Knowledge Discovery and Knowledge Validation in Intensive Care[1]

Katharina Morik*, Michael Imhoff +, Peter Brockhausen*,
Thorsten Joachims*, Ursula Gather++

*) Department of Computer Science,
++) Department of Statistics
Universität Dortmund, D-44221 Dortmund, Germany

+) Surgical Department, City Hospital Dortmund
Beurhausstrasse 40, D-44137 Dortmund, Germany

{morik, brockhausen, thorsten}@ls8.cs.uni-dortmund.de
gather@statistik.uni-dortmund.de
mike@imhoff.de

1 Abstract

Operational protocols are a valuable means for quality control. However, developing operational protocols is a highly complex and costly task. We present an integrated approach involving both intelligent data analysis and knowledge acquisition from experts that supports the development of operational protocols. The aim is to ensure high quality standards for the protocol through empirical validation during the development, as well as lower development cost through the use of machine learning and statistical techniques. We demonstrate our approach of integrating expert knowledge with data driven techniques based on our effort to develop an operational protocol for the hemodynamic system.

1. (To appear in "***Artificial Intelligence in Medicine***", thematic issue on Knowledge-Based Information Management in Intensive Care and Anaesthesia)

Morik et al: Knowledge Discovery and Knowledge Validation in Intensive Care

2 Key words

operational protocols, online-monitoring, time series analysis, machine learning, statistical pattern recognition, knowledge based systems, knowledge validation

3 Introduction

An abundance of information is generated during the process of critical care. Much of this information can now be captured and stored using clinical information systems (CIS) that have become commercially available for use in intensive care over the last years. These systems provide for a complete medical documentation at the bedside and their clinical usefulness and efficiency has been shown repeatedly [8, 9, 13]. While databases with more than 2,000 separate patient-related variables are now available for further analysis [10], the multitude of variables presented at the bedside even without a CIS precludes medical judgement by humans. A physician may be confronted with more than 200 variables in the critically ill during a typical morning round [25]. We know, however, that even an experienced physician is often not able to develop a systematic response to any problem involving more than seven variables [22]. Moreover, humans are limited in their ability to estimate the degree of relatedness between only two variables [15]. This problem is most pronounced in the evaluation of the measurable effect of a therapeutic intervention. Personal bias, experience, and a certain expectation toward the respective intervention may distort an objective judgement [6]. These arguments motivate the use of decision support systems.

Clinical decision support aims at providing health care professionals with therapy guidelines directly at the bed-side. This should enhance the quality of clinical care, since the guidelines sort out high value practices from those that have little or no value. The goal of decision support is to supply the best recommendation under all circumstances [26]. The computerized protocol of care can take into account more aspects of the patient than a physician can accommodate. It is not disturbed by circumstances or hospital constraints. It bridges the gap between low-level numerical measurements (the level of the equipment) and high-level qualitative principles (the level of medical reasoning). While knowledge-based systems have mostly been applied for diagnosis and therapy planning (e.g., [28], [19]), some systems also aim at on-line patient monitoring [7, 21, 26].

Methods that have proved their value in handling low-frequency patient data are not applicable for on-line monitoring [21]. Quantitative measurements and qualitative reasoning have to be integrated in a system that recommends interventions in real-time. The numerical measurements of the patients‘ vital signs have to be abstracted into qualitative terms of high abstraction. The aspect of time has to be handled both at the level of measurements and the level of expert knowledge [4, 17, 21, 28]. In the expert's reasoning, time becomes the relation between time intervals, abstracting from the exact duration of, e.g., an increasing heart rate, and focusing on tendencies of other parameters (e.g., cardiac output) within overlapping time intervals.

One of the big obstacles to the more frequent implementation of decision support systems is the tedious and time-consuming task of developing the knowledge base. The decision support system for respiratory care at the LDS Hospital, Salt Lake City, USA [26], for instance, has been developed in about 25 person years. The method of guideline development itself is not supported by a computer system. Mechanisms of temporal abstraction and reasoning presuppose manually designed models or ontologies [4, 21, 28]. Why not use techniques of knowledge discovery and statistical time series analysis in order to ease the process of guideline generation? Machine learning and statistical analysis have been applied in building-up diagnostical systems successfully (e.g., [18]). We now want to exploit the huge amount of data for the development of guidelines for on-line monitoring. Our task is to build a decision support system for on-line hemodynamic monitoring in the critically ill. We do not aim at modeling the actual physician's behavior. Imitating the actual interventions made by physicians is not the goal. Actual behavior is influenced by the overall hospital situation, e.g., how long is the physician on duty, how many patients require attention at the same time. Machine learning from patients‘ data could lead to a knowledge base that mirrors such disturbing effects. Therefore, the learned decision rules have to be checked by additional rules about effects of drug and fluid administration. Our approach is to combine statistics, knowledge acquisition, and machine learning. Our aim is to develop a method for guideline generation that is faster and more reliable than current methods.

Data for statistical evaluation and learning can be provided by the CIS. However, the nature of the data is different from that gathered in controlled experiments. While a CIS in modern intensive care can take numerous measurements every minute, the values of

Morik et al: Knowledge Discovery and Knowledge Validation in Intensive Care

some vital signs are sometimes recorded only once every hour. Other vital signs are recorded only for a subset of the patients. Hence, the overall high dimensional data space is sparsely populated. Moreover, the average time difference between intervention as charted and estimated hemodynamic effect can show a wide variation [12]. Even the automatic measurements can be noisy due to manipulation of measurement equipment, flushing of pressure transducers, or technical artifacts. In some cases, relevant demographic and diagnostic parameters may even not be recorded at all. In summary, we have a large amount of high dimensional, numerical time series data that contains missing values and noise. Using this data already at the stage of development of the decision support system stave off surprises at the stage of clinical experience as have been reported in [21, p. 572]: "The huge number of measurements classified as invalid is quite astonishing although it reflects the real clinical environments."

In addition to problems of knowledge acquisition, we see a particular need for knowledge validation. It should be noted that many medical guidelines published today are neither evidence-based nor sufficiently validated against real patient data. The current procedure is to first develop the guideline, then represent it in a knowledge-based system, and finally to test it in clinical studies. In this ‘waterfall´ process, unrealistic assumptions, mistakes, and flaws are recognized at a late stage. In contrast, our approach includes validation from the very beginning. Using a knowledge-based system early on supports the validation of the knowledge base at earlier stages. Inconsistencies within the knowledge base as well as a mismatch of rules and patient data are detected while developing the knowledge base. This facilitates and focuses the knowledge-acquisition process.

