

# MdtDSS: A Decision Support System with Intelligent Recommendation for Multi-Disciplinary Medical Treatment

Nengjun Zhu<sup>a</sup>, Jian Cao<sup>a,\*</sup>, Kunwei Shen<sup>b,\*</sup>, Xiaosong Chen<sup>b</sup>, Zhengfei Shen<sup>a</sup>, Siji Zhu<sup>b</sup>

<sup>a</sup>Shanghai Institute for Advanced Communication and Data Science,  
Department of Computer Science and Engineering Shanghai Jiao Tong University, Shanghai, China  
<sup>b</sup>Comprehensive Breast Health Center, Shanghai Ruijin Hospital, Shanghai, China

---

## Abstract

A newly emerging trend in the healthcare field is for multi-disciplinary medical teams to collaborate to devise personal medical treatment programs. Argumentative discourse and collaboration among experts can help decision makers make more reliable medical treatment decisions, but they also challenge the efficiency of the decision-making process. With the development of machine learning technologies, recommendations can be automatically made by learning from historical data. In addition, cases can be categorized so that different process models can be applied to different cases so that more time can be allocated to intractable cases. This paper describes MdtDSS, a real-world decision support system for multi-disciplinary treatment. The recommendation approaches together with flexible process control are presented. Moreover, how to present the recommendation results so that human knowledge and machine knowledge can be integrated is also discussed. The evaluation results from the experiments and an empirical study based on real data and systems demonstrate that the system is efficient.

**Keywords:** Decision support system, recommender system, breast cancer, MDT, medical guidelines, workflow engine

---

## 1. Introduction

An incorrect treatment program has the potential to have many negative consequences both for doctors and patients; doctors' reputations are adversely affected and they may be accused of medical negligence, whereas patients have to bear the cost of unnecessary treatment and also the pain and inconvenience of having their health put at risk. One of the most important skills of a highly proficient doctor is to make rational medical diagnoses and design a rational treatment program. Unfortunately, due to the complexity of the medical domain and the limited knowledge of each individual doctor (even highly proficient doctors), there is always a high risk associated with the prescription of a treatment program.

In order to arrive at a more reliable decision on a treatment program, many medical institutions build a multi-disciplinary team for a particular disease, e.g., breast cancer, where doctors from various medical disciplines, such as pathology, oncology, and nutrition, collaborate. When a decision is made by a multi-disciplinary team, this is called multi-disciplinary treatment (MDT) [1], which is a

new trend in the field of clinical treatment, especially in interdisciplinary fields with complex knowledge dependency, such as oncotherapy. In order to support MDT meetings, the decision support system (DSS) [2, 3] as a useful and efficient tool, has been studied.

In order to lower the risk of designing improper treatment programs, many authoritative medical guidelines, such as the National Comprehensive Cancer Network (NCCN) [4], have been published and used by various medical institutions. In a DSS, the guidelines can be implemented as a rule-based system, which guides doctors to make correct decisions. However, the treatment programs recommended by these guidelines are commonly at a coarse level, such as *Need chemotherapy* instead of *Need which type of chemotherapy*. At the same time, the data-driven approach for making treatment program decisions is becoming very attractive since an increasing amount of data has been collected and recorded [5, 6, 6]. In such an approach, the inherent latent factors leading the rational treatment program decision can be mined from historical cases or the academic literature, and are further applied in the process of designing a treatment program for a patient. This data empowers the machine decision maker to be not only an assistant to help doctors determine the final treatment program, but to be similar to an independent doctor participating in the MDT process with its own empirical knowledge base. However, data-driven approaches suffer from certain inherent problems. For example, concept-drift [7] occurs when the statistical properties of data are changed

---

\*Corresponding author

Email addresses: zhu\_nj@sjtu.edu.cn (Nengjun Zhu), caojian@sjtu.edu.cn (Jian Cao), kwshen@medmail.com.cn (Kunwei Shen), chenxiaosong0156@hotmail.com (Xiaosong Chen), little\_wings@foxmail.com (Zhengfei Shen), zsj\_mu@yeah.net (Siji Zhu)

as time passes, e.g., the hospital introduces a new and revolutionary treatment program. Furthermore, the performance of data-driven approaches depends on the quality of data, which is very challenging to control in reality. The rule-based and data-driven approach should be integrated so that a more rational treatment program may be suggested by the system automatically.

