Comparison of different inference algorithms for medical decision making*

Guven Kose
Information Management Department, Hacettepe University
Ankara, 06800, Turkey

Hayri Sever
Computer Engineering Department, Hacettepe University
Ankara, 06800, Turkey

Mert Bal[†]
Mathematical Engineering Department, Yildiz Technical University
Istanbul, 34220, Turkey

Alp Ustundag
Industrial Engineering Department, Istanbul Technical University
Istanbul, 34367, Turkey

Abstract

A medical diagnosis system (DRCAD), which consists of two sub-modules Bayesian and rule-based inference models, is presented in this study. Three types of tests are conducted to assess the performances of the models producing synthetic data based on the ALARM network. The results indicate that the linear combination of the aforementioned models leads to a 5% and a 30% improvement in medical diagnosis when compared to the "Rule Based Method" and the "Bayesian Network Based Method", respectively.

Keywords: Medical Decision Support Systems, Bayesian Networks, Rule-Based Systems, ALARM Network

1. Introduction

A Decision Support System (DSS) is a computer-based information system that supports organizational and business decision-making activities. Medical Decision Support Systems, which are variants of decision support systems, are intelligent software systems that are designed to improve clinical diagnosis system and to support the healthcare personnel in their decision. Intelligent decision support systems use artificial intelligence system techniques to support the healthcare personnel for selecting the best method for both diagnosis and also for treatment especially when the information about the treatment is incomplete or uncertain. These systems can work in both active and passive modes. When they are in passive mode, they will be used only when they are required. When they are in active mode, they will be making recommendations as well. When we look at the approaches of the inference mechanisms, which constitute the most important part of the medical decision support systems, these approaches can be divided into two parts such as rule-based systems and data-driven systems.

Rule-based systems are constructed on the knowledge base, which are formed by if-then structures. In this structure, the information base is formed by the rules. The operation logic of the system is to find relevant rules on the basis of the available information operate them and continue to search for a rule until a result has been obtained. Those rule-based systems have some strong features as well as some disadvantages. For example, the performance of the system decreases and the maintenance of the system becomes difficult in case of the number of the rules are large enough. Examples of the medical decision support systems are MYCIN[1,2], TRAUMAID[3], and RO[2]SE[4]. Data-driven systems, on the other hand, operate in large data stacks and support the decision-making process using data mining methods. Several studies can be found in the related literature about data-driven systems. Some of these studies can be referred to Bayesian networks[5], Rough sets[6] and artificial neural networks[7]. Data-driven systems are more flexible compared to the rule-based systems and they have the ability to learn by themselves.

The aim of this paper is to evaluate the performance of the Bayesian and rule-based inference models on medical data using a medical diagnosis system called as DRCAD. In this study, three types of tests are conducted to assess the standalone performances of the Bayesian model, the rule-based model, and the linear combination of these two separate methods. For these evaluations, synthetic data have been produced based on the probabilities on the ALARM (A Logical Alarm Reduction Mechanism) network - a structure prepared by using real patient information for many variables showing the probabilities derived from the real life circumstances. Performance evaluation is repeated for 100, 1.000 and 2.000 batches of data sets to validate the accuracy of the results. This paper also contributes to the relevant literature by presenting a medical decision support system called DRCAD system with two sub-modules comprised of Bayesian and rule-based inference models.

The rest of this paper is organized as follows. Section 2 provides an overview of decision support systems. In Section 3, Bayesian networks are explained briefly. In Section 4, the ALARM Network structure is described. In Section 5 and 6, the Rule Based Method and the Bayesian Network Method Testing Scenarios are explained, respectively. Section 7 presents the experimental results of DRCAD Medical Diagnosis Software. Finally, Section 8 concludes.

2. Decision support systems

DSSs are interactive computer-based systems or subsystems that are designed to help decision makers to decide and complete the decision process operations and also to determine and solve problems using communication technologies, information, documents and models. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model-based reasoning. They support framing, modeling and problem solving. Typical application areas of DSSs are healthcare, management and planning in business, the military and any area in which management will encounter complex decision situations. DSSs are typically used for strategic and tactical decisions faced by upper-level management -decisions with a reasonably low frequency and high potential consequences- in which the time taken for thinking

through and modeling the problem pays off generously in the long run.

Generally, decision support systems should include the following features:

- DSSs are used to support the decision making process not to accomplish operational processes.
- DSSs should support each phase of the decision making process.
- DSSs support on the half or full configured decision environments.
- DSSs support each management levels from bottom to top.
- DSSs have interactive and user-friendly interfaces.
- DSSs use data and model as a basis.

Decision support systems and relevant operation methods can be divided into four main subjects. These subjects are called inference mechanism, knowledge base, explanation module and active memory. Inference mechanism constitutes the basis of decision support systems. In this part, the results are generated under the consideration of the current information and/or the information that was entered to the system by the user. The generated results may be a decision or they may include guiding information. The second part is the knowledge base which holds the expert information used when the decision support system is making inference. The active memory part holds the information, which is supplied by the user and/or current inference processes. Also, explanation module, which may not be present on each decision support system, generates an accuracy validation and explanation in consideration of the results generated by the inference mechanism and knowledge base. Those subjects and their relations are shown in Figure 1.
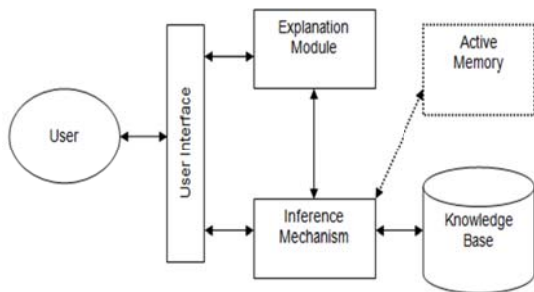

Fig. 1. The main structure of Decision Support System (DSS)

