox[†] [25], hereafter the Toolbox, is an attempt to take advantage of these strengths and provide a customizable and easy-to-use implementation of the SOM as a free function library for the MATLAB environment.

The advantages of the Toolbox are mainly in fast customization and visualization. A major benefit is that as the MATLAB's language is inter-

[†] Available in `http://www.cis.hut.fi/projects/somtoolbox/`

preted, the user may give on-line commands to change various parameters or visualizations. Furthermore, the Toolbox is constructed in a modular manner. Therefore, it is convenient to tailor the code for the specific needs of each user. Other toolboxes — commercial or freeware — may be used together with the Toolbox to provide domain specific processing capabilities. For example, a toolbox related to system simulation might be used in a process control task.

The basic procedures — SOM initialization, training and visualization — have been collected under high level functions which provide heuristic choices for various parameter values. This gives an automated data-to-visualization operation to start with. The Toolbox also implements some variants of the basic SOM. The topology of the SOM can be $n$-dimensional, and several SOM shapes are supported: rectangular, cylinder and toroid — as well as several neighborhood functions. In order to facilitate the data analysis process, the Toolbox keeps track of labels associated with individual data vectors, vector component names, component normalization information and information on the training procedure.

A standard implementation of the SOM and related tools are available as the SOM_PAK [13]. It is a public domain software package[‡] developed in the Neural Networks Research Centre of the Helsinki University of Technology, written in ANSI C language for UNIX and PC environments. In map training, it is faster than the Toolbox and has a better capability to be applied to large data sets than the Toolbox. However, while the SOM_PAK is the choice for heavy duty, the Toolbox is meant for experimental and/or interactive purposes. If the scalability is a problem, the SOM_PAK can be accessed from the Toolbox. It is possible to first train the map with the SOM_PAK and then use the Toolbox for visualization.

### 2.4.2 *The SOM visualization as a user interface platform*

The SOM grid is an effective base for building visualizations and user interfaces for accessing multidimensional data. Assume that we need to attach some information (text, symbols, colors) to the projected points. The projection methods that produce a nonuniform visualization may cause problems as the labeling information easily becomes unreadable in the dense parts of the projection.

[‡] Available in `http://www.cis.hut.fi/nnrc/som_pak/`

In the SOM the amount of the units in a certain region of the space is proportional to the density of the training data in that region, i.e., the map uses more units to represent the dense parts of the data. This increases readability as the map automatically "zooms up" areas that are dense. On the other hand, the topology preserving property gives access to cluster or variable value visualization through u-matrix and component planes which can be easily used as browsers. The nodes can be used as clicking points to access the data underneath. The idea to use the SOM visualization as a user interface has been used earlier, e.g., in the WEBSOM [14] in browsing large document collections.

### 2.4.3 *Interactive tool for time-series exploration*

As an example, we shortly describe an interactive time-series tool designed on the SOM Toolbox. The purpose of the tool is to facilitate the inspection of the connections between the multidimensional data space presented by the SOM and the time-series plot. In analysis, the feature data have been extracted and the map is trained using them. The analyst may now evaluate how certain feature variables are distributed and what kind of clusters there are in the map visualization. The analyst sees how different regions of the map are related to a time-series representing the same data from a different point of view. After this, the analyst may reconsider if the feature data really represent the investigated phenomena in a sensible way or if the features should be extracted in some other way. The tool in Fig. 2.5 allows the analyst to

- see the connection between original time-series and the feature space visualized by the SOM.
- run the process using a slider on the time-series. A trajectory — showing the connected BMUs for the current and some past samples of the time-series — is animated on the map.
- define some areas on the map and tag them with specific colors. The same markers are shown on the time-series. Now the analyst may inspect how a region on the map is connected to the time-series. This may be done to the opposite direction, too, in order to see how the time-series is projected to the map.

