Modeling the Cognitive Task Load and Performance of Naval Operators

Mark A. Neerincx[1,2], Stefan Kennedie[1,5], Marc Grootjen[2,3], Franc Grootjen[4,5]

[1]TNO Human Factors, Kampweg 5, 3769 ZG Soesterberg, The Netherlands
[2]Delft University of Technology, Mekelweg 4, 2628 GA Delft, the Netherlands
[3]Defense Materiel Organization, Directorate Materiel Royal Netherlands Navy, P.O. Box 20702, 2500 ES The Hague, the Netherlands
[4]Donders Centre for Cognition, Radboud University Nijmegen, P.O. Box 9104, 6500 HE, Nijmegen, The Netherlands
[5]Radboud University Nijmegen, the Netherlands
mark.neerincx@tno.nl, Kennedie@gmail.com, marc@grootjen.nl, grootjen@acm.org

Abstract. Operators on naval ships have to act in dynamic, critical and high-demand task environments. For these environments, a cognitive task load (CTL) model has been proposed as foundation of three operator support functions: adaptive task allocation, cognitive aids and resource feedback. This paper presents the construction of such a model as a Bayesian network with probability relationships between CTL and performance. The network is trained and tested with two datasets: operator performance with an adaptive user interface in a lab-setting and operator performance on a high-tech sailing ship. The "Naïve Bayesian network" tuned out to be the best choice, providing performance estimations with 86% and 74% accuracy for respectively the lab and ship data. Overall, the resulting model nicely generalizes over the two datasets. It will be used to estimate operator performance under momentary CTL-conditions, and to set the thresholds of the load-mitigation strategies for the three support functions.

Keywords: mental load, emotion, Bayesian networks, cognitive engineering, Defense and Space operations.

1 Introduction

Crews on naval ships have to operate in dynamic, critical and complex task environments, which impose high fluctuations of the required cognitive resources. These resources are constrained and may not fit the momentary task demands, resulting in performance decrements. To mitigate such load bottlenecks, three operator support functions are being developed: adaptive task allocation, cognitive aids and resource feedback [1, 2, 3]. Important foundations of these support functions are *situated* theories on cognitive task load (CTL) and emotional state (ES) [4]. Such theories include accepted features of cognition such as limited processing capacity, are validated in the context of a specific domain and possibly group of task performers, and provide predictions of the task performance within this domain.

Consequently, they can provide the "context-awareness" for the proposed support functions. Face validity is required to realize adequate trust and involvement of users. This paper presents the construction of a Bayesian network model for CTL as refinement of a situated theory on naval operators' information processes.

1.1 Cognitive Task Load

The cognitive task load (CTL) theory distinguishes three load dimensions. The first dimension is the *time occupied*, which is high when the operator has to work with maximum cognitive processing speed to search and compare known visual symbols or patterns, to perform simple (decision-making) tasks, and to manipulate and deal with numbers in a fast and accurate way. With respect to the second dimension, the *level of information processing*, (a) information that is processed automatically, results into actions that are hardly cognitively demanding, (b) routine procedures involve rather efficient information processing, and (c) problem solving and action planning for relatively new situations involve a heavy load on the limited capacity of working memory. *Task-Set Switches* is the third load dimenssion, addressing the demands of attention shifts or divergences in which different sources of human task knowledge have to be activated. It should be noted that the effects of cognitive task load depend on the concerning task duration. In general, the negative effects of under- and overload increase over time.

1.2 Emotional State

Neerincx [4] proposes to combine the CTL-model with a model of the Emotional State (ES) for high-demand task domains in which the human sometimes works in extreme and critical conditions. The ES-model distinguishes two dimensions: the arousal level—low versus high—and the valence level—positive versus negative [5]. Emotion and CTL are related: for specific load conditions a specific emotional state ("response") can be expected. For example, when task load increases, an adequate response is to invest extra effort (i.e., arousal increases) in order to maintain good performance [3].

1.3 Model Levels

For the CTL-ES model, we distinguish three levels (Fig 1). The first level describes the *human act observables*, which are behavioral and bodily variables that correlate with human information processes (HIP).