However, in practice, there are many situations where a doctor is required to not only apply their medical knowledge but also to consider the patient’s individual health condition, financial situation and even their personal emotional state. Therefore, a doctor’s experience plays an important role in treatment decisions. Hence, a method to seamlessly combine the doctor’s knowledge with machine knowledge to assist the decision-making process is necessary.

In addition to the issue of how to generate a ”correct” treatment program according to the available knowledge, another issue is how to ensure the decision is efficient, since medical resources are usually limited, especially in countries like China. In order to streamline the process, workflow technology can be used to model and automate the group decision-making process like MDT [8, 9]. However, it is not an efficient way to allocate a doctor’s time if each medical case follows the same workflow. Generally, in MDT, some cases are more complex than others so they should be allocated a longer time to allow doctors to discuss them thoroughly while some cases are routine and can be treated in a regular way. Unfortunately, for MDT, dividing cases into different categories and applying different decision-making process models on them has not been addressed in current research.

The purpose of this paper is to develop a real-world decision support system called MdtDSS that addresses the rationality of the decision-making results and the efficiency of the decision-making process simultaneously. To achieve this goal, we propose a practical framework of DSSs for MDT and design a flexible workflow engine to control the decision-making process. Furthermore, the framework integrates an expert voting process and a recommender system. In particular, in order to provide intelligent recommendations, we explore multiple recommendation approaches including  $k$ -NN classifiers and medical guidelines-based methods (i.e., rule-based methods), which are combined to suggest rational treatment programs. Our contributions are as follows:

- We propose a framework of DSSs for MDT that combines human knowledge and machine knowledge to help decision making.
- We optimize the decision process for MDT, which integrates expert voting and automated recommendation. Moreover, cases are categorized so that they can be processed in different ways.
- We investigate multiple recommendation techniques for MDT and propose an integrated approach.

Based on the framework and the techniques, we implement a real-world DSS for MDT, namely MdtDSS and we evaluate it in practice.

The remainder of this paper is organized as follows: Section 2 introduces the work which is related to both DSSs and the associated recommendation algorithms. Section 3 details the proposed framework for the decision-making process of MDT followed by the implementation and experiments of MdtDSS discussed in Section 4. Then, in Section 5, two key issues of MdtDSS are discussed. The conclusions and suggestions for future work are presented in Section 6.

## 2. Related Work

DSSs have a great potential to contribute to clinical decisions, given their successful implementation in a wide range of applications [10, 11, 12]. Clinical decision support systems (CDSSs) are computer systems designed to impact clinician decision making about individual patients at the point in time when these decisions are made. Depending on their framework, CDSSs differ from each other in the timing at which they provide support (before, during, or after the clinical decision is made) and how active or passive the support is, that is, whether the CDSS actively provides alerts or passively responds to doctor input or patient-specific information [13, 14, 15, 16]. Although CDSSs have been developed over the last thirty years, many of them have been stand-alone systems or part of non-commercial computer-based patient record systems. Many works [17, 18, 19] have been conducted to explore the effects of computerized CDSSs on doctor performance and verify that DSSs can improve doctor performance.

A large body of research has been conducted in the area of medical recommendations based on various machine learning technologies. For example, iDoctor [5] provides personalized and professionalized medical recommendations based on hybrid matrix factorization. Wang et al. proposed Safe Medical Recommendation [6] which is a novel framework to learn embeddings of a patient-disease-medicine graph for medicine recommendation. Le Hoang Son et al. considered the integration of intuitionistic fuzzy sets (IFS) and recommendation systems (RS) into the proposed methodology and presented a novel intuitionistic fuzzy recommender system (IFRS). HIFCF is an effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis [20]. Sojen Pradhan et al. proposed an innovative approach to derive trust from social networks and to improve the matching in dental care recommendation systems [21].