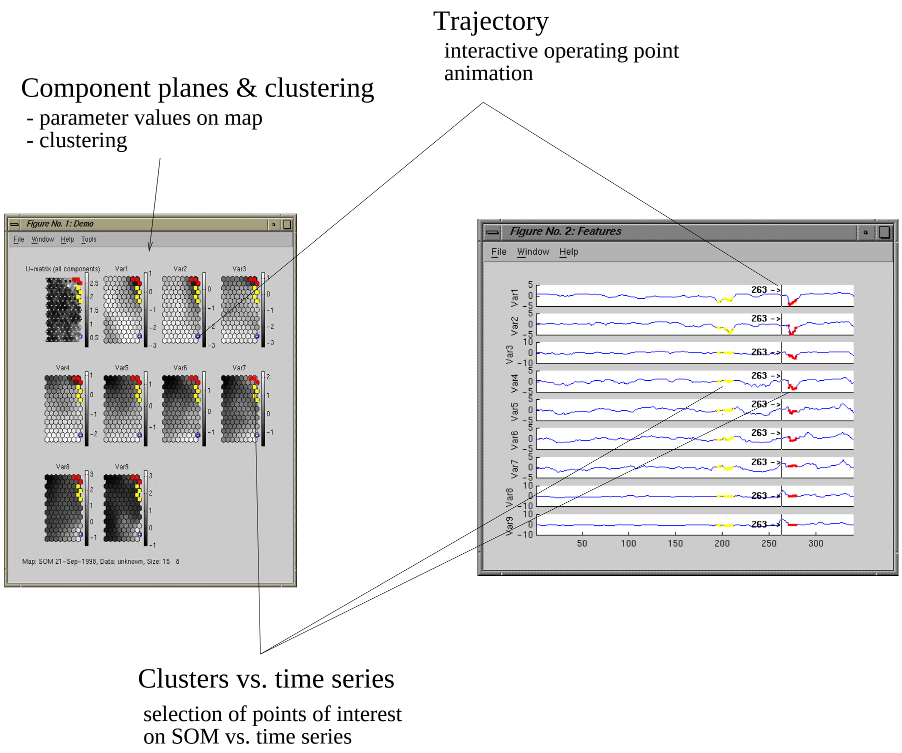


Fig. 2.5 Time-series tool. The analyst may inspect changes in the operation point using a slider. The analyst may mark regions on the map and in the time-series using different colors.

## 2.5 Case studies

### 2.5.1 *Analysis of a continuous pulp digester*

In the first case study, behavior of a continuous pulp digester was analyzed. An illustration of the digester and separate impregnation vessel is shown in Fig. 2.6. Wood chips and cooking liquor are fed into the impregnation vessel. After the impregnation, the chips are fed into the digester. At the top of the digester, they are heated to cooking temperature using steam, and the pulping reaction starts. During the cook, the chips slowly move downwards the digester. The cooking ends at extraction screens, where the pulping reaction is stopped by cooling the chips using wash liquor. The wash liquor is fed to the digester bottom and it moves upwards, countercurrent to the chip flow.

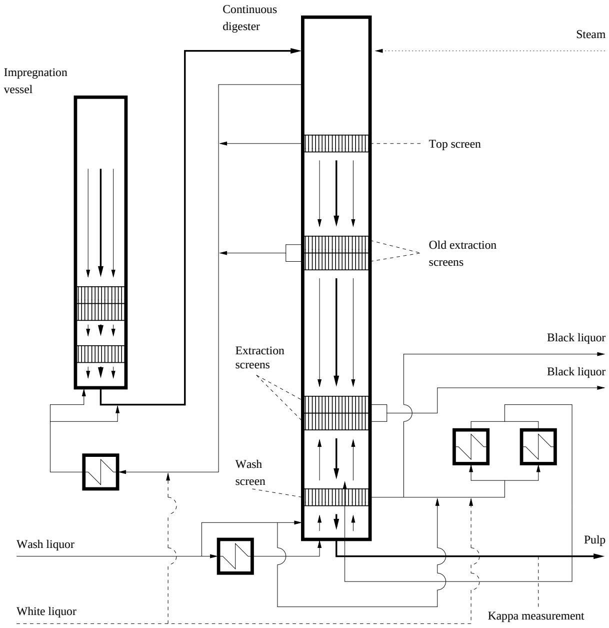


Fig. 2.6 The continuous digester and the impregnation vessel. The cooking and wash liquor flows are marked by thin lines and the chip flow by thick line.

Problems in digester operation indicated by drops of pulp consistency in the digester outlet were the starting point for the analysis. In those situations, end product quality variable (kappa number) values were lower than the target value.

Measurement data were obtained from the automation system of the mill. The analysis was started with several dozens of variables which were gradually reduced down to six most important measurements during data analysis process. The data used in the following experiments consisted of three separate measurement periods during more than one month of normal pulping operation. The periods were segmented by hand in such a way that they mainly consisted of faulty situations of the process. The production speed was required to be constant. During the measurement periods there were no significant errors in the measurements. Process delays between signals were compensated using known digester delays.