In order to test our approach to using real clinical data for building and validating a knowledge base for on-line monitoring, we have constructed a system. Its overall architecture is shown in Figure 1. The patients' measurements are used to recommend an intervention and are abstracted with respect to their course over time. The recommendation of interventions constitutes a model of physician behavior. This asks for further validation. Therefore, a recommended intervention is checked by calculating its expected effects on the basis of medical knowledge. In this way, a qualitative assessment of a statistical prediction enhances the model of physician behavior in order to obtain a model of best practice. The medical knowledge constitutes a model of the patients' hemodynamic system. This model is validated with respect to past patients' data. In detail, the processes we have designed are:

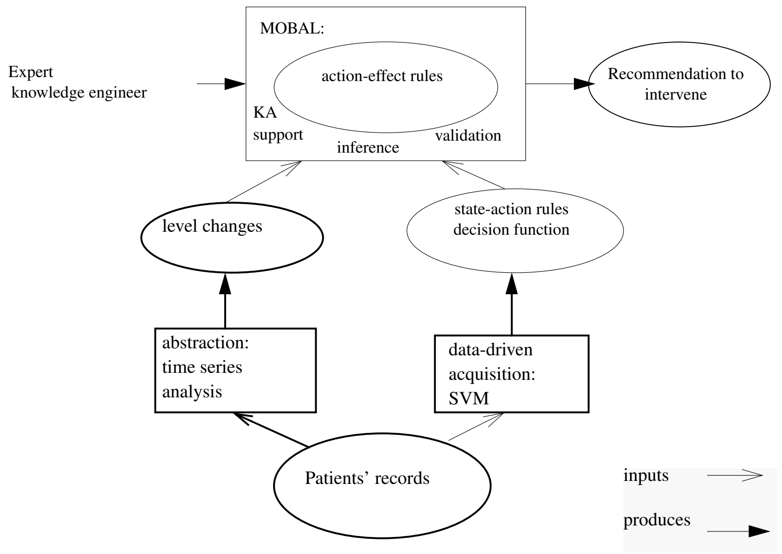

Figure 1. Overall architecture.

data abstraction: Given series of measurements of one vital sign of the patient, detect and possibly eliminate outliers and find level changes by good statistical practice. This abstracts the measurements to qualitative propositions with respect to a time interval, e.g., within time point 12 and time point 63, the heart rate remained about equal, from time point 63 to time point 69 it was increasing. Our approach is based on statistical time series analysis. Classical ARMA (autoregressive moving average) modelling [3] is applied with corresponding outlier- and level shift detection procedures using the new tool of a phase space embedding.

data-driven acquisition of state-action rules: Given the numerical data describing vital signs of the patient and his or her current medication, find the appropriate intervention. An intervention is formalized as increasing, decreasing or not changing the dose of a drug. The decision is made every minute. These rules were learned by the Support Vector Machine [38].

acquisition of medical knowledge: Given text book knowledge and explanations by an expert, represent the effects of substances in different dosages, relations between vital signs, and interrelations between different substances, and validate the knowledge on the basis of past patients' data. The knowledge acquisition and validation was supported by the MOBAL system [24].

validation of recommended interventions: Given

- the state of a patient described in qualitative terms,
- medical knowledge
- a sequence of interventions, and
- a current intervention,

find the effects of the current intervention on the patient. The derivation of effects is made for each intervention as forward inference within MOBAL. The effect should result in a stable state of the patient.

The outline of this paper is as follows. Throughout the paper we report on the continuous development of a decision support system for intensive care as performed at the city hospital and the university of Dortmund. We start with a description of the data acquisition process at the hospital and the resulting data set [13]. A statistical method for data abstraction is described in section 5. The next section (6) shows, how we applied the support vector machine (SVM) to learn state-action rules. A short introduction to the MOBAL system [24] and its representation of medical knowledge leads to the issue of validation which is presented in section 8.

4 Data acquisition and data set

4.1 Data acquisition

Most variables are entered by hand at the bedside. For entities such as clinical observations, nursing procedures, therapeutic measures, medications, or orders it appears very unlikely that entry of these variables can be automated in the foreseeable future. Only 5-

10% of all variables in a CIS are acquired automatically. This includes the majority of bedside devices, e.g. physiologic monitors, ventilators, infusion devices. Additional data is interfaced from the hospital information system (HIS), the laboratory (LIS) or the microbiology information systems, where the LIS represents the clinically most relevant set of data among these centralized information systems. Although device data account for a comparatively small number of variables, they can, depending on the sampling rate, generate large amounts of data.

The data structure of a CIS shows a wide variety of different data types on different scales (nominal scales, e.g. sex, breathing sounds; ordinal scales, e.g. neurological scoring; absolute scales, e.g. vital signs), which are stored at different time intervals (ranging from seconds for vital signs to once during the length of stay for demographic data). Time intervals may also be regular or irregular.

For further analysis data must be structured, so that it can be subjected to statistical algorithms. Numeric data, e.g. vital signs, intake/output, is typically directly accessible for most applications. Free-text data, which traditionally makes up a large portion of medical documentation, cannot be statistically analyzed in any structured way. Therefore, free-text entries into a CIS should be avoided wherever possible. Qualitative information, such as clinical observations or interventions, should be documented in a strictly structured fashion with selection lists and menu items. This approach provides a consistent terminology throughout the entire medical institution. It is highly efficient and fast, especially for users not well trained in the use of computers and keyboards in particular. In clinical practice, with the stringent implementation of structured tabular documentation, it was possible to reduce the use of free-text notes by more than 90%. Structured qualitative data can, in contrast to free-text information, be directly exported for statistical analysis.

These general propositions also hold for the city hospital of Dortmund, a 1,900-bed tertiary referral center. There, all medication data of the 16-bed surgical intensive care unit was charted with a CIS, allowing the user one minute time resolution for all data. Moreover, data from bedside devices, e.g. patient monitors, is gathered automatically every minute.

Table 1. Overall attribute set for learning state-effect rules.

16 demographic attributes	5 intensive care diagnoses	6 continuously infused drugs
11 vital signs	9 derived parameters	14 respiratory variables
37 intake/output variables	10 bolus drugs	10 laboratory tests

4.2 Data set

The entire database of intensive care patient records at the city hospital of Dortmund comprises about 2,000 different variables (attributes). Data from the CIS is selected through customizable data filters and copied into a standard relational database where it is accessible for further data analysis.

For this investigation, data was acquired from 148 consecutive critically ill patients (53 female, 95 male, mean age 64.1 years), who had pulmonary artery catheters for extended hemodynamic monitoring. Recording in one minute intervals, this amounts to 679,817 sets of observations.

From the original database 118 attributes in 9 groups were taken for learning state-action rules (table 1).

Categorical attributes are broken down into a number of binary attributes, each taking the values {0,1}. Real valued parameters are either scaled so that all measurements lie in the interval [0,1], or they are normalized by empirical mean and variance: $norm(X) = (X - means(X))/\sqrt{var(X)}$

We systematically evaluated a large number of plausible attribute sets using a train/test scheme on the learning task described in section 6.2. The set with the best performance is given in table 2. These attributes are actually the most important parameters of the patient according to expert judgement. Only the relevant attributes "Cardiac Output" and "Net Intake/Output" are missing, but they cannot be used as they are not continuously available.

We also experimented with different ways of incorporating the history of the patient. We tried:

- using only the last minute before the intervention

Table 2. Best feature set for learning state-action rules using SVM.

Vital Signs (measured every minute)	Continuously Given Drugs (changes charted at 1-min-resolution)	Demographic Attributes (charted once at admission)
Diastolic Arterial Pressure	Dobutamine	Broca-Index
Systolic Arterial Pressure	Adrenaline	Age
Mean Arterial Pressure	Glycerol trinitrate	Body Surface Area
Heart Rate	Noradrenaline	Emergency Surgery (y/n)
Central Venous Pressure	Dopamine	
Diastolic Pulmonary Pressure	Nifedipine	
Systolic Pulmonary Pressure		
Mean Pulmonary Pressure		

- using the last up to 10 minutes before the intervention
- using the averages of up to 60 minutes before the intervention
- combinations of these
- the state of the patient at the previous intervention

None of the more complex approaches gave significantly better results on the learning task in section 6.2 than just using the measurements from one minute before the intervention. All the feature selection experiments were done on the training set, leaving a separate test set to measure the results presented in this paper.