IBM Watson Health<sup>1</sup> and DeepMind Health<sup>2</sup> are two real-world platforms which can be configured to support medical decision-making. IBM Watson was originally a research experiment to determine whether a computer could

<sup>1</sup><https://www.ibm.com/watson/health/>

<sup>2</sup><https://deepmind.com/applied/deepmind-health/>

be taught to read volumes of text such as Wikipedia, newspapers, and other text-based sources of information and produce reliable evidence-driven answers in response to natural language questions [22, 23]. Then, it was adopted to address diverse business challenges, such as IBM Watson for Oncology [24]. Although IBM Watson can provide doctors with evidence-based treatment options, it mainly focuses on integrating the data from hundreds of sources and understanding their various formats based on cognitive computing technologies or natural language processing technologies [25, 22]. Compared with IBM Watson, our MdtDSS possesses more high-quality data and strives to learn the decision model directly because the data are carefully prepared in a manual way. DeepMind Health [26] works with hospitals on mobile tools and AI (artificial intelligence) research to help patients go from test to treatment as quickly and accurately as possible. It supports various specific systems for healthcare, such as the automated analysis of retinal imaging [27] and automated segmentation of head and neck tumor [28]. They published the details of their algorithms but did not include any information on the system itself.

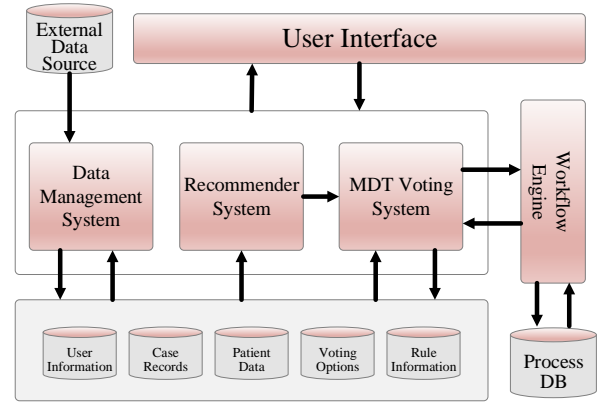
Currently, medical experts are becoming increasingly specialized with the appearance of a variety of new branches. Working together as an interdisciplinary team, many highly trained health professionals can deliver higher quality health care services [29]. Our work provide a DSS to support an interdisciplinary team to make joint decisions efficiently. Unfortunately, both IBM Watson Health and DeepMind Health do not support the group decision-making model.

### 3. A Framework for Decision Process of MDT

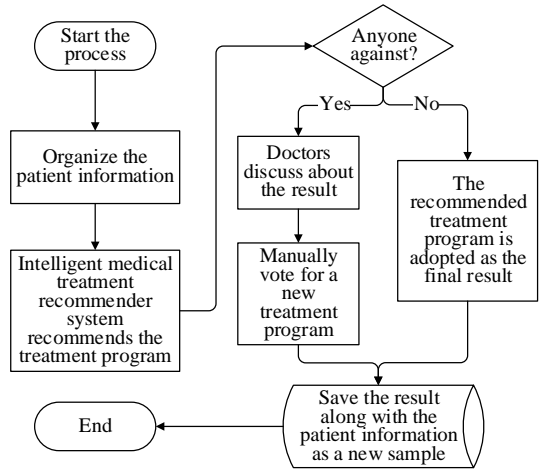
In this section, we first propose a practical framework of the general DSS for MDT. Then, the core parts of the framework, i.e., the MDT voting sub-system and the recommender sub-system, are introduced. Finally, a method to predict the degree of difficulty of the decision for each case is presented.