At the second level, *HIP dimensions* represent variables that correlate with human performance. SOWAT, an activity monitoring tool, can be used to derive the CTL-dimensions' values from observables as user-interface acts [2], while affective computing techniques can be used to derive the ES-dimensions' values from, for example, facial and speech expressions [6]. An operator profile can be applied for personalized estimation of HIP-dimensions' values from observables. For example,

the level of experience influences the Level of Information Processing (LIP): the higher the experience, the lower the LIP value. The dimensional model is trained in advance by datasets that include performance measures. This estimation may concern the current performance and the near-future performance.

At the third level, *HIP classes* are derived from the dimensional models. CTL-classes are underload (UL), overload (OL), vigilance (VI), cognitive lock-up (CL), and neutral (NE); ES-classes are boredom (BO), relaxed (RE), excited (EX), stressed (ST), and neutral (NE).

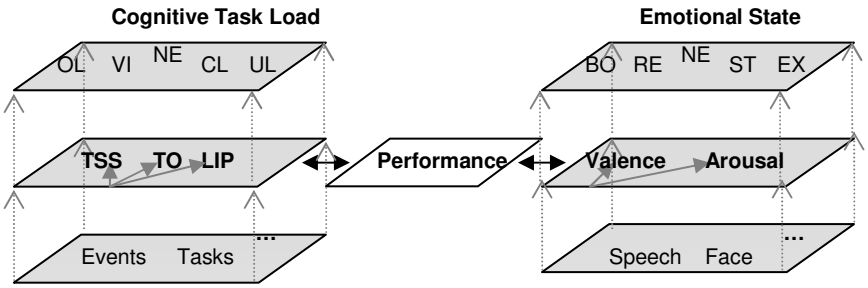

Fig. 1. The Performance, Cognitive Task Load and Emotional State model.

1.4 Performance Estimation

This paper focuses on the construction of the dimensional CTL-model (i.e., the 2nd level of Fig. 1). For this purpose, we need a method to analyze data from training and actual task performances, which can cope with missing data. Furthermore, it should be easy to extend the model, for example, starting with CTL-dimensions and adding ES-dimensions when appropriate. In addition, the model should be transparent (i.e., providing a structure that gives insight in which variable influences other variables), enabling estimations of near-future values. Bayesian networks seem to fulfill these requirements. This paper investigates whether a Bayesian network can be constructed that provides adequate estimations of the CTL-performance relationships for two datasets: operator performance on a high-tech sailing ship and operator performance with an adaptive user interface in a lab-setting.

2 Bayesian Networks

Bayesian networks are graphical models for reasoning under uncertainty. A Bayesian network consists of a network structure and conditional probability tables. The structure of a Bayesian network consists of nodes and arcs. The nodes represent variables, and the arcs represent direct dependencies between the variables. If there is an arc from one node to another, then the first node is called the parent of the latter (the child). The structure of a Bayesian network is a directed acyclic graph (DAG). In other words, the structure does not contain any cycles. Each node has a conditional

probability table. This table defines the probabilities of that node on taking each of its values, given its parent(s). Bayesian networks are often applied in the medical domain. Given symptoms, the Bayesian network can compute the probability of the presence of a disease using Bayes' Theorem (see Equation 1).

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{1}$$

This is called Bayesian inference and can be explained with the hypothetical network structure of Fig. 2d as example, in which performance relates to TO, TSS and LIP. Table 1 shows possible conditional probability tables. If there is evidence that a certain person has low performance, the probability that this person experiences high TO, TSS and LIP can be read in the tables. These probabilities are respectively 0.5, 0.4 and 0.6. The other way around, it is possible to calculate the probability that a person has a low performance when high TO, TSS and LIP are observed. This can be done using Bayes' Theorem:

$$\begin{aligned}
&P(\text{low performance with high TO, TSS and LIP}) \\
&= P(\text{PERF} = l \mid \text{TO} = h \wedge \text{TSS} = h \wedge \text{LIP} = h) \\
&= \frac{P(\text{TO} = h \wedge \text{TSS} = h \wedge \text{LIP} = h \mid \text{PERF} = l) \cdot P(\text{PERF} = l)}{P(\text{TO} = h \wedge \text{TSS} = h \wedge \text{LIP} = h)} \\
&= \frac{P(\text{TO} = h \wedge \text{TSS} = h \wedge \text{LIP} = h \mid \text{PERF} = l) \cdot P(\text{PERF} = l)}{\sum_{p \in \{l,m,h\}} P(\text{TO} = h \wedge \text{TSS} = h \wedge \text{LIP} = h) \mid P(\text{PERF} = p)} \\
&= \frac{0.5 \cdot 0.4 \cdot 0.6 \cdot 0.3}{0.5 \cdot 0.4 \cdot 0.6 \cdot 0.3 + 0.5 \cdot 0.1 \cdot 0.5 \cdot 0.4 + 0.1 \cdot 0.0 \cdot 0.1 \cdot 0.3} \\
&= \frac{0.036}{0.036 + 0.01 + 0} \\
&= 0.783
\end{aligned}$$