In Fig. 2.7, the six signals and production speed of the fiber line are shown. The three segmented parts are shown by solid line and the parts that were left out of the analysis by dotted line. In Fig. 2.8, the component planes of a 17 by 12 units SOM trained using signals of Fig. 2.7 are presented. Five of them depict behavior of the digester and the last one is the output variable, the kappa number. The problematic process states are mapped to the top left corner of the SOM: the model vectors in that part of the map have too low kappa number value.

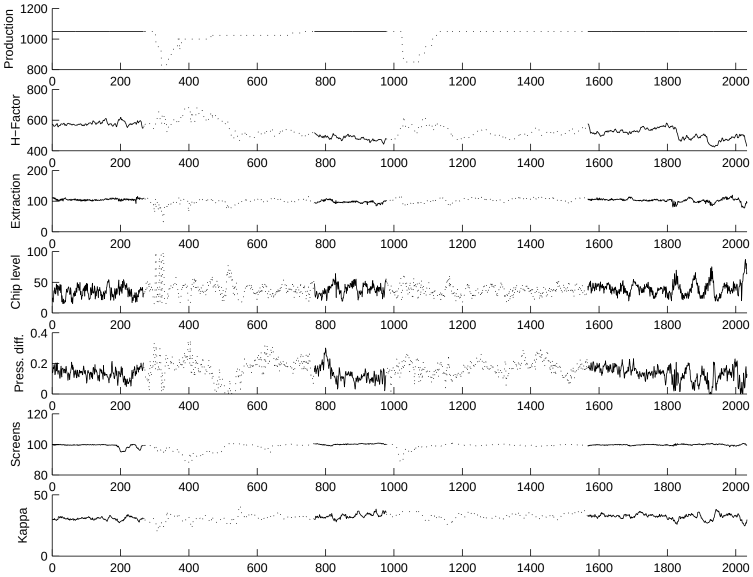


Fig. 2.7 Measurement signals of the continuous digester. The analyzed parts are marked by solid line and the parts that were ignored by dotted line.

Correlations between the kappa number and other variables are shown in Fig. 2.9, where the SOM of Fig. 2.8 has been presented using color coding. The colors were originally chosen in such a way that adjacent map units had almost similar colors; here we are only able to use four gray levels. The five scatter plots are based on *model vector component values* of the SOM. They all have the values of kappa number on the x-axes and the five other

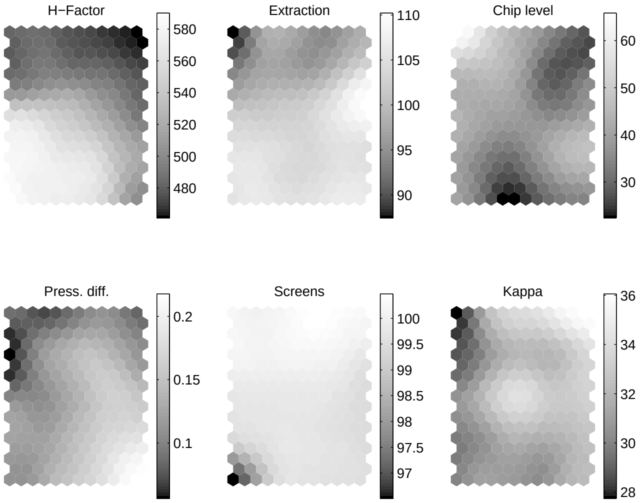


Fig. 2.8 Component planes of the SOM trained using six measurement signals of the digester. Dark color indicates low and light color high variable value, respectively.

variables on the y-axes. In Fig. 2.10, a similar technique for coloring the scatter plots is utilized. In this case, however, the scatter plots are based on *data vectors* — not values of the model vectors of the SOM. The color of each data vector is the one assigned to the the SOM unit that is nearest to the data vector. It should be noted that even though the plots differ from the ones of Fig. 2.9, the SOM has been able to capture the shape of the data cloud quite accurately. The scatter plots indicate that *in the faulty states* denoted by dark grey color (top left corner of the map), there is only weak correlation between kappa number and H-Factor, which is the variable used to control the kappa number. Otherwise, there is a negative correlation as might be expected. On the other hand, the variables *Extraction* and *Chip level* seem to correlate with the kappa number in the faulty process states. Also, the values of *Press. diff.* are low and value of variable *Screens* (which during the analysis was noticed to indicate digester fault sensitivity) is high.