#### 3.1. The Structure of Framework

An overview of the system structure is shown in Figure 1(a). In addition to a web-based user interface, there are three modules, namely the data management sub-system, the recommender sub-system, and the MDT voting sub-system. The data management system inputs data from the external data source, e.g., an electronic medical record system (EMR) [30, 31] and converts them into a proper data format that is appropriate for later processing. In addition, rules and historical cases are maintained. The MDT voting system is in charge of the procedure of collaborative decision-making and it interacts with the workflow engine [8, 9]. For each case to be discussed in the MDT meeting, a process instance is created by the workflow engine. Then, the workflow engine tracks the status of each process instance as well as records doctors' behaviors



(a) The system structure of DSS for MDT.



(b) Procedure of a MDT meeting supported by intelligent treatment recommendation.

Figure 1: The structure of framework.

during the whole procedure for further analysis. The recommender system provides intelligent recommendations to doctors during the decision-making process. It can be integrated with MDT voting system in different ways. For example, one process model is shown in Figure 1(b). In this model, the system recommends treatment programs firstly. Doctors will examine the recommendation. Once the result is challenged by any doctor, a discussion is started, otherwise the recommended result is adopted. With the help of an intelligent treatment recommender system, we can speed up multi-disciplinary meetings so as to save the experts' time.

#### 3.2. MDT Voting System

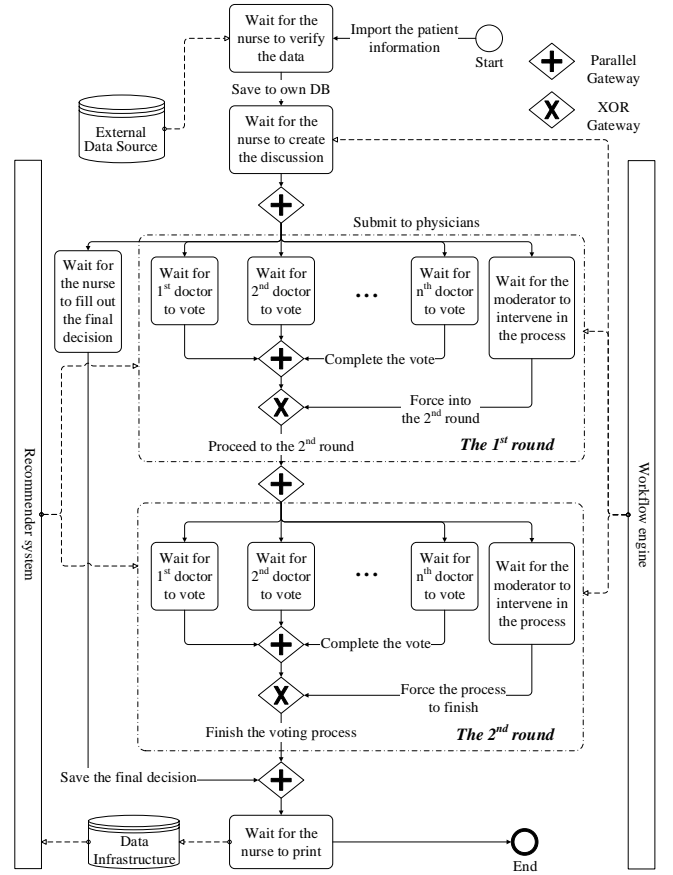
Voting is an effective method in a group decision process. The MDT voting system has the following requirements:

- (1) **Normative:** The change in the decision making process state is co-determined by the moderator and the current state of the process, and all participating users should follow a unified process model.
- (2) **Flexible:** The MDT voting procedure can be customized according to different requirements.
- (3) **Smooth:** Changes between states of the decision-making process are not disturbed by the individual actions of users except the moderator. When the process is finished, the final decision can be made.