In rule-based systems, the knowledge base is formed by the rule group. The results are obtained for various circumstances on the problem relevant to the subject, using the generated rules. The rules forming the knowledge base are prepared by if-then structure. The content of an inference system, which is developed using rule-based methods, consists of the rules generated by if-then, the facts and an interpreter that interprets the facts using the rules in the system. There are two methods used to process the rules in the rule-based methods. These methods are forward chaining and backward chaining. In forward chaining method, the results are obtained using the preliminary facts with the help of the rules. In backward chaining method, it is started with a hypothesis (or target) and the rules, which will reach that hypothesis, are searched. The retrieved rules generate sub-rules and the process continues this way. In cases, where the result is estimated and this estimation should be verified, backward chaining method should be used instead of forward chaining method. In order to generate the rule set in rule-based methods of inference systems, people who are experienced on the problem should contribute to the design of the system. This process usually proceeds with the help of experienced people in the rule development phase by determining the faults and defects in the estimations and using the planned system as a reference. The designer usually develops simple interfaces for experts to contribute in the development phase. In the beginning of the process, the experts start testing the systems as if they will use the system for operational purposes. The questions asked to the experts in the scope of the limited information of the systems are answered by the same experts. The aim is to test the system in order to improve it. The expert who answered the questions, evaluates the system by looking at the results generated by the system, and then tries to correct the defects and faults by using the rule development tool. The rule set in the inference systems, which use rule-based methods, can be generated by the expert on the problem. Data-driven systems examine large data pools in organizations. These systems usually work with the systems that collect data like a data warehouse and what not. Data-driven systems take place in decision making process with On Line Analytical Processing (OLAP) and data mining methods. These systems work on very large datasets. The relations in these datasets are analyzed electronically and make predictions for future

data relations. Data-driven systems use the bottom-up procedure to explain the characteristics of the data system[8,9].

3. Bayesian networks (BNs)

Probabilistic models have become increasingly popular in the last decade because of their ability to capture non-deterministic relationships between variables describing many real world domains. Among these models, graphical models have received significant attention due to their ability to compactly encode conditional independence assumptions over random variables and because of the development of effective algorithms for inference and learning based on these representations[10]. BNs, also known as causal networks or probabilistic belief networks, are widely used for knowledge representation and probabilistic reasoning in various domains under uncertainty[11,12,13].

The advantages of BNs to traditional approaches are given below:

- They give good results in missing and incomplete data.
- They are more efficient in modeling complex systems than other methods.
- The comprehensibility of BNs are easier than models such as artificial neural networks, etc.
- Each value that the variables in BNs take can have a value more than True/False.

Because of all these advantages BNs have widely and successfully been used in the areas such as Medical decision support systems, diagnostic and classification systems, data mining, information retrieval, expert systems, fraud detection, robotics and so forth.

A Bayesian Network $B=(P_r,G)$ is a model of a joint, or multivariate, probability distribution over a set of random variables; it consists of a graphical structure G and the associated distribution P_r. The graphical structure takes the form of a Directed Acyclic Graph (DAG), $G=(V(G),A(G))$ with nodes $V(G) = \{V_1, V_2,.......V_N\}$, $N \geq 1$ and arcs $A(G) \subseteq V(G) \ X \ V(G)$. Each node V_i in G represents a random variable that takes one of a finite set of values. The arcs in the digraph model the probabilistic influence between the variables. Informally, an arc $V_i \rightarrow V_j$ between two nodes V_i and V_j indicates that there is an influence between the associated variables V_i and V_j; absence of an arc between V_i and V_j means that the corresponding

variables do not influence each other directly. More formally, a variable V_i is taken to be dependent of its parents and children in the digraph, but is conditionally independent of any of its non-descendants given its parents; this property is commonly known as the Markov condition.

Associated with the graphical structure of a Bayesian Network is a joint probability distribution P_r that is represented in a factorized form. For each variable V_i in the digraph is specified a set of conditional probability distributions $P_r (V_i | \pi(V_i))$; each of these distributions describes the joint effect of a specific combination of values for the parents $\pi(V_i)$ of V_i, on the probability distribution over the values of V_i. These sets of conditional probability distributions with each other define a unique joint probability distribution that factorizes over the digraph's topology through[14].

$$\Pr(V_1,V_2,.....,V_N) = \prod_{i=1}^{N} \Pr(V_i \mid V_1,V_2,.....,V_{i-1})$$

$$\Pr(V_1,V_2,.....,V_N) = \prod_{i=1}^{N} \Pr(V_i \mid \pi(V_i))$$

(1)

If V_i has no parents[15], then the set $\pi(V_i)$ is empty, and therefore $P_r (V_i | \pi(V_i))$; is just $P_r (V_i)$. An example of conditional independence in Bayesian Network is shown in Figure 2 below:
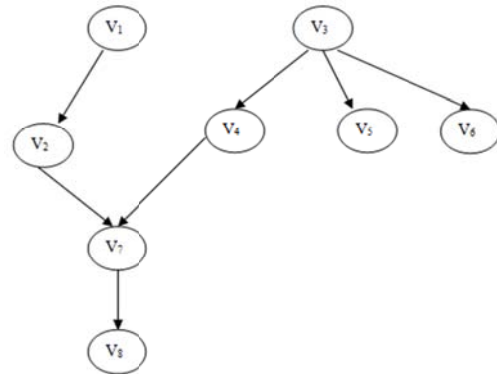

Fig 2. Conditional independence in BNs

The conditional probability of Bayesian network which is shown in Figure 2 is defined above. There are two inference methods that can be used in BNs. These are defined as exact inference and approximate inference methods. The number of variables in the methods based on BNs can be abundant. Although some

exact inference algorithms have been developed for Bayesian Network, belief update is NP-Hard[16] in BNs and these algorithms are not easy to apply very large and complex methods. Thus, it is important to focus on approximate inference algorithms[17].

Exact inference algorithms have many classes. Examples of these algorithms are, loop cutset conditioning, clique-tree propagation, node reduction, symbolic probabilistic inference, dynamic conditioning, recursive conditioning, Shenoy-Shafer, lazy propagation, bucket elimination, variable elimination and so on. Examples of approximate inference algorithms are stochastic simulation, Markov Chain Monte Carlo, probabilistic logic sampling, heuristic importance sampling, Gibbs sampling, hypercube sampling, incremental symbolic probabilistic inference, deterministic approximation, loopy belief propagation etc[18].

4. ALARM network structure and datasets

In this study, to compare the performances (in terms of accuracy) of rule based and Bayesian inference models on medical data, the network structure, that is common in scientific studies and known as ALARM Network[19] in the literature is used. ALARM network is a network structure that is prepared by using real patient information for many variables and shows the probabilities derived from the real life circumstances. ALARM network calculates the probabilities for different diagnosis based on the current evidences and recently it has been used by many researchers. In total, there are 37 nodes in ALARM network and the relationships and conditional probabilities between these have been defined. The medical information has been coded in a graphical structure with 46 arches, 16 findings and 13 intermediate variables that relates the examination results to the diagnosis problems that represent 8 diagnosis problems, Two algorithms have been applied to this Bayes network, one of which is a message-passing algorithm, developed by Pearl[12] to update the probabilities in various linked networks using conditioning methods and the other one is the exact inference algorithm, developed by Lauritzen-Spiegelhalter[20] for local probability calculations in the graphical structure. There are three variables named diagnosis, measurements and intermediate variables in the ALARM network:

(i) Diagnosis and the qualitative information are on the top of the network. Those variables do not belong to any predecessors and they are deemed mutually independent from the predecessors. Each node is linked to the particular and detailed value sets that represent the severity and presence of a certain disease.