The interpretation of the results is that in a faulty situation, the downward movement of the chip plug in the digester slows down. The plug is so tightly packed at the extraction screens that the wash liquor cannot pass it as it should. There are two consequences: the wash liquor slows down the

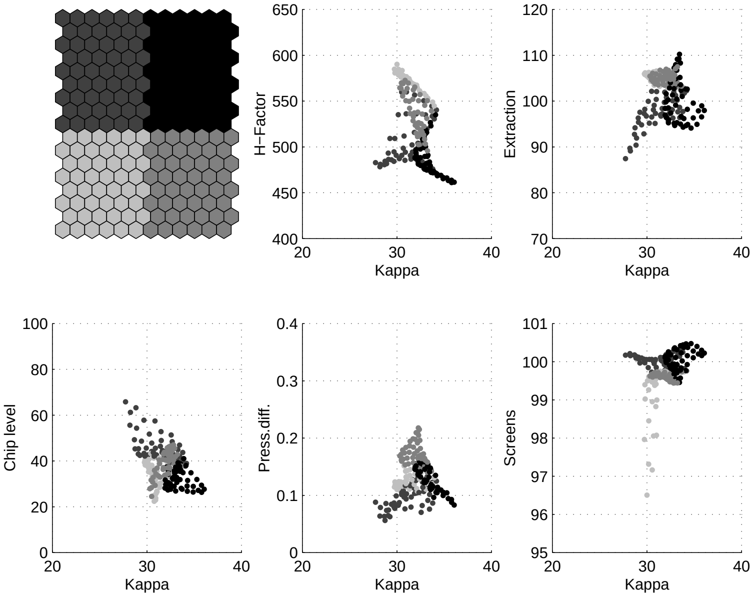


Fig. 2.9 Color map and five scatter plots of model vectors of the SOM. The points have been dyed using the corresponding map unit colors.

downward movement of the plug and the pulping reaction does not stop. Because the cooking continues, the kappa number becomes too small. In addition, the H-factor based digester control fails: in the H-factor computation, cooking time is assumed to be constant, while in reality it becomes longer due to slowing down of the chip plug movement.

### 2.5.2 *Analysis of the quality of the hot rolled strip*

In the second case study, a hot rolling system was analyzed. Hot rolling is a process where steel slabs are heated, rolled, cooled and coiled into final products, strips. Figure 2.11 illustrates the composition of the hot strip mill in Raahe (at the time of the data acquisition; currently the mill construction is somewhat different). First, the slab is heated in the slab reheating furnaces (1) into temperature appropriate for the following rolling process. Then, after the formed scale is removed with high-pressure water

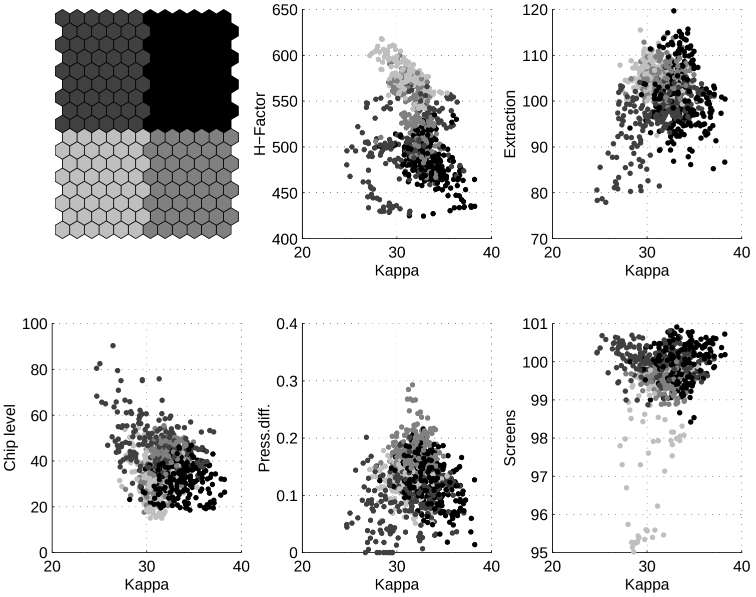


Fig. 2.10 Color map and five scatter plots of data vectors. The points have been dyed using the color of BMU.

shower (2), the slab passes to the roughing mill. The slab is rolled back and forth several times vertically in the edger (3) and horizontally in the reversing rougher (4). The resulting transfer bar travels under the heat retention panels (5) through another descaling and possible shearing of the head (6) into the finishing mill (7), where it is rolled into desired end product. The finishing mill consists of six stands. The transfer bar goes through them with high accelerating speed. After the rolling, the strip is cooled with several water curtains (8) and coiled (9).

The process is controlled hierarchically by several separate automation systems. Basically, each process stage introduced above has its own automation system. Furthermore, a lot of additional computation, control, and information processing is made within and between the systems. This causes difficulties in the data acquisition. Hence, the process data available for this case study consisted only of averages and standard deviations of the measured process variables of one strip.