Since each patient record covers several interventions, data from 148 patients gives us sufficiently large sets of examples. For learning state-action rules, we used a total of 1319 training and 473 test examples. For the rule validation we analyzed 8200 interventions corresponding to 27400 intervention-effect pairs.

5 Statistical analysis of time series

Time series analysis was employed for data abstraction of the time oriented variables

with the goal of detecting outliers and level changes. The classical and widely used statistical approach to modelling time series is so called ARMA modelling [3] which assumes that a time series $(x_t)_{t = 1, 2, \dots}$ can be written as

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \qquad (1)$$

where ε_t is an unobservable shock at time t. This assumption means that each observation is a linear combination of past observations and past shocks with (unknown) coefficients $\phi_1, \dots, \phi_p$ and $\theta_1, \dots, \theta_q$ respectively. The integers p and q are the orders of the model, while the model itself is denoted $\mathrm{ARMA}(p, q)$ model. In case of $q = 0$, i.e., when only the current shock and past observations have influence, the model is called $\mathrm{AR}(p)$ model.

In order to better understand the dynamics of time dependent phenomena, another representation of a time series can also be applied, the so called phase space embedding [27, 32]. This tool, though originally designed to analyse nonlinear, chaotic systems, has proven useful for the purpose of detecting outliers and level changes also in an ARMA framework [2, 11]. Here it does not even need any strict model assumptions to be applied. The phase space approach is based on a simple transformation of the time series into some Euclidean space - the phase space embedding. Instead of the time series $x_1, \dots, x_N$ itself, one considers the sequence of all, say m, consecutive values of the series as m-dimensional vectors

$$\vec{x}_t = (x_t, x_{t+1}, \dots, x_{t+m-1}), \qquad t = 1, \dots, N - m + 1. \qquad (2)$$

Here, $m \in \{1, 2, \dots\}$, is called embedding dimension. Numerous rules exist for choosing m in nonlinear models. There, in most cases the components of the phase space vectors are not neighboring observations, but they are separated by a time delay [14, 34]. Focusing on stochastic processes it is better to take into account the dependencies of neighboring observations. Therefore, we choose the components of $\vec{x}_t$ as chronological observations always with a time delay (lag) of one. To improve pattern identification, m should be chosen such that exactly those preceding observations are considered, which have a direct influence on the present observation [2, 13]. Of course, to detect outliers, level changes etc., the classical statistical methodology may also be applied [5, 29, 31, 35]. We suggest to work with the more recent phase space method here mainly because it is able to detect even patchy outliers, because it works almost graphically without strict model assumptions, and because it may reveal deviations from the dynamics of the series which are difficult to detect otherwise [2].

Figure 2 visualizes a two-dimensional phase space embedding. We connect all consecutive phase space vectors, i.e., all points (x_1,x_2), (x_2,x_3), (x_3,x_4), and so on, in the two-dimensional Euclidean space. Typically, in the steady state this yields an elliptical cloud and outliers show up as abberations from this cloud.

The identification procedure, that we developed, uses the differenced time series d_t, defined by $d_t = x_t - x_{t-1}, t = 2, \ldots, N$. In a differenced series, an abrupt level change shows the same abberation as an outlier in the original series. The procedure focuses on the identification of such abberations from that elliptical cloud which describes the steady state.

The phase space vectors of the differenced time series are analyzed in consecutive order and it is checked whether they are located in a "critical region" . If a vector lies in such a critical region, i.e., intuitively speaking, if it extrudes "too far" from the elliptical cloud describing the steady state, it can then be discriminated between different patterns after observing further values. The "critical region" is formally defined as the region outside some ellipse, which in the steady state for a given level α must not contain any phase space vector out of N such vectors with probability larger than $1 - \alpha$. The level α is chosen by the investigator and, if possible, depending on the observed length N of the time series. In any case, as the true data generating mechanism of the time series is unknown, the critical region has to be estimated from some starting sequence of the time series. Then the outlying phase space vectors located in this critical region can be detected and, finally, on the basis of the movement of the phase space vectors, a discrimination between different patterns can be done for those observations which have constituted an outlying phase space vector.

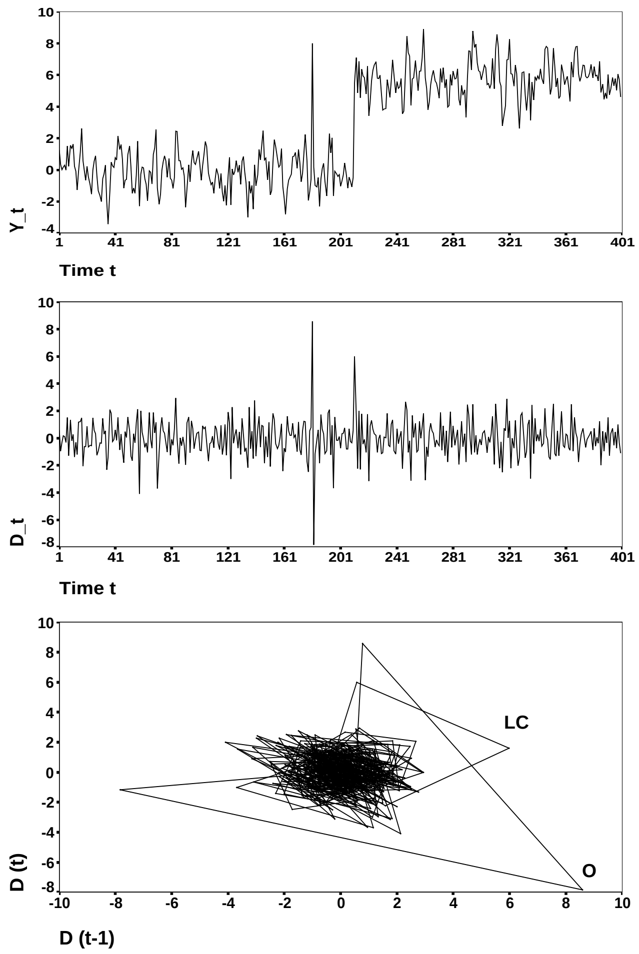

Figure 2. Phase Space Embedding of a simulated time series:
Top: Simulated AR(1)-process with outlier and level change
Middle: Differenced AR(1)-process with outlier and level change.
Bottom: Phase space embedding of the differenced series (O = outlier, LC = level change).
Plot of d_n versus d_(n-1). After a change occurred the phase space vectors fall back into the ellipse describing the steady state.

A detailed description of this methodology and a comparison with other established time series procedures is given in [12, 2]. A graphical example may visualize this approach. In Figure 2 a simulated time series following an AR(1) process, the differenced series and the two-dimensional embedding of the differenced series with the corresponding estimated ellipse are shown. This example discriminates only between the patterns of outliers and abrupt level changes. If at the time point t the vector $\vec{d}_t$ extrudes from the cloud, the decision between outlier and abrupt level change takes place at the time point $t+1$. If the distance between $\vec{d}_{t+1}$ and the detected point LC in Figure 2 is smaller than the distance between $\vec{d}_{t+1}$ and the detected point O, then a level change can be diagnosed, otherwise, an outlier is present. If more patterns are considered, more fixed points have to be determined and the time point at which the decision takes place is delayed.