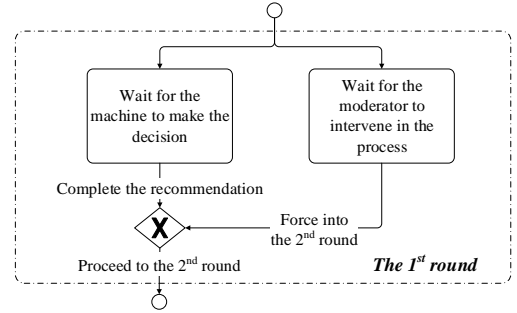
To achieve these goals, we provide a customized workflow engine to support the process management [32]. Figure 2(a) shows a typical current workflow diagram of MdtDSS. There are three main roles involved in the decision-making process, i.e., doctor, nurse, and moderator. The different user roles have different privileges and thus have different tasks to undertake. The nurse inputs/imports the patient information, initiates a vote and records the final decision results. The moderator controls the MDT voting process. The doctors cast their votes. The decision-making process flows automatically according to the designed steps. However, the moderator can take special action to manage the process, for example, by skipping a particular step or stopping an unfinished process if necessary. Typically, the whole process includes two rounds of votes by doctors before the final medical decision is made. Doctors are able to cast their first-round votes in an offline way. But they must sit together to cast their second-round votes by reviewing and debating the first-round results. When the case is finished, all the related information is recorded in the system. During the decision-making process, all users can refer to the system's recommendation results and they can also report any errors if the recommendation is not correct. This feedback is useful for recommendation model improvement. Note that, to save medical resources, routine cases can be processed in a simplified way, as shown in Figure 2(b). The machine completes the first-round vote alone while doctors only need to cast a second-round vote by reviewing and debating the results recommended by the machine.

An interesting question is how to provide recommendation results to doctors. Obviously, the recommended results have an explicit or implicit impact on the doctors' decisions. Therefore, we explore two ways to show the recommendations as follows:

- W1:** The recommendation results are shown on an individual page, as shown in Figure 4(c), which needs a spontaneous click. Doctors can refer to these in the decision-making process if they so choose.
- W2** The recommendation results work as a constraint set in the first-round vote, as shown in Figure 4(d). It only checks whether the doctor's choice is consistent with the recommendation results. If not, it sends an alert to the doctor.



(a) The workflow diagram of MdtDSS.



(b) The simplified workflow diagram of MdtDSS' first-round voting process.

Figure 2: The workflow diagrams.

We conducted experiments for these two settings in order to find the most appropriate way to show the recommendation results in MdtDSS. The details of the experiments and observations are discussed in the next section.

### 3.3. Recommender System

According to the characteristic of treatment program recommendation problem, we explored three approaches,

i.e.,  $k$ -NN classifiers [33], the rule-based method, and the synthesis recommendation method.

### 3.3.1. Problem Description

Assume there are  $N$  cases (samples) in the dataset  $D$  of MdtDSS with their treatment program. The treatment program can be regarded as a class label of a case. Let  $t^{th}$  sample in dataset  $D$  be denoted as follows:

$$D_t = \langle \vec{x}^{(t)}, y^{(t)} \rangle \quad (1)$$

where  $\vec{x} = (x_1, x_2, \dots, x_d)^T$  is a feature vector and  $y$  takes  $c$  values in class set  $C$  denoting the labeled class of  $D_t$ . Then, the task of the recommender system is to learn a decision model from the dataset  $D$ , and further provide a reliable classification for the new feature vector  $\vec{x}$ .

### 3.3.2. Feature Selection and Transformations

To better illustrate the recommender algorithms, we introduce feature transformations and their formats in breast cancer treatment as an example. The transformation step occurs when importing the patient records from the external data source to MdtDSS. The data infrastructure of the MdtDSS records both textual descriptive attributes and structured attributes. However, only predefined structured variables are transferred into a numeric feature vector. These variables are carefully selected and extracted by medical workers and thus are more valuable and credible than the information mined from text analysis (natural language processing). Actually, they cover most of the core information on the patients.