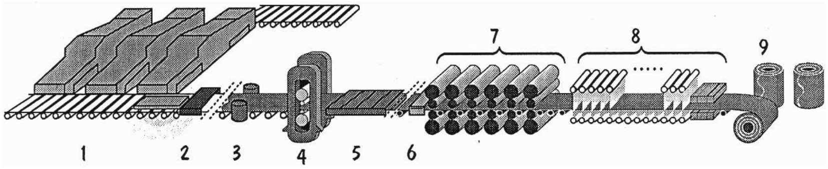


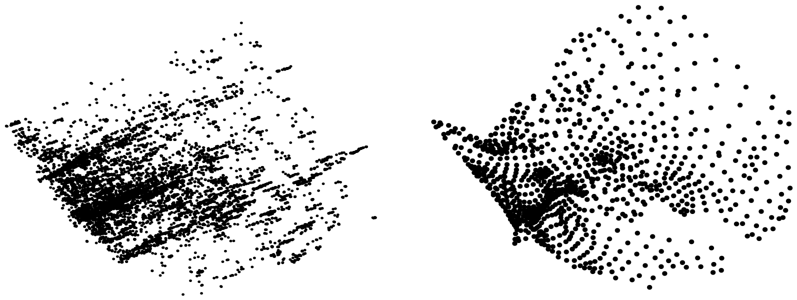
Fig. 2.11 Rautaruukki hot strip mill. The different process stages are marked with numbers. See text for their explanations.

Due to ever increasing competition and customer requirements, the steel producers are under growing pressure to improve the cost efficiency of the production and the quality of their products. This is also the motivation for the analysis, the purpose of which was to study which process parameters and variables affect the quality of the rolled strips. This can be done, e.g., with correlation analysis for process data, which was the approach in this case.

The data was collected from factory data bases in co-operation with the process experts. In the data set it was chosen 47 variables. The average and standard deviation of five process parameters were chosen to represent the quality: width, thickness, profile, flatness and wedge of the rolled strip. The other variables included information about the slab (analyzed chemical content), finishing mill parameters (average bending forces, entry tensions, and axial shifts for each stand), and process state (strip strength, target dimensions, and average and standard deviation of the temperature after the last stand). After preprocessing of data, the amount of the strips included in the study was slightly over 16500.

In the beginning, in order to get to know the general dependencies between the parameters, a very simple global linear correlation analysis was performed. This showed, e.g., that the entry tensions of the stands were controlled based on the tensile strength calculated from the chemical analysis results. Due to redundant information of the variables caused by the controlling principles of the process, the data dimension could be reduced to 36 variables.

The structure of the data set was then studied. This was done by projecting the data on the two largest principal components of the data (Fig. 2.12(a)). As an alternative approach, the prototype vectors of a SOM trained with the data were projected with Sammon's mapping (Fig. 2.12 (b)). The data seem to be somehow clustered as was expected. Furthermore, it can be seen that the different projection algorithms provide more or less similar results and the SOM has approximated the data quite well.


(a) PCA

(b) Sammon

Fig. 2.12 The original data projected with PCA (a) and the prototype vectors of the SOM projected with Sammon's mapping (b).

Due to the quite large amount of variables, finding correlations between them using the typical component plane representation (where the planes are plotted next to each other in the same order as the variables in the data) became extremely difficult. Fortunately, the task could be made easier by reorganizing the component planes using the procedure explained in Sec. 2.3.2 so that the possibly correlating planes were placed near each other. The result is illustrated in Fig. 2.13.

Using this approach, some of the interesting relationships between the variables could be detected. Based on this information and the *a priori* knowledge of the system, the variables to be used in the more detailed analysis of the strip quality could be chosen. In this case, the strip thickness was chosen to be studied further. The variables included in the new data set were quality parameters, thickness average deviation and standard deviation, strip target dimensions, strip strength, bending forces, temperature after the last stand, and strip profile.