(ii) Measurements represent any current quantitative information. All continuous variables are represented categorically with a discrete interval set that divides the value set.

(iii) Intermediate variables show the element that cannot be measured directly. The probabilities in the Bayes network can represent both objective and subjective information. ALARM network includes statistical data, logical conditional probabilities, that are calculated from the equations relevant to the variables and a certain number of subjective valuations and it is usually used to form the network structure over synthetic data.

In cases for all given different predecessor nodes, it is required to obtain a conditional probability for a node. The structure of ALARM network and defined variable are shown in Figure 3.
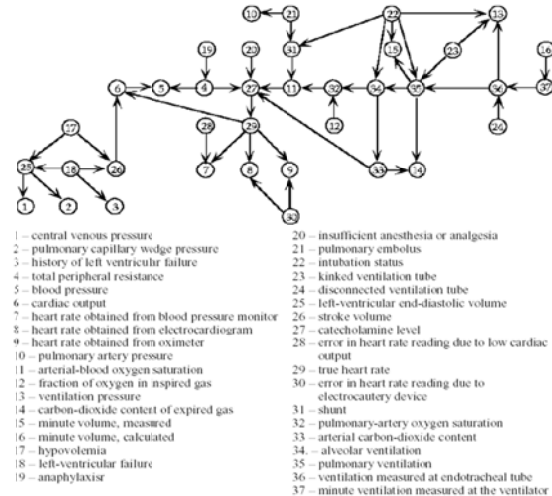

1 – central venous pressure
2 – pulmonary capillary wedge pressure
3 – history of left ventricular failure
4 – total peripheral resistance
5 – blood pressure
6 – cardiac output
7 – heart rate obtained from blood pressure monitor
8 – heart rate obtained from electrocardiogram
9 – heart rate obtained from oximeter
10 – pulmonary artery pressure
11 – arterial-blood oxygen saturation
12 – fraction of oxygen in inspired gas
13 – ventilation pressure
14 – carbon-dioxide content of expired gas
15 – minute volume, measured
16 – minute volume, calculated
17 – hypovolemia
18 – left-ventricular failure
19 – anaphylaxis

20 – insufficient anesthesia or analgesia
21 – pulmonary embolus
22 – intubation status
23 – kinked ventilation tube
24 – disconnected ventilation tube
25 – left-ventricular end-diastolic volume
26 – stroke volume
27 – catecholamine level
28 – error in heart rate reading due to low cardiac output
29 – true heart rate
30 – error in heart rate reading due to electrocautery device
31 – shunt
32 – pulmonary-artery oxygen saturation
33 – arterial carbon-dioxide content
34 – alveolar ventilation
35 – pulmonary ventilation
36 – ventilation measured at endotracheal tube
37 – minute ventilation measured at the ventilator

Fig.3. ALARM network structure and the variables defined in the network[21].

In order to use in the study, 100, 1000 and 2000 records have been produced based on the possibilities on the ALARM network. For these operations, based on the ALARM network structure, NETICA 3.18 software[22] has been used. Conditional probability

diagram for the ALARM network structure and a variable defined in the structure are shown in Figure 4. Each record on those generated data shows probable values for each 37 variables that were defined on this network. On each record, the values for intermediate variable as well as 12 input and 11 output variables. The tests, which were carried out, send the input variable values on each record to the relevant module and keep the resulting list as a separate file. The accuracy of the results is decided by comparing the variable values on the relevant record on the test data. For each record, 11 probable results have been obtained.
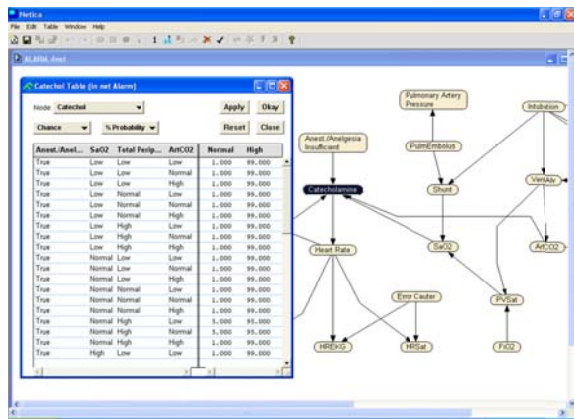

Fig.4. Conditional probability diagram for Alarm.net catechol variable

The results that were obtained by using JavaBayes[23] open source software is applied to each of the generated synthetically data sets separately. 11 output variables for one record belonging to 100-data-set are shown in Table 1. JavaBayes uses a generalized version of "variable elimination" method as an inference algorithm[24]. It has generated, 1100 output variables in the 100-data-set, 11000 output variables in the 1000-data-set, and 22000 output variables in the 2000-data-set. Table 1 below, for each data set, only 11 output variables for one record are presented. In this table, first column shows the variable name (disease name) and the second column shows the accuracy and they are calculated by the software using Bayes theorem, third column shows the real situations in the ALARM network, fourth column shows the results, generated by the DRCAD software and fifth column shows the comparison between the real situation and the results generated by the software. In the fifth column, if the real situation and the results

generated by the software are the same, POSITIVE result, if the real situation and the results generated by the software are not the same, NEGATIVE result will be generated. POSITIVE values indicate correct diagnosis, whereas NEGATIVE values indicate incorrect diagnosis.

Table 1. 11 Output variables for one record (100-data-set)

Variable name (Disease)	Accuracy degree	Real situations	Results produced by the DRCAD software	The comparison of the real situation and the results produced
History	0.9900	False	False	POSITIVE
Pres	0.9412	Normal	Zero	NEGATIVE
MinVol	0.9136	Normal	Zero	NEGATIVE
ExpCO2	0.9136	Normal	Zero	NEGATIVE
PAP	0.9000	Normal	Normal	POSITIVE
HRBP	0.8229	High	High	POSITIVE
HREKG	0.8229	High	High	POSITIVE
HRSat	0.8229	High	High	POSITIVE
CVP	0.7075	Normal	Normal	POSITIVE
PCWP	0.6970	Normal	Normal	POSITIVE
BP	0.4052	Low	Low	POSITIVE

For example, in Table 1, the accuracy of the MinVol variable has been calculated as 0.9136 by the software. Because this value is not the same with the real situation, correct diagnosis has not been obtained. Similarly, for HREKG, the accuracy has been calculated as 0.8228 by the software. Because this value is the same with the real situation, correct diagnosis has been obtained. Similar interpretations are also valid for other data sets.