Categorical variables (attributes) are transformed into several features through one-hot-encoding, which consists of encoding a variable through binary numbers. In our case, we create one column for each of the categories of the variable. If the variable is a type of multiple choice attribute, the features of the variable may consist of multiple 1s, e.g., the value 3 can be represented by the vector [0,0,1,1] rather than being represented by a vector like [1,0,0,0], since the value 3 means the variable belongs to two categories at the same time. Numeric variables are recorded in the feature vector directly.

Different from the fixed length format data in traditional machine learning problems, the feature vector in our case is hard to align since one variable may have multiple different values. For example, a patient has multiple tumors where all tumors have homogeneous variables. We concatenate them to form new feature vectors and then align them to the longest vector. This technique allows the training model to fit such variables, but as dimensionality grows, the model loses generality. Fortunately, in line with medical practice, the doctors usually focus on the information on the main lesion when making a treatment program for patients with multiple lesions, because the main lesion contains representative information on the others. Thus, only the information on the main lesion is transferred, e.g., only the information of the largest tumor

Table 1: Therapies.

Chemotherapy (CT)			
#	Specific therapy	Subclass	Main class
1	No need	No need	No need
2	TC*4	TC	Need
3	TC*6	TC	Need
4	CMF	AC/EC/CMF	Need
5	EC(AC)	AC/EC/CMF	Need
...	...	...	...
19	ddAC-ddP	ddAC	Need
20	ddAC-wP	ddAC	Need
21	EC-wPCb	EC-WP	Need
22	EC-wP	EC-WP	Need
Radiotherapy (RT)			
#	Specific therapy	Subclass	Main-class
1	No need	No need	No need
2	APBI	Need	Need
3	WBI	Need	Need
...	...	...	...
11	Tumor bed boost	Need	Need
12	Other	Others	Need
Endocrinotherapy (ET)			
#	Specific therapy	Subclass	Main-class
1	No need	No need	No need
2	TAM*5y	SERM(TAM)	Need
3	TAM*10y	SERM(TAM)	Need
4	AI+OFS	AI+OFS	Need
...	...	...	...
14	AI*5y	AI	Need
15	AI*5y	AI	Need
Targeted therapy (TT)			
#	Specific therapy	Subclass	Main-class
1	No need	No need	No need
2	During CT	Need	Need
...	...	...	...
8	Sequential TT	Need	Need

will be considered to train the model. Note that, for a patient with bilateral primary breast cancer (BPBC), we consider each side as one individual case (sample), which means each side has its own feature vector with a label (the treatment program). Finally, each sample has 48 attributes.

Another key aspect of supervised machine learning is the definition of the label for each data entry. In our case, we face a multiple-classification problem where each data entry has to be labeled as a class belonging to one of the multiple classes, i.e., one of the therapies. There are four master categories of therapies: 1) chemotherapy (CT), 2) radiotherapy (RT), 3) targeted therapy (TT) and 4) endocrine therapy (ET), with each having multiple more specific therapies. As Table 2 shows, CT has the most options for voting (22), while TT has the least options (8). How-

ever, the situation with 8 options still challenges the performance of the machine decision maker due to the lack of historical samples for some options. In order to address the sample unbalance problem, these specific therapies are categorized into a more general subclass according to their medical properties. Then, the number of predicted classes with respect to CT, RT, TT and ET is 12, 4, 2 and 8 respectively, as shown in Table 2. Furthermore, the subclasses can be merged into two classes, i.e., *Need* or *No need*.

### 3.3.3. *k*-NN Classifiers

There are many data-driven approaches for decision making which are supported by appropriate machine learning techniques, such as neural networks (NNs) [34], support vector machines (SVMs) [35] and Bayesian classification [36]. These approaches have been proved very effective in practice. However, they have an obvious weakness, i.e., it is difficult to explain the recommendation results to doctors or patients. The *k*-NN classifier is a non-parametric method which explores the most similar samples of the target sample and makes a decision by referring to the results of these similar samples. It is exactly how human doctors make decisions and thus can easily be accepted by showing information on similar historical samples, as shown in Figure 4(c).