The whole process of pattern recognition can be described as follows: The first 60 observations are taken and retrospectively analyzed (i.e., outlying regions are estimated and patterns in this time interval identified). After this, a time window of length 60 is moved through the data. That means, that at time point 61 we determine if the phase space vector $\vec{d}_{61}$ is in the critical region. If not, then no pattern is detected, and the estimated critical region is replaced by a new one, that is estimated from the last 60 observations $d_2, \ldots, d_{61}$. This is repeated for every new observation as long as for the time point t the phase space vector $\vec{d}_t$ once falls into a critical region. Then the system is said to be no longer in a steady state, and after analyzing the consecutive observations $d_{t+1}, d_{t+2}, \ldots$, it is decided if a pattern is present similar to the retrospective analysis.

In a previous study [13] this approach showed excellent results when compared to pattern recognition by highly trained experts. From this investigation it can be assumed that clinically significant patterns will be reliably detected by this method. An initial weakness in the detection of trends or "slow" level changes was overcome by the implementation of a delayed moving window.

The data abstraction process using phase space methods handed the recognized patterns over to the other components of the hybrid system. This information contained type, time of onset, direction, and duration of the pattern.

6 Data-driven acquisition of state-action rules

6.1 Support vector machine

Support vector machines (SVMs) [38] represent a method to learn binary classifiers from examples. For a set of training examples $(\vec{o}_1, y_1), \ldots, (\vec{o}_n, y_n)$ they find the classification rule h for which they can guarantee the lowest error rate on new observations. Each example consists of a vector $\vec{o}_t$ (describing e.g. the state of a patient represented by the current measurements of blood pressures, heart rate, etc.) and its classification $y_t \in \{1, -1\}$. In their basic form, SVMs learn linear decision rules $h(\vec{o}) = \operatorname{sgn}(\vec{w} \cdot \vec{o} + b)$. The weight vector $\vec{w}$ and the threshold b are the result of learning and describe a hyperplane. Observations are classified according to which side of the hyperplane they are located. A typical decision rule is given in Figure 3. During training, the SVM calculates the hyperplane so that it classifies most training examples correctly while keeping a large "margin" around the hyperplane. If the training data can be separated without error, the margin is the distance from the hyperplane to the closest training examples.

Since we will be dealing with very unbalanced numbers of positive and negative examples in the following, we introduce cost factors C_+ and C_- to be able to adjust the cost of false positives vs. false negatives. Training an SVM can now be translated into the following optimization problem:

Minimize: $$J(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C_+ \sum_{i:y_i = 1} \xi_i + C_- \sum_{j:y_j = -1} \xi_j \qquad (3)$$

subject to: $$\forall t \in [1 \ldots n] : y_t[\vec{w} \cdot \vec{o}_t + b] \geq 1 - \xi_t \quad \wedge \quad \xi_t \geq 0 \qquad (4)$$

Training error is represented by the variables ξ_t, while the margin is measured by $\vec{w}$. We solve this optimization problem in its dual formulation using SVM^{light} [1] [16], extended to handle unsymmetric cost-factors.

1. Available at http://www-ai.cs.uni-dortmund.de/svm_light

Table 3. Accuracy in predicting the right direction of an intervention.

Drug	Accuracy	StdErr
Dobutamine	83.6%	2.5%
Adrenaline	81.3%	3.7%
Glyceroltrinitrate	85.5%	3.0%
Noradrenaline	86.0%	5.2%
Dopamine	84.0%	7.3%
Nifedipine	86.9%	7.0%

6.2 Learning the directions of interventions

The first question we asked ourselves was: Given that we know the physician changed the dosage of some drug, can we learn when he increased the dosage and when he decreased the dosage based on the state of the patient? For each drug, examples are taken from the points in time where, in fact, the dosage changed. For all drugs, linear SVMs are trained on the problem "increase of dosage" ($y_t = 1$) vs. "decrease of dosage" ($y_t = -1$) using the attributes in table 2 for describing the state of the patient. The performance of the respective SVM on a previously untouched test set is given in table 3.

To get an impression about how good these prediction accuracies are, we conducted an experiment with a physician. On a subset of 41 test examples we asked an expert to do the same task as the SVM for Dobutamine, given the same information about the state of the patient. In a blind test the physician predicted the same direction of dosage change as actually performed in 32 out of the 41 cases. On the same examples the SVM predicted the same direction of dosage change as actually performed in 34 cases, resulting in an essentially equivalent accuracy.

6.3 Learning when to intervene

The previous experiment shows that SVMs can learn in how far drugs should be changed given the state the patient is in. In reality, the physician also has to decide when to intervene or just keep a dosage constant. This leads to the following three class learning problem. Given the state of the patient, should the dosage of a drug be increased, decreased or kept constant? Generating examples for this task from the data is difficult. The particular minute a dosage is changed depends to a large extend on external conditions (e.g. an emergency involving a different patient). So interventions can be delayed and the optimal minute an intervention should be performed is unknown. To make sure that we generate examples only when a physician was closely monitoring the patient, we consider only those minutes where some drug was changed. This leads to 1319 training and 473 test examples.

Morik et al: Knowledge Discovery and Knowledge Validation in Intensive Care

Table 4. Confusion matrix for predicting time and direction of Dobutamine and Adrenaline interventions.

Dobutamine	actual intervention up	actual intervention equal	actual intervention down
predicted up	46	32	3
predicted equal	50	197	54
predicted down	5	30	56
Adrenaline	**up**	**equal**	**down**
predicted up	23	22	3
predicted equal	21	310	15
predicted down	4	34	41

For each drug we trained two binary SVMs. One is trained on the problem "increase dosage" vs. “do not increase dosage (i.e. lower or keep dosage equal)", the other one is trained on the problem "lower dosage" vs. "do not lower dosage (i.e. increase or keep dosage equal)". An intervention is predicted if exactly one such decision rule recommends a change. As an example, Figure 3 shows the decision rule that the SVM learned for increasing the dosage of Glyceroltrinitrate. Since the class distribution is very skewed towards the "do not ... dosage" class, we use a cost model. The cost-factors are chosen so that the potential total cost of the false positives equals the potential total cost of the false negatives. This means that the parameters C_+ and C_- of the SVM are chosen to conform to the ratio

$$\frac{C_+}{C_-} = \frac{\text{number of negative training examples}}{\text{number of positive training examples}} \qquad (5)$$

Table 4 shows the test results for Dobutamine and Adrenaline. The confusion matrices give insight into the class distributions and the type of errors that occur. The diagonal contains the test cases, where the prediction of the SVM was the same as the actual intervention of the physician. This accounts for 63% of the test cases for Dobutamine and for

Table 5. Confusion matrix for predicting time and direction of Dobutamine/Adrenalin interventions in comparison to human performance (results from an experience intensivist in brackets).

Dobutamine	actual intervention		
	up	equal	down
predicted up	10 (9)	12 (8)	0 (1)
predicted equal	7 (9)	35 (31)	9 (9)
predicted down	2 (1)	7 (15)	13 (12)
Adrenaline	**actual intervention**		
	up	equal	down
predicted up	4 (2)	3 (1)	0 (0)
predicted equal	4 (6)	65 (66)	2 (2)
predicted down	1 (1)	8 (9)	8 (8)

79% of the test cases for Adrenaline. The SVM suggests the opposite intervention in about 1.5% for both drugs.