5. Rule based method testing scenario

This section primarily starts by transferring the synthetic data in hand in accordance with the methods utilized properly, setting rules by making use of the data and the network topology, transferring the data and rules in the form of RuleML[25] and explaining the test practices on the data. Two approaches have been utilized here in order to determine the rules to be used by the related method on the ALARM network. First of them is to determine the relationships among the defined variables by using the topological structure of the network and constitute the rules and the second one is to derive some meaningful rules from the synthetic data in hand in the network by means of the practices in data mining methods. In the first method, except for the 12 defined triggering factors (input variables) the rules

have been composed for all variables by utilizing the network structure. In the phase of generating rules, the ALARM network, which is a Bayesian tool, the structure of the network has been screened by using the Netica 3.18 Software and thus related rules have been generated for each variable from the related image presentations. One of the rules to be generated by utilizing the table shown in the Figure 4 is like this;
IF (InsuffAnesth=True and SaO2=Low and TPR=Low and ArtCO2= Low) THEN Catechol= High

The display of this rule through RuleML is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<RuleML
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.ruleml.org/0.9/xsd"
xsi:schemaLocation="http://www.ruleml.org/0.9/xsd
http://www.ruleml.org/0.9/xsd/datalog.xsd">
<Assert>
<formula>
        <Implies>
                <head>
                        <Atom>
                                <op>

        <Rel>Catechol is High</Rel>
                                </op>
                          <Degree>
                           <Data>0.70</Data>
                          </Degree>
                        </Atom>
                </head>
                <body>
                        <And>
                                <formula>
                                        <Atom>

        <op>

        <Rel>InsuffAnesth is True</Rel>

        </op>
                                        </Atom>
                                </formula>
                                <formula>
                                        <Atom>

        <op>

        <Rel>SaO2 is Low</Rel>
```

```
        </op>
                                        </Atom>
                                </formula>
                                <formula>
                                        <Atom>

        <op>

        <Rel>TPR is Low</Rel>

        </op>
                                        </Atom>
                                </formula>
                                <formula>
                                        <Atom>

        <op>

        <Rel>ArtCo2 is Low </Rel>

        </op>
                                        <Ind>True</Ind>
                                        </Atom>
                                </formula>
                        </And>
                </body>
        </Implies
</formula>
</Assert>
</RuleML>
```

The rules generated by utilizing the topological structure of the network can be considered and classified under two headings: "The rules which generate the intermediate results and additional diagnosis" and "The rules which generate the target diagnosis".

The rules in the first group are the auxiliary ones which are utilized in order to reach the target results. Some supplementary and additional diagnosis can also be generated by making use of these auxiliary rules. The conditions of the rules should be affirmed in order to reach the target results within the pool of the rules to be used. In those cases when the data provided by the user to be utilized to affirm conditions of the rules are insufficient, those rules which are expected to affirm the conditions are examined thoroughly. These rules can also be defined as those rules which are in charge of generating intermediate results. Those rules in the second group are the sort of rules which are utilized in order to reach the target conclusions. If necessary, it is

possible to try to reach some results by benefitting from the current realities and/or intermediate conclusions generated by those auxiliary rules. The inference process steps are followed in accordance with the selected inference algorithm in order to reach the target results. Some intermediate conclusions may be necessary depending on the target to achieve or it may be possible to obtain the diagnosis by benefitting from the current facts as in the case of the rules which are in charge of producing some intermediate conclusions. Within the framework of the system created depending on the problem and the nature of the solution, the "disease" is examined and questioned. Thereby, some segregations or distinctions are made such as target diagnosis and intermediate conclusions. The target may vary and constitute different forms in inference machines depending on examining and questioning phases of the disease taken as a target.

188 rules have been added into the pool of rules which are convenient to examine by implementing machine learning techniques and also capable of generating intermediate conclusions and supplementary or additional diagnosis discovery or generation. These rules are created by utilizing ALARM.net network structure and by setting some possible rules for each variable. The text based if-then are transferred so as to become and serve as a valid RuleML document and added into the pool of rules.

See5 application[26] has been used in order to determine the meaningful rules on the synthetic data set we have in hand by utilizing the data mining methods. The relationships within the data sets and the definition of the patterns and related rules are derived by evaluating the See5 data of the cases. Thus, the rules intended to the target diagnosis are generated through this application by making use of decision trees. There are 85 rules intended for target diagnosis. The screen output belonging to See5 programme and an example rule tree are presented below in Figures 5 and 6, respectively.
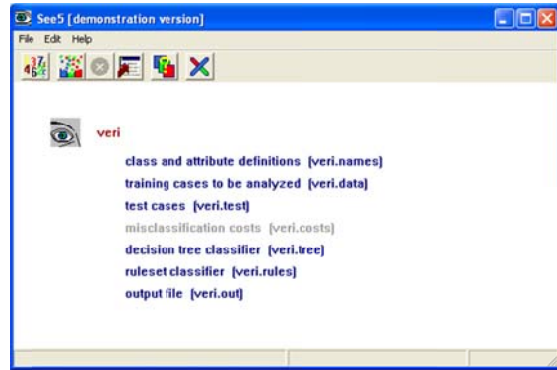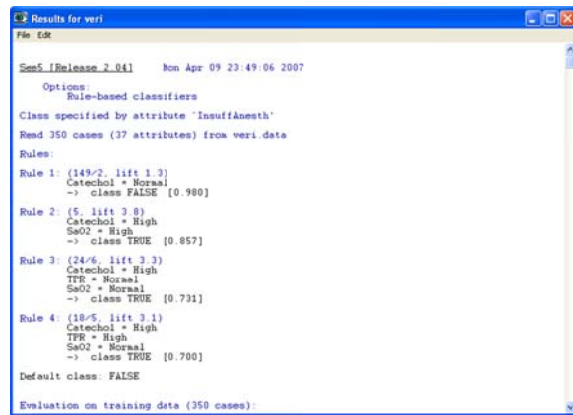

Fig.5. See5 application main screen

Fig. 6. See5 sample decision tree application

Herein, the system as seen in Figure 7 is composed of three main architectural and functional parts so that it will be possible to apply the "Rule Based Method".
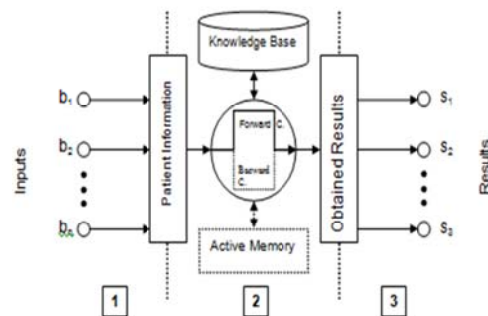

Fig. 7. Functional Architecture of the Rule Based Method

In the functional architecture, the first part is the interface which enables us to obtain the information about the patient in question. This section, which is transformed into practice, is not the only part that can be utilized to support the decision support system, but it can also support more than one decision support system to be used.

Obtaining information about a patient means introducing the facts to the system by means of which the machine will make inferences about the patient. Inferences are made by utilizing these facts and some new facts are obtained. There may be some information regarding the case of the disease, the differences accrue as a result of the application of the treatment or information on the results of the tests performed for the patient among the facts obtained. The information in the system must be in the form which will enable us to process all the information regarding the patient even if this information comes from other systems which use different software. It has been decided that the bridge to be built between other decision support systems should be XML (EXtensible Markup Language) which has been developed in order to make it possible to operate such various systems together in cooperation with each other compatibly. The usage of XML is a standard which is benefitted from when it is intended to exchange information among those platforms which utilize different systems for their operations. The function of the bridge is to send the questioning information taken from the users and transfer them to lower/bottom layers, that is, it transfers the information into forms which are possible for the bottom decision systems to process and send them into other layers by modifying them depending on their kinds and origins according to the purpose of the usage.

RuleML rule markup language is used in order to convey the meanings of the rules and facts utilized within the developed medical decision support system. Thus, it has been possible to exchange XML based information between the bottom systems and the bridge system stated above by utilizing this language.

The information is provided to the developed system in the form of XML based RuleML documents. A sample for the patient information utilized during the test phase of the developed system has been provided below:

- FiO2 is Normal
- Intubation is Normal

- Hypovolemia is False
- Anaphylaxis is False.

The FiO2 value and the Hypovolemia state of the patient given as a sample above are stated as in follows in the RuleML markup language;

The display of the patient's "FiO2 value is normal" for RuleML:

```
<?xml version="1.0"?>
<RuleML
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.ruleml.org/0.9/xsd"
xsi:schemaLocation="http://www.ruleml.org/0.9/xsd
http://www.ruleml.org/0.9/xsd/datalog.xsd">
  <Assert>
   <formula>
    <Atom>
     <degree>
      <Data>0.99</Data>
     </degree>
     <op>
      <Rel>FiO2 is Normal</Rel>
     </op>
    </Atom>
   </formula>
  </Assert>
</RuleML>
```

The display of "Hypovolemia is false" for RuleML:

```
<?xml version="1.0"?>
<RuleML
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.ruleml.org/0.9/xsd"
xsi:schemaLocation="http://www.ruleml.org/0.9/xsd
http://www.ruleml.org/0.9/xsd/datalog.xsd">
  <Assert>
   <formula>
    <Atom>
     <degree>
      <Data>0.8</Data>
     </degree>
     <op>
      <Rel>Hypovolemia is False</Rel>
     </op>
    </Atom>
   </formula>
  </Assert>
</RuleML>
```

G. Kose et al.

The facts, rules and results utilized during the study has been prepared in accordance with the Rule ML 0.9 valid RuleML document.
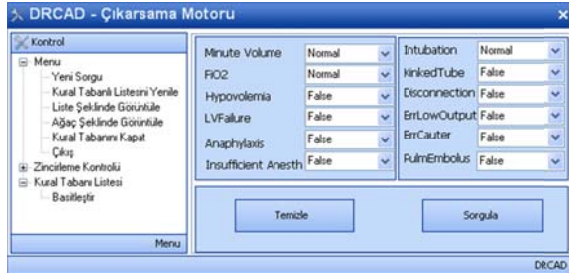

Fig.8. DRCAD rule based inference interface

The DRCAD[27] rule-based inference interface is seen in Figure 8. The information are produced which will come from the intermediate layer among the solutions suggested through this interface. Facts about the patients are generated by means of this programme which can respond to the individual data and those tests which are influential on the structure of the ALARM Network which is selected as the problem to be examined in the system. In accordance with the responses provided as programme output depending on those questions for valid RuleML documents and the results of probable diagnosis are generated accordingly. Those generated documents are copied into the index directories which are recognized by the inference machine. During the testing procedure of the system, 12 RuleML documents have been created for each patient and transferred to the system. Therefore, the system has become capable of operating by itself and available to be benefitted from as a decision support system by means of the system which has been transformed into practice through this programme.

The production of the information concerning the patients, in short, actualizing the task of the intermediate layer has triggered the process of introducing the information regarding to the patient to the system. Before performing the process of inference in the second section where inference operations are carried out, these RuleML documents should be transferred into programme objects which can be utilized within the inference engine. In the functional architecture, the second section where the inference operation is actualized can both process the facts which will come from the intermediate layer and also undertake the tasks

of the intermediate layer which has already been explained in the previous section. Thus, it will also process the facts which are available as a result of functioning as the bottom layer. The task of the inference machine is to generate results according to questioning performed by utilizing those facts to be sent to the machine and according to the rules. The realized inference machine has been inferring conclusions by utilizing those facts and rules chosen in order to be benefitted from, in accordance with the questioning. The inference machine recognizes the rules by using a similar system which is used while adding up the facts to the system. RuleML documents are also used here as in the previous steps. The rules utilized by the inference machine are prepared for operation, they are rules just like the RuleML standards. In order to carry out the inference operation, it is necessary to provide the system with the facts and they must be given on the basis of rules as well. After defining the field where the rules exist to the system, the inference machine starts making inference operations by utilizing those facts and rules.

As a result of simplifying operation of those single level rules, in other words, those rules which contain only the unit control rules are harvested. What is meant by unit control rules is that the structures of the rules must have only one condition or those which have more than one condition should have single level structures. Single level rules are the ones composed of "and" or "or" and each condition is an atomic expression. The inference operation can be achieved successfully by implementing those rules which are generated as a result of simplifying. When a rule which is combined with "or" is separated into its components, it can be processed as a single conditioned rule but the rule concerning the problem examined will not be segregated. Since there are rules composed of "or", this method has not been utilized. If the solution practiced in this subject is examined;

(i) If the rule is composed of "or",
  (a) If the number of variables is not 0,
  (b) Can the current condition be verified?
(ii) If it can be verified, the values of variables are found in the condition should be assigned to all of variables within the whole rule having the same names,

(c) And if there is an unprocessed condition, then choose the next condition and go back to the first step.

(d) If none of the conditions are verified and there aren't any variables in the conclusion sentence, then add the result obtained into the memory.

Thanks to this solution applied that there has been a solution differentiation or distinction between those rules composed of "and" and those rules composed of "or". This solution has been transferred into practice through advanced forward chaining and backward chaining method so that the method can comply with the structure of operation of the system. The other modifications applied to the inference machines are to generate results with the confidence degree which is presented in the structure as a solution to the problem examined. The confidence degree shows the possibility of the correctness of a result which is generated. According to this assumption, their degree of confidence will exist in those results obtained as a result of the inference operations carried out in the inference machine if all those rules and/or the facts are defined in it. Thanks to this, in accordance with the facts utilized within the system and the results obtained in line with the rules on which some comments can also be made saying that they should also be correct and valid. The confidence degrees are the values which may change between 0 and 1. The method concerning this confidence degree of the calculations made during the inference operations can be explained as follows:

(i) While making conclusions from the rules,

(a) If the rule is a sentence constructed by using the word "and" for each verified condition, the confidence of the rule is calculated by using this formula; min (rule confidence, the condition confidence at that moment).

(b) If the sentence of the rule constructed by using "or" for each validated condition, the confidence of the rule is calculated by using this formula: max (the confidence of the rule, the confidence of the condition at that moment).

(c) If all of the conditions of the rule are provided and if a confidence is acquired as a result of justifying these rules;

• If there isn't a reliable value in the conclusion section of the rule, the validity acquired from the concluding conditions is added together to the memory.

• If there is a confidence value in the conclusion section of the rule, the conclusion is added to the memory according to this formula: confidence = (confidence of the conditions * confidence of the result).

In the approach mentioned here, it is stated that the confidence of a rule depends on the confidence of the conditions. In cases where there isn't any confidence in the conditions of a rule, the conclusion is added to the memory together with the confidence degree of the rule. The confidence of operations in the system composed is created to be taken into account in cases where there is confidence.

Some special cases exist in confidence operations. These cases are as follows;

(ii) If a condition can be verified depending on one of the current facts,

(a) If a confidence degree isn't determined in a condition and the fact owns a confidence degree; the confidence degree of the condition is equalized to the value of the fact and the calculation of the confidence degree of the rule is performed by making use of this value.

(b) If there is a confidence degree in the condition and if there is also a confidence degree in the fact; the confidence degree of the condition is calculated by using this formula: confidence = (confidence of the condition* confidence of the fact) and the confidence degree of the rule is calculated by using this value.

(iii) If a conclusion generated by a rule exists in the memory,

(a) If there isn't a confidence degree of a generated conclusion and if there is the confidence degree of fact in the memory, in such cases no operation or calculation is performed

(b) If there a confidence degree in the generated conclusion and there isn't any confidence in the fact within the memory, the confidence degree of fact in the memory is equalized to the confidence degree of the generated conclusion.

(c) If there is confidence degree in both generated result and in the fact in the memory, then the confidence degree of the fact in the memory (the confidence degree of the condition to be

called *ccd* and the confidence degree is the memory is to be called *mcd*);

- If *mcd* >= 0 and *ccd* >= 0; Confidence = *ccd* + *mcd* - (*ccd* * *mcd*)
- If *mcd* < 0 and *ccd* < 0; Confidence = *ccd* + *mcd* + (*ccd* * *mcd*)
- If *mcd* * *ccd* < 0; Confidence = (*ccd* + *mcd*) / (1 – *min* (*ccd* , *mcd*)) are calculated as shown above and updated.

Some easy mathematical expressions can also be processed in the rule processing systems which are transferred into practice in the inference machine. It is possible to do and control these operations in this created structure such as smaller (<), small equals (<=), bigger (>), big equals (>=), equals (==) and not equal (!=)

Advanced forward and backward chaining techniques in the inference machine which is transferred into practice operate on the known general lines in accordance with the stated arrangements. Since advanced forward chain method does not stem from the questioning system, the result harvesting methods can be left for the next section; however, the backward questioning method stem from questioning, it may be necessary to provide it with some initial information before the section of "the generating the results". Although the forward chaining method is generated starting from the present rules and facts new ones are produced in the inference operation, backward chaining method moves in the direction of the questioning.

After the completion of the inference operation for the forward chaining method in the system which was transformed into practice, the operations are performed depending on the facts which are determined by the user or those facts which are obtained through the defined questioning in the pool of rules. Consequently, the results which are compatible with the questioning are transferred into the conclusion set. As it is done in the case of the facts and rules they are added to the system, the questioning operation is also performed through those valid RuleML documents. For instance, the question "What is the disease?" is expressed as the document of a questioning Rule ML as follows;

```xml
<?xml version="1.0" encoding="utf-8"?>
<RuleML
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.ruleml.org/0.9/xsd"
xsi:schemaLocation="http://www.ruleml.org/0.9/xsd
http://www.ruleml.org/0.9/xsd/datalog.xsd">
        <Query>
                <formula>
                        <Atom>
                                <op>
                        <Rel>diagnosis is</Rel>
                                </op>
                        <Var>diagnosis</Var>
                        </Atom>
                </formula>
        </Query>