Using the scatter plots colored with the continuous coloring of the SOM plane, as explained in Sec. 2.3.2, dependencies between thickness and other parameters in different process states could be found. The approach is illustrated in Fig. 2.14, where all the other variables are plotted against average thickness deviation. Note, that here the color code had to be limited

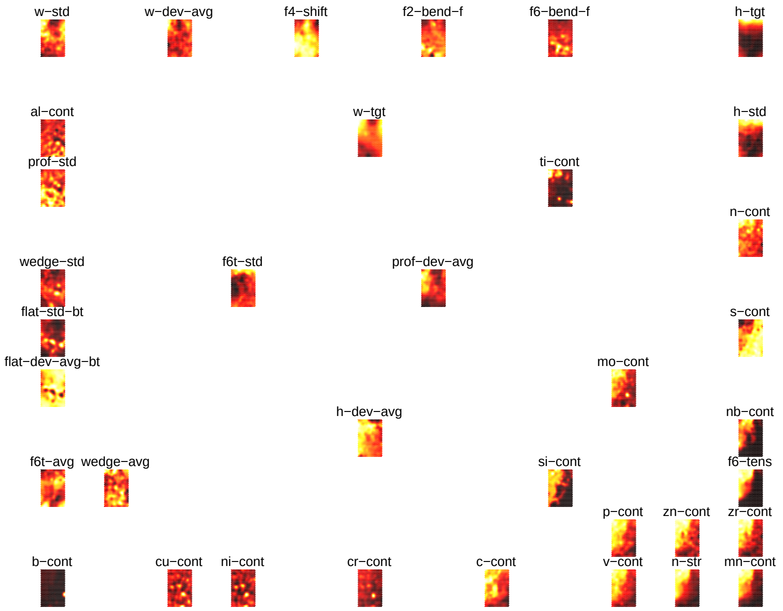
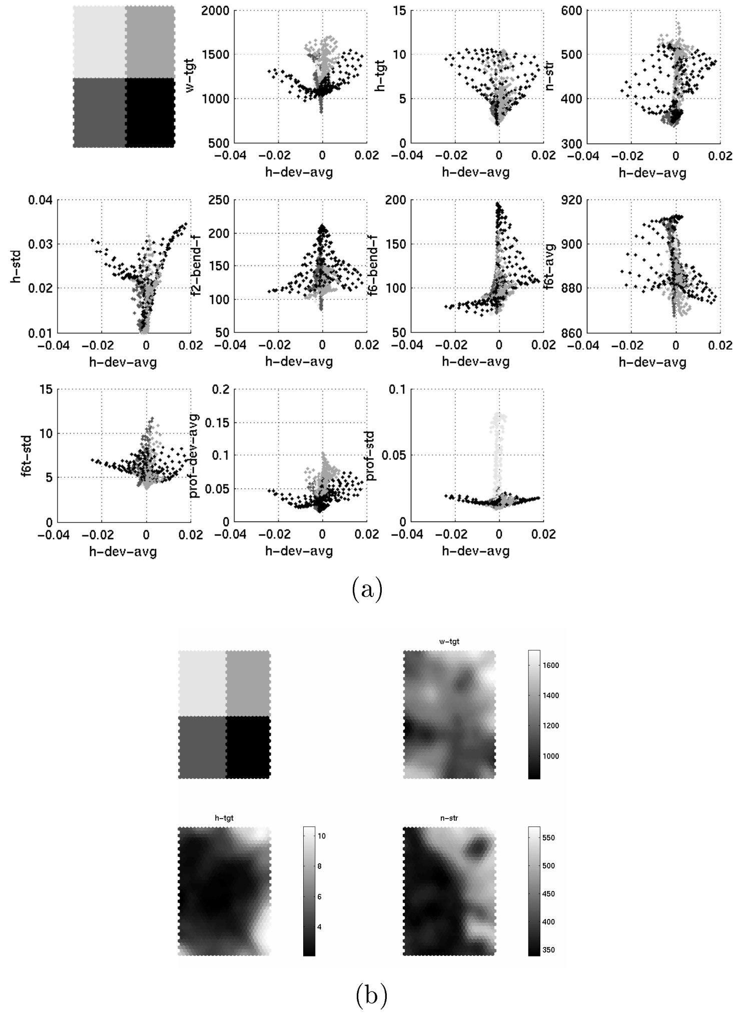


Fig. 2.13 The reorganized component planes of the SOM.

to four gray levels, which drastically deteriorates the results. However, in the actual study a true continuous color code was used. After some inspection of these plots, the following statements regarding the problems with strip width could be made:

- The thickness deviation of the strip seems to increase as the bending forces decrease, especially when the strips are somewhat thick. Then, also the standard deviations of the thickness, the temperature after the last stand, and the strip profile tend to increase.
- The standard deviation of the strip seems to increase as the thickness of the strip increases, especially with hard steels. As with the deviation of the thickness, the standard deviation seems to increase as the rolling temperature and the bending forces decrease. The standard deviation of the temperature after the last stand and the strip profile tend also to increase. However, this does not hold for quite narrow and thin strips.