Again, we would like to put these numbers into relation to the performance of an expert when given the same information. For a subsample of 95 examples from the test set, we asked a physician to perform the same task as the SVM. The results for Dobutamine and Adrenaline are given in table 5. The results of the SVM on this subsample are followed by the performance of the human expert in brackets. Both are aligned remarkably well. Again, the learned functions of the SVM are comparable in terms of accuracy with a human expert. This also holds for the other drugs.

7 Medical knowledge base

Decision rules learned by the SVM reflect the average behavior of a physician, not the "gold standard". As argued above, they have to be checked against medical knowledge about the effects of drugs. This section presents an approach to building a knowledge base that helps accomplish this task automatically and that makes decision support transparent.

Knowledge acquisition from experts is performed according to the current state of the art: first, knowledge is elicited from the expert, second, a knowledge base is modeled, third, the model is inspected, validated, and enhanced in collaboration with the expert.

Morik et al: Knowledge Discovery and Knowledge Validation in Intensive Care

Table 6. Medical knowledge base for hemodynamic effects:
+ = increase of the respective variable or intervention'; - = decrease of the respective variable or intervention; 0 = no change

Intervention		Effect on hemodynamic variables				
		Heart Rate	Mean Arterial Pressure	Mean Pulmonary Artery Pressure	Central Venous Pressure	Cardiac Output
Dobutamine	+	+	+	+	0	+
	-	-	-	-	0	-
Adrenaline	+	+	+	+	0	+
	-	-	-	-	0	-
Noradrenaline	+	-	+	+	0	-
	-	+	-	-	0	+
Nitroglycerine	+	+	-	-	-	+
	-	-	+	+	+	-
Fluid intake/ out- put	+	-	+	+	+	+
	-	+	-	-	-	-

These steps form a cycle, i.e. the third step actually leads to obtain more expert knowledge, which is then modeled, etc.[23]. This expert knowledge augments and validates the data-driven knowledge acquisition using machine learning.

7.1 Knowledge acquisition and representation

The knowledge base of action-effect rules serves three purposes. First, it is used in order to model a protocol of care. Second, it is used to base learned decision functions on explicit and qualitative knowledge. Third, it is used for the validation of predictions. Let us describe the knowledge acquisition from experts before we show how this knowledge is integrated with the learned decision functions (section 7.3) and how it is used for validating predictions (section 8).

A medical expert defined the necessary knowledge. This knowledge is medical textbook knowledge for the cardiovascular system. It reflects direct pharmacological effects of a selected list of medical interventions on the basic hemodynamic variables. Any interaction of these interventions with other organ systems or of other organ systems with the cardiovascular system were ignored. An excerpt of intervention-effect relations is shown in table 6. The dosage intervals indicated for each drug are not shown in the table, but modeled in the knowledge base. Also parameter dependencies have been modeled. It should be noted that the knowledge is qualitative with intervals of dosages, trends of changes, and implicit time intervals.

For the representation of qualitative medical knowledge we chose the MOBAL system [24]. MOBAL is a knowledge acquisition and maintenance system. Several tools facilitate the construction and inspection of a knowledge base. Its representation formalism is a restricted many-sorted first-order logic with explicit negation. A four-valued logic is used in order to allow for unknown and contradictory facts in addition to true and false facts. The inference engine derives new facts on the basis of rules and given facts. Due to the expressive power of first-order logic, compact models can be built. What would be a rule in propositional logic, can be expressed by a mere fact in first-order logic. For instance, using a propositional logic, explicitly stating that *up* is the opposite of *down* requires the rule

heart_rate_trend=up --> not (heart_rate_trend=down)

and its dual form for all parameters. Using first-order logic, the fact

opposite(up, down)

is stated and can be used for any parameter.[1] The pharmacological knowledge from table 6 is expressed by facts of the form

effect(adrenaline, 0.01, 0.03, art, up)

stating that Adrenaline in a dosage between 0.01 and 0.03 µg/kg/min has the effect *up* on mean arterial pressure. Effects are modeled for substances. Additional facts indicate the particular drugs in which the substance is contained.

Patient records are also expressed by facts. The time is indicated by minutes, starting with the first measurement of a patient and ending with his or her discharge from intensive care.

intervention(pat4711, 10, 62, supra,0.02)

means that the patient 4711 from the tenth minute to minute 62 received Suprarenin (a drug containing Adrenaline) in a dosage of 0.02 µg/kg/min.

1. We follow the standard notation of logic programming, where argument variables begin with capital letters and predicate symbols as well as constants start with small letters.

Morik et al: Knowledge Discovery and Knowledge Validation in Intensive Care

Given the abstractions described in section 5, the values of hemodynamic parameters are stated in terms of level changes.

level(pat4711, 11, 62, hr, up)

states that the heart rate of patient 4711 had an upward level change at minute 11 and then remained almost stable until minute 62. In addition to this abstract description of a vital sign in a time interval, its deviation from the stable state is calculated. For each vital sign, the desired range of values is given, e.g. [60, 100] for the heart rate. For a patient‘s parameter values within a time interval, the standard deviation is calculated and added to (subtracted from) the upper (lower) value of the desired range. If the patient‘s actual value does not lie within this enlarged interval, a fact stating a deviation is entered. For instance, the following fact states that arterial mean pressure of patient 4999 is beyond the desired range:

deviation(pat4999, 0, 31, art, up)

We now want to use the pharmacological knowledge for deriving expected effects of an intervention on a particular patient. This is done by rules. The advantage of first-order logic is particularly important for modeling relations between intervals. For instance, stating that two time intervals are immediately succeeding, can be expressed by simply unifying the end point of one time interval with the start point of the other time interval. The following statement states, for instance, that two interventions were directly succeeding each other:

intervention(Patient, T1, T2, M, D1)
intervention(Patient, T2, T3, M, D2)

This statement can be instantiated by all patients, points in time, parameters and dosages as long as the same argument variable (e.g. *Patient*) is instantiated by the same value (e.g., *pat4711*) and different argument variables (e.g. *D1, D2*) are instantiated by different values.

intervention(pat4711, 73, 83, supra,0.05)
intervention(pat4711, 83, 177, supra, 0.02)

Intervals of dosages are handled in a similar manner. We can distinguish between major and minor changes of a dosage. A minor change is one within the same interval for which an effect has been stated by pharmacological facts. The rule and an actual instantiation is the following:

```
intervention(Patient, T1, T2, M,D1),
intervention(Patient, T2, T3, M,D2),
contains(M, S),
effect(S, FromD1, ToD1, Param, Trend),
FromD1=< D1 <ToD1, FromD1=< D2 <ToD1
-->
interv_effect(Patient,T2,T3,M,Param,Trend,minor)
```

```
intervention(pat4711,441,968,nitro,1.9),
intervention(pat4711,968,1081,nitro,2.38),
contains(nitro, glyceroltrinitrat),
effect(glyceroltrinitrat, 1, 10, hr, up),
1 =< 1.9 < 10, 1 =< 2.38 < 10
-->
interv_effect(pat4711,968,1081,nitro,hr,up,minor)
```

Changing into another such interval is a major change. The actual dosage of a drug given to a patient is compared with the dosage interval of *effect* facts. The following rule expresses the enforcement of an effect because of a major change of dosage.

```
intervention(Patient, T1, T2, M,D1),
intervention(Patient, T2, T3, M,D2),
contains(M, S),
effect(S, FromD1, ToD1, Param, Trend),
effect(S, FromD2, ToD2, Param, Trend),
FromD1 =< D1 < ToD1, FromD2 =< D2 < ToD2, D1 < D2
-->
interv_effect(Patient, T2,T3, M, Param, Trend, major)
```

Note, that if the substance *S* of drug *M* has a decreasing effect on a parameter of the patient, the rule predicts a further decrease of that vital sign. The variable *Trend* is then instantiated by *down*. Another rule states that decreasing a substance with an increasing effect on a parameter will decrease the parameter‘s value. We use such rules in order to predict effects of interventions. The prediction of intervention effects is used to check

interventions that are proposed by the learned decision rules. Not counting the patient records, the knowledge base consists of 39 rules and 88 facts.