Table 1. Possible conditional probability tables for the network structure of Fig. 2d.

Performance		
low	medium	high
0.3	0.4	0.3

	TSS		
Performance	low	medium	high
low	0.3	0.3	0.4
medium	0.6	0.3	0.1
high	0.6	0.4	0.0

	TO		
Performance	low	medium	high
low	0.0	0.5	0.5
medium	0.1	0.4	0.5
high	0.7	0.2	0.1

	LIP		
Performance	low	medium	high
low	0.0	0.4	0.6
medium	0.1	0.4	0.5
high	0.7	0.2	0.1

3 Experiment: Analysis of two Datasets

To create a Bayesian Network for Performance and Cognitive Task Load, we analyzed two datasets: the first dataset was automatically collected during operator's interaction with a prototype user interface, and the second dataset was manually collected during operator's performance on a sailing ship.

3.1 Lab Dataset

The Lab data were acquired during an experiment at the MBO Shipping & Transportation College of Rotterdam (for details, see [7]). 12 students participated, all second and third year students (average age of 20.1 with a standard deviation of 2.1, 11 males, 1 female; relevant knowledge about the maritime domain). All participants had to deal with alarms during platform supervision, damage control and navigation tasks. All performed actions were recorded in log files and used to calculate TO, TSS, LIP and performance, with use of SOWAT [2].

The Lab data contained 1407 cases with data for LIP, TSS, TO and Performance. Each case in the data file corresponds to a sliding window of 60 seconds with 50 seconds overlap. The values for LIP range from 0 (low) to 6.5 (high), TSS ranges from 0 to 5, TO ranges from 0% to 100%, and performance ranges form 0 (low) to 4 (high). All values of the variables were converted to the values low, medium and high for our analyses. Since Bayesian networks are best trained with data that have an equal distribution, we have chosen the thresholds to accomplish this as much as possible (see Table 2 for the distribution).

From this data file we created a balanced train and test set. We have selected 333 cases with low performance randomly form the total of 427 cases with low performance, and did the same for medium and high performance. The test set contained 150 cases, also with an equal distribution that was randomly selected. The other 258 cases were not used for training or testing since this would result in unbalanced train and test sets.

Table 2. Distribution of cases over CTL and Performance for the two datasets.

	Lab data				Ship data			
	TO	TSS	LIP	Perf.	TO	TSS	LIP	Perf.
Low	476	722	462	427	571	1123	426	373
Medium	460	425	468	398	599	378	591	390
High	471	260	477	582	582	251	735	989

3.2 Ship Dataset

The Ship data were acquired during an experiment in the Ship Control Centers of three sailing air defense and command frigates (for details, see [8]). Each ship was manned with four active duty teams, data collection concerned two persons of each team. In total there were 12 teams and 24 participants (all male). Each team had to

perform three scenarios that varied in TO, TSS and LIP. All scenarios were recorded on video and scored by experts afterwards on TO and LIP. LIP was scored by the participants themselves. SOWAT [2] was used for integration of all data and generation of 1752 cases. Each case in the data file corresponds to a sliding window of 60 seconds with 40 seconds overlap. The values for LIP range form 1 (low) to 5 (high), TSS ranges from 0 to 6, TO ranges from 0% to 100%, and performance ranges form 0 (incorrect or too slow response) to 2 (correct response). All values of the variables were converted to the values low, medium and high. For this dataset we have also chosen the thresholds to accomplish an equal distribution as much as possible (see Table 2 for the distribution).

From this data file we created a train set and a test set. The train set contained 969 cases with an equal distribution of performance. The cases were also randomly selected. The test set contained 150 cases, also with an equal distribution that was randomly selected. The other 633 cases were not used for training or testing.