(a)

(b)

Fig. 2.14 In Fig. (a) variables from the prototype vectors of SOM are scatter plotted using the color coding shown in the upper left picture. For example, the last scatter plot (*prof-std* vs. *h-dev-avg*) shows that on the lightest gray region of the color coded map the thickness deviation (*h-dev-avg*) does not increase/decrease, as on the other regions, when the profile standard deviation (*prof-std*) increases. In Fig. (b), it can be seen that on the lightest gray region of the map are the data samples mostly from quite narrow, thin, and mild strips, as on this region the component planes *w-tgt* (target width), *h-tgt* (target thickness), and *n-str* (strip strength) indicate low values simultaneously.

## 2.6 Conclusions

The Self-Organizing Map has proven to be a powerful tool in knowledge discovery and data analysis. It combines the tasks, and benefits, of vector quantization and data projection. The various novel visualization methods presented in this chapter offer efficient ways to enhance the visualization of the SOM in data exploration. There are many kinds of tasks in exploratory visualization, but as the proposed principles are simple, they can be easily modified to meet the needs of the task. Future work is still needed to enable the methods to automatically take heed of the properties of the underlying data.

The SOM can be effectively used to find and visualize correlations between process variables in different operational states of the process. The topology preserving property together with the regular presentational form of the SOM visualization gives a compact base where many kinds of visualizations and interfaces may be linked together.

In this chapter, we have used the basic SOM visualizations together with methods that link different kind of visualizations using color. However, there are some aspects in the methods that should be noted:

- One should remember when using color visualizations that there are color-blind people who do not see the color space as the majority of people do.
- The color coding that we have used is of heuristic design, something to start with. Furthermore, a coloring that brings up the cluster structure (see [8; 9]) would certainly be beneficial.
- The linking between the scatter plots and the SOM could be made interactively by highlighting the interesting points. However, the color coding brings an automated overall sight to this procedure.
- The scatter plots connected to the map grid will benefit the analysis only if the dependencies are such that a variable can be considered to be (locally) a function of mainly one other latent variable. If the dependencies are more complex, the scatter plot visualization with the color linking becomes useless.

Despite their evident limitations, the methods presented have facilitated the industrial data analysis, especially in the explorative phase of the work.

It should be emphasized that the data analysis process usually is iterative, i.e., the most important variables can be determined only after various steps of the data mining process. In the beginning, there are usually several dozens of measurements which will then be reduced to the most important ones affecting the behavior of the process. Several tests must be made and interpreted using knowledge of process experts.

## 2.7 Acknowledgments

The research in this work was carried out in the technology program “Adaptive and Intelligent Systems Applications” financed by the Technology Development Centre of Finland (TEKES) and in the “Application of Neural Network based Models for Optimization of the Rolling Process” (NEUROLL) project financed by the European Union. The cooperation of the following enterprises is gratefully acknowledged: Jaakko Poyry Consulting, Rautaruukki and UPM-Kymmene.

We acknowledge Mr. Juha Parhankangas for carrying out various elaborate programming tasks in our projects.

## References

[1] E. J. Ainsworth, "Classification of Ocean Colour Using Self-Organizing Feature Maps," Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (Eds. T. Yamakawa, G. Matsumoto), Vol. 2, pp. 996–999, World Scientific, 1998.

[2] E. Alhoniemi, J. Himberg, J. Vesanto, "Probabilistic Measures for Responses of Self-Organizing Map Units," Proceedings of International ICSC Congress on Computational Intelligence and Applications (CIMA'99) (Eds. H. Bother, E. Oja. E. Massad, C. Haefke), pp. 286–290, 1999.

[3] V. Aristide, "On the use of two traditional statistical techniques to improve the readability of Kohonen Maps," Proceedings of the NATO ASI on Statistics and Neural Networks, Les Arcs, France, 1993.

[4] P. Demartines, J. Hérault, "Curvilinear Component Analysis: a Self-Organizing Neural Network for Nonlinear Mapping of Data Sets," IEEE Transactions on Neural Networks, Vol. 8, pp. 148–154, 1997.

[5] J. Himberg, "Enhancing SOM-based data visualization by linking different data projections," Proceedings of 1st International symposium on Intelligent Data Engineering and Learning 1998 (IDEAL'98) (Eds. L. Xu, L. W. Chan, I. King, A. Fu), pp. 427–434, Springer, 1998.

[6] L. Holmström, A Hämäläinen, "The Self-Organizing Reduced Kernel Density Estimator," Proceedings of the International Conference on Neural Networks (ICNN'93), San Francisco, pp. 417–421, 1993.