7.2 Validating action-effect rules

In order to validate the knowledge base we applied it to the data of 148 patients. The data contain 8,200 interventions. The validation is easy, since rules can directly be applied to patient data. MOBAL’s inference engine derived 27,400 effects of the interventions using forward chaining. For 22,599 effects the atual effects in terms of level changes could be computed by the time series analysis (see section 5). When matching the derived effects with the actual ones, the system detected:

- 13,364 effects (i.e. 59.14%) took place in the restricted sense, that the patient's state remained stable. E.g., a drug with an increasing effect on a patient's vital sign does not lead to a significant level change of this parameter. This is not in conflict with medical knowledge, but shows best therapeutical practice. Smooth medication keeps the patient's state stable and does not lead to oscillating reactions of the patient.
- 5,165 effects (i.e. 22.85%) took place in the sense, that increasing or decreasing effects of drugs on vital signs match corresponding level changes.
- 4,070 contradictions (i.e. 18.01%) were detected. The observed level change of a vital sign went into the opposite direction of the knowledge-based prediction.

The ratio of 83.56 percent correct predictions of effects is quite positive. Some decisive features are not present in the data. Particularly the lack of data about cardiac arrhythmias and cardiac output could possibly explain many deviations of observed from predicted effects.

7.3 Integrating learned decision functions with the knowledge base

Since the goal of our work is an integrated system for intensive care monitoring, the numerical approach using the SVM has to be incorporated into the logic of MOBAL. While training SVM classifiers can take place offline in a separate program, MOBAL needs to be able to evaluate SVM decision rules and access the results online. We achieve this by introducing the special predicate *svm_calc/6* with the following semantic. The first two arguments indicate the patient and the drug. The third argument is either “up” or “down” depending on whether the *svm_calc* fact belongs to the SVM predicting dosis increase or decrease (compare section 6.3). The fourth argument is the time and the fifth is the current dosage of the drug. The last argument finally contains the value $\vec{w}_r \cdot \vec{o} + b_r$ of that particular SVM rule for the measurements $\vec{o}$ at that time. Calculating $\vec{w}_r \cdot \vec{o} + b_r$ can be done very efficiently, since it mainly consists of computing a dot product between the SVM weight vector $\vec{w}_r$ and the measurement vector $\vec{o}$. From each pair of decision rules (i. e. up and down) an intervention for the respective drug is recommended, if exactly one decision rule has a value $\vec{w}_r \cdot \vec{o} + b_r$ larger than a confidence threshold of 0.8.

$$h_{nitroup}(\vec{o}) = \operatorname{sgn}\left[\begin{pmatrix} 0.014 \\ 0.019 \\ -0.001 \\ -0.015 \\ -0.016 \\ 0.026 \\ 0.134 \\ -0.177 \\ -9.543 \\ -1.047 \\ -0.185 \\ 0.542 \\ -0.017 \\ 2.391 \\ 0.033 \\ 0.334 \\ 0.784 \\ 0.015 \end{pmatrix} \cdot \begin{pmatrix} Artsys & 174.00 \\ Artdia & 86.00 \\ Artmn & 121.00 \\ Cvp & 8.00 \\ Hr & 79.00 \\ Papsys & 26.00 \\ Papdia & 13.00 \\ Papmn & 15.00 \\ Nifedipine & 0.00 \\ Noradrenaline & 0.00 \\ Dobutamine & 0.00 \\ Dopamine & 0.00 \\ Glyceroltrinitrate & 0.00 \\ Adrenaline & 0.00 \\ Age & 77.91 \\ Emerg & 0 \\ BSA & 1.79 \\ Broca & 1.02 \end{pmatrix} - 4.368\right]$$

Figure 3. Decision rules for predicting an intervention that increases the dosage of Glyceroltrinitrat.

The decision rule for an increase of Glyceroltrinitrat (nitro) together with the actual para-

Table 7. Equivalence of decisions regarding effects.

Interventions	mean arterial pressure	heart rate	same effect all parameters	same behavior
Dobutamine	403	395	383	299
Adrenaline	407	406	393	374
Glyceroltrinitrate	437	388	380	342
Noradrenaline	436	428	424	420
Nifedipine	457	457	455	438

meter values $\vec{o}$ of patient 4999 at time 32 is shown in Figure 3. The dot product plus -4.368 (the value of *b*) is 1.85598. The fact entered into the fact base for patient 4999 is *svm_calc(pat4999, nitro, up, 32, 0.0, 1.85598)*. An intervention to increase nitro is derived. The dose is calculated on the basis of the former dose. The SVM actually only decides whether to increase, to decrease, or not to change the dose. For each drug, a level of granularity is defined. For instance, the granularity of Glyceroltrinitrat is 1, whereas that of Suprarenin (containing adrenaline) is 0.01. The dose is changed by just one step. In our example, the proposed intervention is:

pred_intervention(pat4999, 32, nitro, 1.0)

8 Using the knowledge base of effects to validate interventions

Medical knowledge is used for validation in two different ways. On the one hand, learned decision rules are validated on patient data by comparing the effects of their recommended interventions with the effects of actual physicians‘ interventions. This validation means to incorporate an evaluation step already into the knowledge acquisition phase. On the other hand, we believe that even an evaluated decision support system should check its decisions by considering their effects.

8.1 Validating learned decision rules

There are usually several different combinations of drugs that achieve the same goal of keeping the patient in a stable state. And indeed, different physicians, depending on their experience in the ICU, do use different mixtures and follow different strategies to reach this goal. For comparing treatment strategies, the real criterion is whether the recommendations have the same effect as the actual interventions. Therefore, we apply the action--effect rules from the knowledge base to both the proposed intervention of the SVM classifiers and to the intervention actually performed by the physician. If the derived effects are equal, then the proposed decision of the SVM classifiers can be considered as "equivalent" to the intervention executed by the physician. The results of this comparison for 473 interventions are shown in Table 7. The right-most column indicates the accuracy, i.e. in how many cases the classification of SVM and physician were identical (same behavior of SVM and physician). The other columns state how often the SVM's intervention leads to the same effects as the intervention of the physician. The first two columns show, how many of interventions had the same effect on arterial blood pressure or heart rate, respectively. The third column gives a more concise evaluation. Here it is stated, how many interventions recommended by the SVM had the same effects on all vital signs as the actual intervention. For instance, the SVM correctly classifies 299 test cases for Dobutamine (63%). If we compare the resulting effects of the predicted interventions concerning Dobutamine with the effects of the actual physician's interventions, we find that in 383 cases (81%) the deduced effects will be equal. Thus, in 84 cases the recommendation of the SVM does not match the physician's behavior, but the derived effects are the same, since the physician has chosen an "equivalent" drug or combination of drugs. An inspection of these cases helps to clarify issues of best practice and thus supports knowledge acquisition.