</RuleML>
```

When the created RuleML document regarding the questioning is examined, it can be seen that the relationship to be searched for is the disease. In this advanced forward questioning, those facts, which accommodate disease in these facts obtained as a result of questioning, will be selected and the variable zone discovered during the questioning phase if the structures of the units are compatible, they will be filled in and they will be added to the conclusion set. The questioning document shown here also will be evaluated as a kind of object within the system as an element.
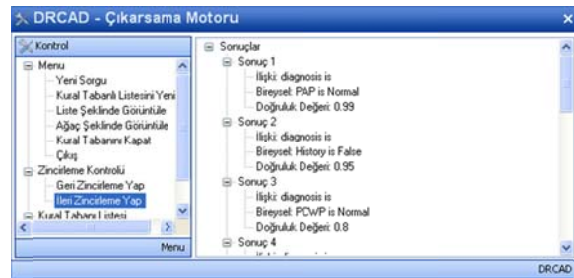

Fig.9. Sample conclusions generated through the inference engine

After the completion of the chaining operation and obtaining the conclusions, if there are any, they are arranged according to the degree of their confidence rate starting from the biggest to the smallest ones as in

Figure 9. Each of the sets within the obtained conclusion sets are recorded as valid RuleML documents. After transferring this phase into practice, the generated solution to the defined problem has been actualized. Thus, conclusions are generated with the facts taken as in the forms of Standard RuleML documents of facts and regulations. Again, the result is produced by means of the Standard RuleML documents.

6. Bayesian network based method testing scenario

During the development of DRCAD Medical Diagnosis Software, open source code JavaBayes has been used to implement Bayesian Network based inference method. JavaBayes software consists of two main modules. First of these modules is the Bayes Editor, which has been developed for creating Bayesian network, forming variables and links, making the essential corrections on the network, and observing the values of the network variables. The second module is the inference algorithm that has been developed for making the essential inference and implementing the query on the network.

The interface of the example Bayesian network which has been developed and organized by making the essential corrections is shown below in Figure 10.
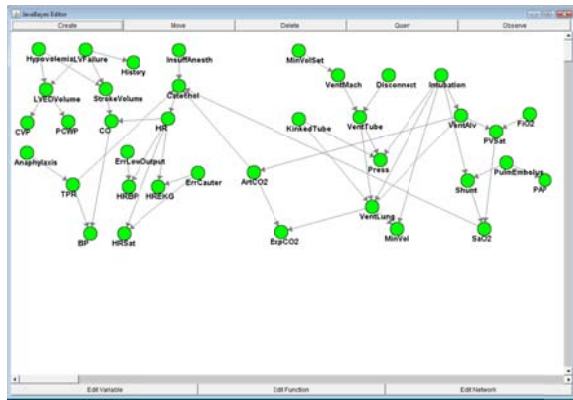

Fig.10. JavaBayes network creating and editor

JavaBayes software retrieves the possible values of variables that are determined according to the given inputs about the network by the help of the related algorithm module. After this phase, it is easy to determine a diagnosis list by ordering the related values. JavaBayes software uses the generalized interpretation of "Variable Elimination" method as the inference algorithm. In DRCAD system, a driver application is

developed that gives the list of possible diagnosis according to the related inputs by using the infrastructure of JavaBayes software. The interface of the related application and inference screen are shown in Figure 11 and Figure 12, respectively.
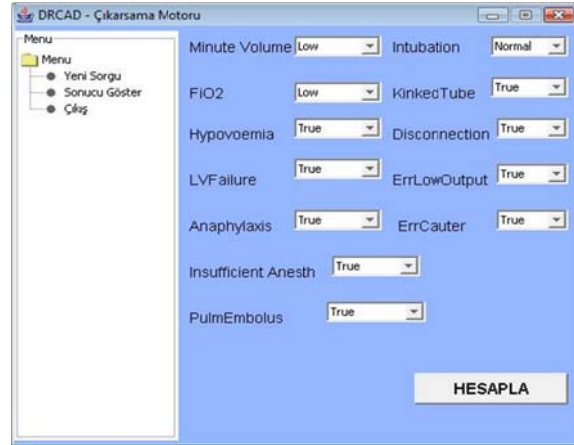

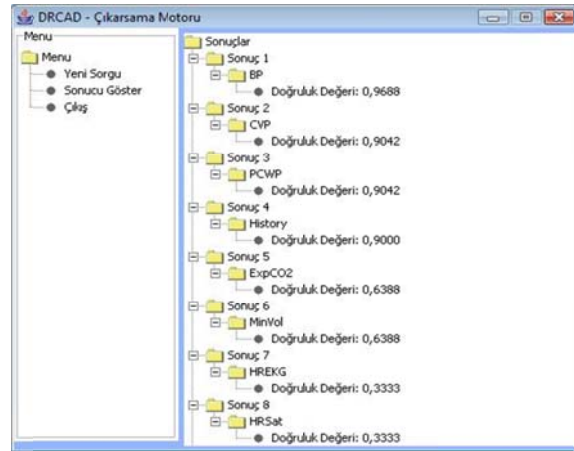Fig.11. DRCAD Bayesian based inference engine interface

Fig.12. DRCAD Bayesian-Based inference engine example of inference results

7. Test results of drcad medical diagnosis software

Within the DRCAD Medical Diagnosis Software, the results of the "Bayesian Network Based Inference" and "Rule Based Inference" methods are compared together by generating synthetic test data consisting of 100, 1000 and 2000 records that reflect the possibilities on the ALARM network structure[28]. Additionally, the results obtained from these two methods are linearly combined

and compared to the individual ones. Each record composed on these data shows the values belonging to 37 variables registered on the network. There are 12 input and 11 output variables on each record and the records of all the intermediate values are also saved and kept for further use. Those tests which are actualized send the input variables for each record to the inference engine and it saves the files of the lists created following each of the inference operation. The correctness of the inference results are decided by comparing the records of the related variables of the recorded test data. 11 probable results are obtained for each record. The tests are composed of the application of the "Rule Based Method" and "Bayesian Network Based Method" on the synthetic data set and additionally the linear combination of results of these two methods. The test results obtained according to this schedule are shown in Table 2.

Table 2.The results for the DRCAD inference methods

Synthetic Dataset with 100 Records			
Method	Number of Correct Diagnosis	Number of Total Diagnosis	Accuracy
Rule-Based	856	1100	77.82%
Bayesian	576	1100	52.36%
Linearly Combined	896	1100	81.46%

Synthetic Dataset with 1000 Records			
Method	Number of Correct Diagnosis	Number of Total Diagnosis	Accuracy
Rule-Based	8341	11000	75.83%
Bayesian	5550	11000	50.46%
Linearly Combined	8908	11000	80.98%

Synthetic Dataset with 2000 Records			
Method	Number of Correct Diagnosis	Number of Total Diagnosis	Accuracy
Rule- Based	16783	22000	76.29%
Bayesian	11118	22000	50.54%
Linearly Combined	17810	22000	80.96%

The first column in this table shows the method by means of which the inference is performed, the second column shows the number of correct decisions generated by the DRCAD Inference Engine, the third column shows the number cases examined and finally in the last column the percentages of the accuracy rate of the decisions taken are shown. When the conclusions are examined, it can be seen that the Rule Based Method is more successful in the rate of 25% than the "Bayesian Network Based" method in all dimensions of the data sets. Besides, when both of these methods are combined and utilized together the success rate rises to 80% much higher rates are acquired in comparison to the values obtained by applying these methods individually. In other words, the method in which the conclusions are combined in a linear manner are 5% more successful than the "Rule Based Method when applied individually and 30% more successful than the cases where the "Bayesian Network Based" method is utilized.