[7] J. Iivarinen, T. Kohonen, J. Kangas, J., S. Kaski, "Visualizing the Clusters on the Self-Organizing Map," Proceedings of the Conference on Artificial Intelligence Research in Finland (Eds. C. Carlsson, T. Järvi, T. Reponen), number 12, pp. 122–126, Finnish Artificial Intelligence Society, 1994.

[8] S. Kaski, J. Venna, T. Kohonen, "Tips for Processing and Color-Coding of Self-Organizing Maps," in Visual Explorations in Finance (Eds. G. Deboeck, T. Kohonen), Ch. 14, pp. 195–202, Springer-Verlag, 1998.

[9] S. Kaski, J. Venna, T. Kohonen, "Coloring that Reveals High-Dimensional Structures in Data," Proceedings of the 6th International Conference on Neural Information Processing (ICONIP'99), Vol. II, pp. 729–734, 1999.

[10] M. Kasslin, J. Kangas, O. Simula, "Process State Monitoring Using Self-Organizing Maps," in Artificial Neural Networks (Eds. I. Aleksander, J. Taylor), Vol. 2, pp. 1531–1534, North-Holland, 1992.

[11] K. Kiviluoto, E. Oja, "S-Map: A network with a simple self-organization algorithm for generative topographic mappings," in Advances in Neural Processing Systems (Eds. M. I. Jordan, M. J. Kearns, S. A. Solla), no. 10, pp. 549–555, MIT Press, 1997.

[12] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences 30, Springer, 1995.

[13] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, "SOM_PAK: The Self-Organizing Map Program Package," Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[14] T. Kohonen, S. Kaski, K. Lagus, T. Honkela, "Very Large Two-Level SOM for the Browsing of Newsgroups, " Proceedings of ICANN'96, pp. 269–274, 1996.

[15] T. Kohonen, E. Oja, O. Simula, A. Visa, J. Kangas, "Engineering Applications of the Self-Organizing Map," Proceedings of the IEEE, 84(10), pp. 1358–1384, 1995.

[16] MathWorks Inc, MATLAB — The Language of Technical Computing: Using MATLAB Version 5, MathWorks Inc., 1997.

[17] N. R. Pal, V. K. Eluri, "Two Efficient Connectionist Schemes for Structure Preserving Dimensionality Reduction," IEEE Transactions on Neural Networks, Vol. 9, no. 6, pp. 1142–1154, 1998.

[18] J. W. Sammon, Jr., "A Nonlinear Mapping for Data Structure Analysis," IEEE Transactions on Computers, C-18(5), pp. 401–409, 1969.

[19] O. Simula, J. Kangas, "Process monitoring and visualization using self-organizing maps," Computer-Aided Chemical Engineering, Vol. 6, Ch. 14, pp. 371–384, Elsevier, 1995.

[20] V. Tryba, S. Metzen, K. Goser, "Designing of Basic Integrated Circuits by Self-Organizing Feature Maps," Proceedings of Neuro-Nimes '89, Int. Workshop on Neural Networks and their applications, pp. 225–235, Nanterre, France, 1989.

[21] V. Tryba, K. Goser, "Self-Organizing Feature Maps for Process Control in Chemistry," in Artificial Neural Networks (Eds. T. Kohonen, K. Mäkisara, O. Simula, J. Kangas), Vol. 1, pp. 847–852, North-Holland, 1991.

[22] A. Ultsch, H. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," Proceedings of the International Neural Network Conference (INNC'90), pp. 305–308, Kluwer, 1990.

[23] J. Vesanto, J. Himberg, M. Siponen, O. Simula "Enhancing SOM Based Data Visualization," Proceedings of the 5th International Conference on

Soft Computing and Information/Intelligent Systems (Eds. T. Yamakawa, G. Matsumoto), pp. 64–67, World Scientific, 1998.

[24] J. Vesanto, J. Ahola, "Hunting for Correlations in Data Using the Self-Organizing Map," Proceedings of International ICSC Congress on Computational Intelligence and Applications (CIMA'99) (Eds. H. Bother, E. Oja. E. Massad, C. Haefke), pp. 279-285, 1999.

[25] J. Vesanto, E. Alhoniemi, J. Himberg, K. Kiviluoto, J. Parviainen, "Self-Organizing Map for Data Mining in MATLAB: the SOM Toolbox," Simulation News Europe, 25, p. 54, ARGE Simulation News, 1999.

[27] J. Vesanto, "SOM-based data visualization methods," Intelligent Data Analysis, Vol. 3, No. 2, pp. 111-126, Elsevier, 1999.