8.2 Validating proposed interventions

As depicted in the overall architecture (cf. Figure 1), we have chosen a design which allows us to use the action--effect rules in the knowledge base for validating predicted interventions. The underlying argument is that accuracy measures only reflect how well SVM's learning results fit actual behavior of the physician. However, we aim at best practice. Hence, we validate a proposed intervention with respect to its effects on the patient. If the effects push vital signs in the direction of the desired value range, the recommendation is considered sound, otherwise it is rejected. An example may clarify this. Patient 4999 is older than 75 years and stays at the ICU after a surgical operation. He suffers from high arterial mean pressure (around 124), where the heart rate is normal (around 80). Using its decision rules, the SVM recommends to increase Glyceroltrinitrat (see Figure 3). This proposed intervention is checked by the medical knowledge about

effects. The derived effects are an increase of the heart rate and a decrease of arterial mean pressure as well as left ventricular stroke work index (lvswi) and systemic vascular resistance (svr): *interv_effect(pat4999,32, T, art, down)*. The observed deviation is *deviation(pat4999, 0,31, art, up)*. Since *down* is the opposite of *up*, the proposed intervention is considered sound. In this way, the prescriptive medical knowledge (action-effect rules) is used to control the knowledge that is learned from actual therapies (state-action rules).

9 Comparison with related work

Using data from the most comprehensive singular clinical data repository at the LDS Hospital, Salt Lake City, Utah, USA, the group of Morris [26] developed a rule-based decision support system (DSS) for respiratory care in acute respiratory distress syndrome. Time is handled by introducing time points into the rules where a certain parameter value needs to be obtained. The development of this highly specialized system required more than 25 person years. It is a propositional rule base without a mechanism for consistency checking or matching rules and data. All validation efforts started only after the knowledge base had been completed.

Temporal reasoning is taken seriously in other developments [4, 7, 21, 28]. The Stanford approach uses an explicit time ontology for low-frequency data [28]. This approach is not feasable for our application. The VIE-VENT system is comparable with our approach in that it combines numerical data and a knowledge base [21]. Qualitative abstractions are derived for deviations of measurements from the target range. Time intervals refer to the validity of a measurement. The detection of outliers (data validation) is handled by a trend-based component. The validated measurements are used by the therapy planning component which aims at pushing vital signs into the value ranges of a stable state. Similar to our approach, therapy planning is divided into state-action rules (therapeutic actions based on status interpretation) and verifying the effectiveness of interventions. However, the system was developed without using actual patient data. Hence, the observation that parameter values oscillate considerably was made as late as the first clinical experience. In contrast, this observation has motivated our phase space procedure for abstracting from numerical time series. Temporal correlations can also be included in trend templates, which are used by Haimowitz and Kohane [7]. Trend templates consist of sets of low order polynomial regression models describing qualitative characteristics. Pattern abstraction is done based on the fit of these templates to the observed data. The major drawbacks of this method are the demand for predefined expected behavior and absolute value treshholds. However, time series in intensive care often show irregular behavior like patchy outliers, or outliers and level changes occuring in short time lags. Such behavior is difficult to specify in advance. Moreover, treshholds should be dynamically depending on the patient's status in the past. This has already been included in our approach, which does not need prespecified patterns either. Altogether, statistical time series analysis seems to be the most sophisticated method to model and investigate dynamical data since other approaches capture only parts of the time dependent structure of the data.

Our goals of easing the development of guidelines and validating the knowledge early on is shared by the two-step approach by Mani and coworkers [19]. They use machine learning in order to first characterize scores of dementia with respect to six categories (e.g., memory, orientation). These learning results are then used to learn the global clinical dementia rating. After a two years effort an efficient and effective system was accomplished. While the goals are the same, the application characteristics and, hence, the methods are completely different. The clinical rating is a classification task and the patient data is of qualitative nature, whereas our task is on-line monitoring and the patient data are time series of numerical measurements.

10 Conclusions

We presented an approach towards integrating statistical and knowledge-based methods for the development of decision support algorithms in critical care. This application involves high dimensional time series data, demanding high quality decision support under real time constraints. These properties make this case study a representative for a large number of applications in medicine and engineering.

This paper gives the necessary steps for solving this task as a whole. We identified how the application can be split up into manageable parts. We proposed an overall architecture that integrates a number of tasks, organized both sequentially and in parallel. All tasks are embedded in a single system, while selecting the most appropriate technique

and representation including the difficult effort of selecting and constructing appropriate features for each task individually.

The time series approach to data abstraction using phase space models was also validated in an independent clinical study. Here it showed a pattern recognition similar as that of an experienced intensivist [11]. The technical implementation proved highly efficient in data abstraction providing input for applications requiring qualitative information from time series.

The present hybrid system of statistical and machine learning methodologies is a typical example of how to integrate statistical analysis and knowledge discovery methods, i.e., time series analysis and machine learning, using their strengths and reaching better results than with each method alone.

The SVM was chosen for learning state-action rules due to its ability to handle multiple features. For modeling medical knowledge in terms of action-effect rules we chose a first-order logic representation using MOBAL. This allowed a compact representation of medical knowledge with a small number of rules, fulfilling the real-world demand for a knowledge base to be understandable by humans and accessible for expert validation. Current work deals with the interactions of diverse medications. The rules combining opposite effects of different drugs are not sufficient yet.

The validation issue has been treated with special care. Each process has been validated in the standard way, i.e. tested on data not used for training. In addition, the results of state-action rules were compared with the results of a human expert who classified the same data. Moreover, recommended interventions of state-action rules are validated by formalized medical knowledge. On the one hand, the effect of a recommended intervention is compared with the effect of an actual intervention. Of course, this comparison can only be made for past cases. In case of conflict, the expert inspects the particular cases. This may lead to the generation of explicit additional knowledge. On the other hand, the formalized effects of interventions are applied to current cases and evaluated with respect to the target ranges of vital signs.

Our new approach combines modeling of expert knowledge with data-driven methods. This eases the task of building operational protocols. Moreover, the data-driven method allows for an ongoing enhancement of the knowledge base on the basis of current practice. The knowledge base is validated against existing patient data. This approach is meant to be significantly more effective than the tedious, time-consuming, and costly process of traditional development of on-line operational decision support systems.

11 Acknowledgements

This work has been funded in part by the collaborative research center "Complexity Reduction in Multivariate Data Structures" (DFG, SFB475). We thank the reviewers for valuable comments.

12 References

[1] M. Bauer, U. Gather, and M. Imhoff , Analysis of high dimensional data from intensive care medicine, in: R. Payne, ed., *Proceeding in Computational Statistics* (Springer-Verlag, Berlin, 1999).

[2] M. Bauer, U. Gather, and M. Imhoff, The identification of multiple outliers in online monitoring data. Technical Report 29, SFB 475, University of Dortmund, 1999.

[3] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis. Forecasting and Control.* Third Edition (Prentice Hall, Englewood Cliffs, 1994).