3.3 Creating the Network Structure

When creating a Bayesian network, the structure of the network can either be defined by an expert, or learned from a dataset. We used GeNIe 2.0[1] to create four network structures for each dataset. GeNIe is equipped with four structure learning algorithms:

- Essential Graph Search (EGS) algorithm [9]
- PC algorithm [10]
- Greedy Thick Thinning (GTT) algorithm
- Naïve Bayesian network (NBN) algorithm [11]

After creating the network structures we created the conditional probability tables using Netica-J's[2] parameter learning algorithm. This algorithm was applied to the same train sets that were used for structure learning.

Finally, the performance of the created Bayesian networks was tested with the test sets using Netica-J's performance testing algorithm. These results were evaluated using a Chi-square test.

3.4 Results

This section first shows the results for the Lab and Ship datasets, then discusses the generalizability of the networks.

3.4.1 Lab Data

The network structures that were created by the four structure learning algorithms using the Lab train set are, with the exception of the NBN algorithm, very similar. The first three algorithm produce a fully connected network structure, the only difference is the direction of the arcs (Fig. 2).

[1] http://genie.sis.pitt.edu/
[2] http://www.norsys.com/

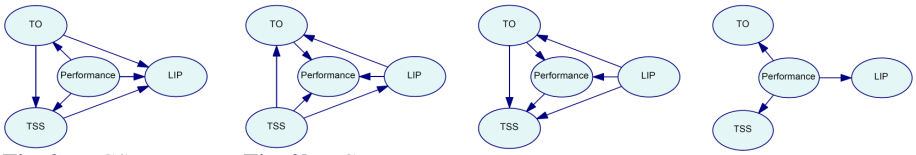

Fig. 2a. EGS **Fig. 2b.** PC **Fig. 2c.** GTT **Fig. 2d.** NBN

Fig. 2. The created network structures for the Lab dataset using the four algorithms.

As a result of the similarity in the network structure of Fig. 2, the percentages of cases classified correctly are the same for these three algorithms. All four algorithms perform overall significantly better than random guessing a performance level (all $p<0.000$). The NBN algorithm performed overall slightly better than the other algorithms, but this difference was not significant ($p<0.816$). When we zoom in to the different performance categories, we see that de difference between random guessing and the NBN algorithm is significant for low performance ($p<0.014$). For medium performance, the difference between the EGS, PC and GTT algorithms and random guessing are significant ($p<0.008$). Finally, for high performance the difference between the four algorithms and random guessing is significant ($p<0.001$).

When we look at the network with the highest performance in detail, we see that it is not able to distinguishing well between low and medium performance (see Table 3, left). When we join the performance categories low and medium together, the percentage correct classified increased from 58% to 85% (see table 3, middle), while the expectation value ("random") increased to from 33% to 50%. A drawback of this method is that the dataset is not distributed equally for performance. To accomplish an equal distribution we adjusted the threshold for performance. The Bayesian network was trained with a train set that consisted of 500 cases with low and 500 cases with high performance. The network was tested with a test set that contained 50 cases with low and 50 cases with high performance. This network classified 86% of the cases correct. More importantly, all cases with low performance were recognized, see Table 3, right. This Table is the same for all network structures that were tested. In other words, all network structures perform the same, see Fig. 3, right.

Table 3. Performance of the networks with the highest percentage correct classified with three (left) and two performance levels, unbalanced (middle) and balanced (right).

	Prediction		
Actual	low	medium	high
low	29	21	0
medium	20	23	7
high	10	5	35

	Prediction	
Actual	low	high
low	93	7
high	15	35

	Prediction	
Actual	low	high
low	50	0
high	14	36

The networks that were trained with two performance categories performed overall better than the networks that were trained with three performance categories, even after correction for chance using Cohen's Kappa.

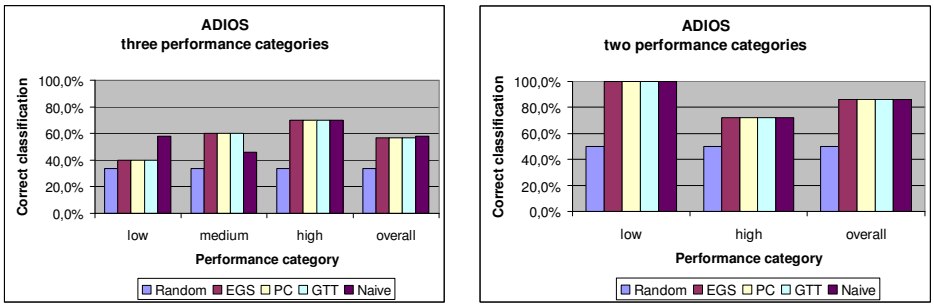

Fig. 3. Network performance of the different algorithms for the for the Lab dataset with three (left) and two (right) performance levels (balanced).