In summary, "Bayesian Network Based" method captures causal dependencies using the conditional probabilities based on the relationships between each node on the ALARM network. On the other hand, the "Rule Based" method allows for a direct construction of classification relations by capturing the knowledge retrieved from the data. In this study, the performance of the "Rule Based" method shows better results than those of the "Bayesian Network Based" method since the rules extracted by using data mining techniques describe the complicated relations between the variables better for the generated data sets. However, adding the rules extracted by the "Bayesian Network Based" method to the rule base increases the accuracy rate of the medical diagnosis since the structural information of the Bayesian network is reflected to the rule base. So, it is concluded that two methods' results complement each other and provide higher accuracy rates for medical diagnosis.

8. Conclusion

In cases of uncertainty and the lack of information, the most important part of the decision support systems which supports decision-making process is the inference mechanism. There are data mining methods such as Support Vector Machine, Multilayer Perceptron, Decision Trees, and so forth that are available in inference mechanism. Those methods can be used

separately in an inference mechanism or also as a hybrid system, which consists of a combination of those methods. In this study, for the generated synthetic data, ALARM network structure which is widely used in scientific studies has been used. This network structure is a structure that has been prepared using real patient information for many variables and shows the possibilities derived from real-life circumstances. When the results are examined, it can be seen that the Rule Based Method is more successful in the rate of 25% than the "Bayesian Network Based" method in all dimensions of the data sets. Besides, when both of these methods are combined and utilized together the success rate rises to 80%, i.e., much higher rates are acquired in comparison to the values obtained by applying these methods individually. In other words, the method in which the conclusions are combined in a linear manner are 5% more successful than the "Rule Based Method when applied individually and 30% more successful than the cases where the "Bayesian Network Based Method is utilized. Finally, DRCAD system is more innovative and interesting than the classical diagnosis support systems by collecting possible diagnosis of the patients from two sub modules. In future, different methods should be combined within this framework and their performances should be compared.

References

1. E.H. Shortliffe, S. G. Axline, B. G. Buchanan, T. C. Merigan, S. N. Cohen, An Artificial Intelligence Program to Advice Physicians Regarding Antimicrobial Therapy, *Computers and Biomedical Research* **6** (1973) 544-560.
2. E.H. Shortliffe, Clinical Decision based on Physician-Computer Interactions: A Symbolic Reasoning Approach, *Proc. of the Annual Meeting Society for Computer Medicine*, Las Vegas, 1977.
3. J.R. Clarke, D. P. Cebula, B. L. Webber, Artificial Intelligence: A Computerized Decision Aid for Trauma, *Journal of Trauma* **28** (1988) 1250-1254.
4. J.R.Quinlan, Decision Trees and Decision Making, *IEEE Transactions on Systems, Man and Cybernetics* **20** (1990) 339-346.
5. Milho, I. and Fred, A., "A User Friendly Development Tool for Medical Diagnosis Based on Bayesian Networks", *Proceedings of the Second International Conference on Enterprise Information Systems*, ICEIS, (Stafford, 2000) pp. 176-180.
6. A.W. Deja, P. Paszek, Diagnose Progressive Encephalopathy Applying the Rough Set Theory, *International Journal of Medical Informatics* **46** (1997) 119-127.
7. H. Yan, Y. Jiang, J. Zheng, C. Peng, Q. Li, A Multilayer Perceptron-Based Medical Decision Support System for Heart Disease Diagnosis, *Expert Systems with Applications* **30** (2006) 272-281.
8. S.C. Yucebas, HIPPOCRATES-I: Medical Diagnosis Support System Based on Bayesian Network, *M. Sc. Thesis*, Baskent University, Ankara, 2006. (In Turkish) (181360)
9. M.E. Corapcioglu, Core of Medical Decision Support System, *M. Sc. Thesis*, Baskent University, Ankara, 2006. (In Turkish) (181361)
10. R.S. Niculescu, T.M. Mitchell, P.B.Rao, A Theoretical Framework for Learning Bayesian Networks with Parameter Inequality Constraints, *Proc. of the 20 th Int. Joint Conf. on Artificial Intelligence*, IJCAI'07, Morgan Kaufmann, San Francisco, CA, 2007, pp. 155-160.
11. E. Castillo, J.M. Gutierrez, A.S. Hadi, *Expert Systems and Probabilistic Network Models*, Springer Verlag, New York (1997).
12. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, CA, USA, 1988.
13. L.M de Campos, J.M. Fernandez-Luna, J.A. Gamez, J.M. Puerta, Ant Colony Optimization for Learning Bayesian Networks, International *Journal of Approximate Reasoning* **31** (2002) 291-311.
14. P.J.F. Lucas, LC. Van der Gaag, A. Abu-Hanna, Bayesian Networks in Biomedicine and Health-Care Artificial Intelligence, *Artificial Intelligence in Medicine* **30** (2004) 201-214.
15. L.M de Campos, J.M. Fernandez-Luna, J.F.Huete, The BNR Model: Foundations and Performance of a Bayesian Network-Based Retrieval Model, *International Journal of Approximate Reasoning* **34** (2003) 265-285.
16. G.F Cooper, The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks, *Artificial Intelligence*, **42** (1990) 393-405.
17. Y. Lin, M.J Druzdzel, Stochastic Sampling and Search in Belief Updating Algorithms for Very Large Bayesian Networks", *In Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information*, 1999, pp. 277-82.
18. H. Guo, W. Hsu, A Survey of Algorithms for Real-Time Bayesian Network Inference", In: *In the Joint AAAI-02/KDD-02/UAI-02 Workshop on Real-Time Decision Support and Diagnosis Systems*, 2002, Edmonton, Alberta, Canada.
19. I.A. Beinlich, H. J. Suermondt, R. M. Chavez, G. F. Cooper, The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks, Proc. of the Second European Conference on Artificial Intelligence in Medical Care, Springer Verlag, Berlin, 1989, pp.247-256.
20. S.L. Lauritzen, D. J. Spiegelhalter, Local Computations with Probabilities on Graphical Structures and Their

Application to Expert Systems, *Journal of the Royal Statistical Society* **50** (1988) 157-224.

21. J. Cheng, D. Bell, W. Liu, Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory, *Technical report*, http://webdocs.cs.ualberta.ca/~jcheng/Doc/report98.pdf

22. Netica Software, from http://www.norsys.com/netica_vb_api.htm.

23. Java Bayes Software, from http://www.cs.cmu.edu/~javabayes/home/.

24. F.G. Cozman, Generalizing Variable Elimination in Bayesian Network, *Proc. of the 7 th Ibero-American Conference on Artificial Intelligence*, IBERAMIA/SBIA'00, Sao Paulo, 2000, pp. 27-32.

25. RuleML, from http://ruleml.org

26. See5, from http://www.rulequest.com/see5-info.html

27. http://netcad.com

28. http://web.itu.edu.tr/ustundaga/datasets.zip