[4] M. Dojat and C. Sayettat, A Realistic Model for Temporal Reasoning in Real-Time Patient Monitoring, *Applied Artificial Intelligence* (1995), 121-143.

[5] A. J. Fox, Outliers in Time Series, *J. Roy. Stat. Soc. B 3* (1972) 350-363.

[6] G. Guyatt, M. Drummund, D. Feeny, P. Tugwell, G. Stoddart, R. Haynes, K. Bennett, and R. LaBelle, Guidelines for the clinical and economic evaluation of health care technologies, *Soc Sci Med 22* (1986) 393-408.

[7] I. J. Haimowitz, and I. S. Kohane, Managing temporal worlds for medical trend diagnosis, *Artificial Intelligence in Medicine 8* (1996), 299-321.

[8] M. Imhoff, A clinical information system on the intensive care unit: dream or night mare?, in: J. A. G. Rubi, ed., *Medicina Intensiva 1995, XXX. Congreso SEMIUC. Murcia*. (Pictographia, Murcia, 1995) 17-22.

[9] M. Imhoff, 3 years clinical use of the Siemens Emtek System 2000: Efforts and Benefits, *Clinical Intensive Care 7 (Suppl.)* (1996) 43-44.

[10]M. Imhoff, Clinical Data Acquisition: What and how?, *Journal für Anästhesie und Intensivmedizin 5* (1998) 85-86.

[11]M. Imhoff, M. Bauer, U. Gather, and D. Löhlein, Statistical pattern detection in univariate time series of intensive care on-line monitoring data, *Intensive Care Med 24* (1998) 1305-1314.

[12]M. Imhoff, M. Bauer, and U. Gather, Time-effect relations of medical interventions in a clinical information system, in: W. Burgard, T. Christaller, and A. B. Cremers, eds., KI-99: Advances in Artificial Intelligence, LNAI 1701 (Springer-Verlag, 1999) 307-310.

[13]M. Imhoff, J. H. Lehner, and D. Löhlein, 2 years clinical experience with a clinical information system on a surgical ICU, in: N. J. Mutz, W. Koller, and H. Benzer, eds., 7*th European Congress on Intensive Care Medicine.* (Monduzi Editore, Bologna, 1994) 163-166.

[14]V. Isham, Statistical aspects of chaos: a review, in: O. E. Barndorff-Nielsen, W. S. Jensen, and W. S. Kendall, eds., *Networks and Chaos - Statistical and Probabilistic Aspects* (Chapman and Hall, London, 1993) 251-300.

[15]D. Jennings, T. Amabile, and L. Ross, Informal covariation assessments: Data-based versus theory-based judgements, in: D. Kahnemann, P. Slovic, and A. Tversky, eds., *Judgement under uncertainty: Heuristics and biases* (Cambridge University Press, Cambridge, 1982) 211-230.

[16]T. Joachims, Making Large-Scale SVM Learning Practical, in: B. Schölkopf, C. Burges, and A. Smola, eds., *Advances in Kernel Methods -- Support Vector Learning* (MIT Press, Cambridge, 1999) 169-184.

[17]E. T. Keravnou, Temporal diagnostic reasoning based on time-objects, *Artificial Intelligence in Med 8* (1996), 235-265.

[18]M. Kukar, I. Kononenko, C. Groselj, K. Kralj, and J. Fettich, Analysing ad Improving the Diagnosis of Ischaemic Heart Disease with Machine Learning, *Artificial Intelligence in Med 16* (1999), 25-50.

[19]S. Mani, W.R. Shankle, M. B. Dick, and M. J. Pazzani, Two-Stage Machine Learning for Guideline Development, *Artificial Intelligence in Med 16* (1999), 51-71.

[20]J. M. McDonald, S. Brossette, and S. A. Moser, Pathology information systems: data mining leads to knowledge discovery, *Arch Pathol Lab Med 122* (1998) 409-411.

[21]S. Miksch, W. Horn, C. Popow, and F. Paky, Utilizing Temporal Abstraction for Data Validation and Therapy Planing for Artificially Ventilated Newborn Infants, *Artificial Intelligence in Medicine 8* (1996), 543-576.

[22]G. Miller, The magical number seven, plus or minus two: Some limits to our capacity for processing information, *Psychol Rev 63* (1956) 81-97.

[23]K. Morik, Balanced Cooperative Modeling, in: R. Michalski, G. Tecuci, eds. Machine Learning -- A Multistrategy Approach (Morgan Kaufmann, 1994) 295-318.

[24]K. Morik, S. Wrobel, J.-U. Kietz, and W. Emde, *Knowledge Acquisition and Machine Learning -- Theory, Methods, and Applications* (Academic Press, London, 1993).

[25]A. Morris and R. Gardner (1992): Computer applications, in: J. Hall, G. Schmidt, and L. Wood, eds., *Principles of Critical Care* (McGraw-Hill, New York, 1992) 500-14.

[26]A. Morris, Algorithm-Based Decision-Making, in: J. A. Tobin, ed., *Principles and Practice of Intensive Care Monitoring* (McGraw-Hill, New York, 1998) 1355-1381.

[27]N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, Geometry from a time series, *Phys Rev Lett 45* (1980) 712-716.

[28]Y. Shahar and M. Musen, Knowledge-Based Temporal Abstraction in Clinical Domains, *Artificial Intelligence in Med* (1996), 267-298

[29]W. Schmid, CUSUM control schemes for Gaussian processes, *Statistical Papers 38* (1997) 191-217.

[30]V. W. Soo, J. S. Wang, and S. P. Wang, Learning and discovery from a clinical database: an incremental concept formation approach, *Artificial Intelligence in Med 6* (1994) 249-261.

[31]K. D. C. Stoodley and M. Mirnia, The Automatic Detection of Transients, Step Changes and Slope Changes in the Monitoring of Medical Time Series, *The Statistician 28* (1979) 163-170.

[32]F. Takens, Detecting strange attractors in turbulence, in: D. A. Rand and L. S. Young, eds., Dynamical systems and turbulence, *Lecture Notes in Mathematics 898* (Springer, Berlin, 1980) 366-381.

[33]G. E. Thomsen, D. Pope, T. D. East, A. H. Morris, A. T. Kinder, D. A. Carlson, G. L. Smith, C. J. Wallace, J. F. Orme Jr, T. P. Clemmer, et al., Clinical performance of a rule-based decision support system for mechanical ventilation in ARDS patients, *Proc Annu Symp Comput Appl Med Care* (1993) 339-343.

[34]H. Tong, A personal overview of non-linear time series analysis from a chaos perspective, *Scand J Statist 22* (1995) 399-445.

[35]R. S. Tsay, Outliers, Level Shifts, and Variance Changes in Time Series, *J. Forecasting 7* (1988) 1-20.

[36]S. Tsumoto,W. Ziarko, N. Shan, and H. Tanaka, Knowledge discovery in clinical databases based on variable precision rough set model, *Proc Annu Symp Comput Appl Med Care* (1995) 270-274.

[37]S. Uckun, B.M. Dawant, and D.P. Lindstrom, Model-Based Diagnosis in Intensive Care Monitoring -- the YAQ Approach, *Artificial Intelligence in Med 5* (1993), 31-48.

[38]V. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).

[39]F. A. da Veiga, Structure discovery in medical databases: a conceptual clustering approach, *Artificial Intellligence in Med 8* (1996) 473-491.