3.4.2 Ship Data

The network structures that were created by the four structure learning algorithms using the Ship train set show more variation (Fig. 4) than we have seen with the Lab dataset (Fig. 2). The structures are not fully connected and with the exception of the NBN algorithm, there is no direct dependence between TSS and performance.

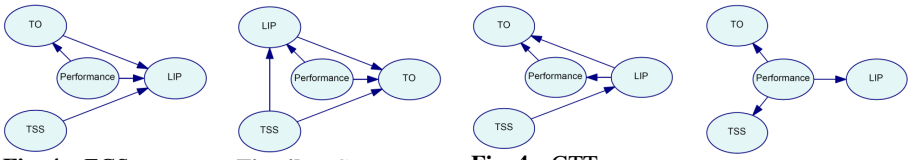

Fig. 4a. EGS **Fig. 4b.** PC **Fig. 4c.** GTT **Fig. 4d.** NBN

Fig. 4. The created network structures for the Ship dataset using the four algorithms.

As a result of the variation in network structure, the percentages of cases classified correctly differ considerable (Fig. 5, left). All four algorithms perform overall significantly better than random guessing a performance level ($p<0.000$ for the EGS and PC algorithm, $p<0.005$ for the GTT algorithm and $p<0.045$ for the NBN algorithm). The PC algorithm shows the best performance, but does only perform significantly better than the NBN algorithm ($p<0.029$).

The Ship dataset was also tested with two performance levels (Fig. 5, right). The percentage classified correct of the best network increased from 57% to 76%, while the expectation value ("random") increased to from 33% to 50%.

The networks that were trained with two performance categories performed overall better than the networks that were trained with three performance categories, even after correction for chance using Cohen's Kappa.

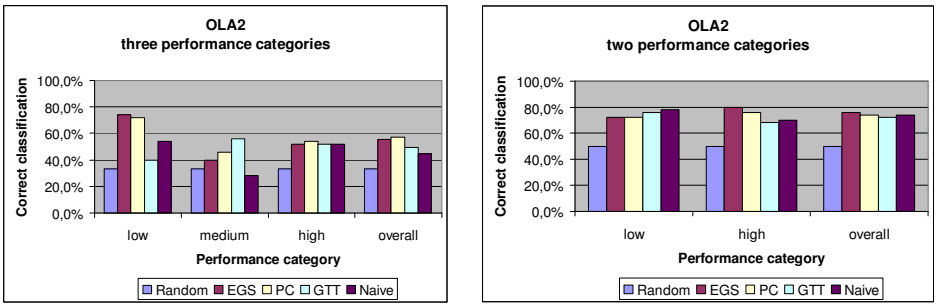

Fig. 5. Network performance of the different algorithms for the for the Ship datasets with three (left) and two (right) performance categories (balanced).

3.5 Generalizability

To test the generalizability of the networks, we tested the performance of the networks that were trained with the Ship train-set with the Lab test-set and vice versa (Table 4). When these results are compared with the results of the networks that have been tested with the same datasets as they were trained, we see that almost all differences are not significant. The only exception is the network that was created with the PC algorithm using the Lab data with two performance categories, and tested with the Ship data ($p<0.009$).

Table 4. Cross dataset testing.

		Correct classification (%)							
		3 performance categories				2 performance categories			
Train set	Test set	EGS	PC	GTT	NBN	EGS	PC	GTT	NBN
Lab	Lab	56.7	56.7	56.7	58.0	86.0	86.0	86.0	86.0
Ship	Lab	56.0	53.3	54.0	56.0	79.0	81.0	84.0	84.0
Lab	Ship	56.7	56.7	56.7	58.0	70.0	63.0	71.0	74.0
Ship	Ship	55.3	57.3	49.3	44.7	76.0	74.0	72.0	74.0

4 Conclusions and Discussion

Previous research showed the effects of CTL on operator task performance, and possible mitigation methods (adaptive task allocation, cognitive aids and resource feedback). This paper provides the first results on applying Bayesian Networks to model these effects in order to estimate and predict possible performance shortcomings. We derived the CTL-performance relationships for two datasets: operator performance with an adaptive user interface in a lab-setting and operator performance on a high-tech sailing ship (Ship). The first dataset provides the best results, probably because the recording was conducted in rather controlled conditions and all three CTL-factors showed variance in the scenario. In contrast, the dataset of

the sailing ships contained relatively few Task-Set Switches (TSS), which might explain the creation of network structures that do not include a direct relationship of TSS with Performance (see Fig. 4). However, the "Naïve Bayesian Network" model that is trained with the more-balanced Lab dataset proves to provide similar performance prediction results for the Ship dataset as the models that are derived from the Ship training dataset (i.e., for the two category performance, see Table 4). So, the "Naïve Bayesian Network" algorithm seems to be a good choice, providing performance estimations with 86% and 74% accuracy for respectively the lab-setting and sailing ship data (with respectively a 100% and 78% hit-rate for the low performance category). Overall, the resulting model nicely generalizes over the two datasets. Although the results are relatively positive, there is a clear room for improvement. Currently, we are extending the modeling approach with emotion, both for the defense and the space domain. A major question is how to adequately address the occurrence of very rare cases for which the dataset is not trained? A method to detect such occurrences would be very beneficial.

References

1. Neerincx, M.A. (2003). Cognitive task load design: model, methods and examples. In: E. Hollnagel (ed.), *Handbook of Cognitive Task Design*. Chapter 13 (pp. 283-305). Mahwah, NJ: Lawrence Erlbaum Associates.
2. Grootjen, M., Neerincx, M.A., Stolk, K.D., Weert, J.C.M. van & Bierman, E.P.B. (2007). Design and user evaluation of an interface prototype that adapts to the operator's cognitive task load. In: D.D. Schmorrow, D.M. Nicholson, J.M. Drexler & L.M. Reeves (Eds), *Proc. 4th Intern. Augmented Cognition*. Arlington, Virginia: Strategic Analysis, Inc. pp. 97-106.
3. Neerincx, M.A., Bos, A., Olmedo-Soler, A. Brauer, U. Breebaart, L., Smets, N., Lindenberg, J., Grant, T., Wolff, M. (2008). The Mission Execution Crew Assistant: Improving Human-Machine Team Resilience for Long Duration Missions. In: *Proc. of the 59th International Astronautical Congress (IAC2008)*, 12 pages. Paris, France: IAF. DVD: ISSN 1995-6258
4. Neerincx, M.A. (2007). Modelling Cognitive and Affective Load for the Design of Human-Machine Collaboration. In: *Engineering Psychology and Cognitive Ergonomics*, HCII 2007, pp. 568-574. Berlin: Springer-Verlag.
5. Truong, K.P, van Leeuwen, D.A. & Neerincx, M.A. (2007). Unobtrusive Multimodal Emotion Detection in Adaptive Interfaces: Speech and Facial Expressions. In: D.D. Schmorrow & L.M. Reeves (Eds), *Foundations of Augmented Cognition*, 3rd ed., LNAI 4565 proceedings, pp. 354-363.
6. Bradley, M., Lang, P. (1994). Measuring emotion: The Self-Assessment Manikin and the Semantic Differential. *Journal of Behavioral Therapy & Experimental Psychiatry* 25, 49-59.
7. Grootjen, M., Reijenga, Y. & Neerincx, M.A. (to appear). A user evaluation of an interface that adapts to the user's cognitive task load.
8. Grootjen, M., Greef, T. de & Neerincx, M.A. (to appear). Effects of Level of Automation on Operator Task Load and Performance on a Sailing Naval Ship.
9. Dash, D., Druzdzel, M. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. In: *Proc. 15th Annual Conference on Uncertainty in Artificial Intelligence* (UAI-99), pp. 142—149. San Francisco, CA: Morgan Kaufmann.
10. Spirtes, P., Glymour, C.N., Scheines, R., Causation (2000). Prediction, and Search. Cambridge: MIT Press.
11. Duda, R.O., Hart P.E. (1973). Pattern Classification and Scene Analysis. Wiley, New York.