The similarity measure is one of the most significant factors of classifiers. To demonstrate our *k*-NN classifiers in detail, we first introduce the way to construct a similarity vector  $S^{(ab)}$  between sample  $D_a$  and  $D_b$ . The value of each element in  $S^{(ab)}$ , namely  $S_k^{(ab)}$ ,  $k = 1, 2, \dots, L$ , denotes the similarity of each single attribute (variable) between samples, which are measured based on Eq. (2) or Eq. (3) according to the type of attribute. For numerical attributes, similarity is calculated as follows [37]:

$$Sim(\vec{x}_k^{(a)}, \vec{x}_k^{(b)}) = \frac{1}{1 + \sqrt{(\vec{x}_k^{(a)} - \vec{x}_k^{(b)})^2}} \quad (2)$$

For categorical attributes, similarity can be measured by cosine similarity since we have encoded the values to a one-hot or multiple-hot vector:

$$Sim(\vec{x}_-^{(a)}, \vec{x}_-^{(b)}) = \begin{cases} \frac{\vec{x}_-^{(a)} \cdot \vec{x}_-^{(b)}}{\|\vec{x}_-^{(a)}\| \|\vec{x}_-^{(b)}\|}, & \vec{x}_-^{(a)} \neq \mathbf{0} \wedge \vec{x}_-^{(b)} \neq \mathbf{0} \\ s, & \text{others} \end{cases} \quad (3)$$

where  $\vec{x}_-$  is a truncation of the feature vector corresponding to an attribute which is a categorical variable and  $\vec{x}_- = \mathbf{0}$  means the value of this attribute is missing, and  $s$  is a constant value ranging from 0 to 1. To distinguish a case where two samples are completely different and a case where there is at least one sample with missing values, we set a non-zero value to  $s$ , e.g., 0.5 in MdtDSS. We remain neutral if we do not know the value of the target attribute.

Then, the final similarity measure between samples is a weight combination of the similarities on every single attribute as follows:

$$Sim(\vec{x}^{(a)}, \vec{x}^{(b)}) = \sum_{k=1}^L w_k \cdot S_k^{(ab)} \quad (4)$$

where  $L$  is the number of attributes and  $w$  is the weight.

We have to determine the weights before using this function. The simplest way is to assign values to the weights manually according to the experiences. For example, we can set a larger value to the weights of attributes which appear in the medical guidelines. However, it is very hard, if not possible, to assign weight values precisely in this way. Fortunately, we can explore the optimal weight values using an optimization algorithm like the genetic algorithm (GA) [38] on the training set. We define the fitness function (objective function) according to the classification error rate as follows:

$$\mathcal{L}_{GA1} = \frac{\#Misclassification}{\#samples} \quad (5)$$

where  $\#Misclassification$  represents the number of samples whose labeled class is not recommended (i.e., not hit) in the size- $N$  recommendation list, and  $\#samples$  represents the number of training samples. The goal of the genetic algorithm is to search the best weight value assignments to minimize the value of the fitness function through multiple iterations.

However, the penalty for all misclassified samples is treated equally in Eq.(5), which is inappropriate in the medical decision-making area because some mistakes are more serious than others. For example, the class *Need AC/EC/CMF chemotherapy* involves two meanings, i.e., the patient **needs** a chemotherapy and *AC/EC/CMF chemotherapy* would be the best one. In such a case, the recommendation *Need ddAC chemotherapy* should be better than *No need chemotherapy* since the former covers at least one aspect of the ground-truth. We argue that the former error (the subclass error) is a minor misclassification error and the latter error (the main class error) is serious. Since all therapies/subclasses can be further merged into a binary class *Need* or *No Need* in practice, the main class error is equal to the binary classification error. The error occurs when the recommendation list contains a subclass belonging to a wrong main class. Therefore, we change Eq.(5) as follows to differentiate the costs of serious and minor misclassification errors:

$$\mathcal{L}_{GA2} = \alpha \cdot \frac{\|B\|}{\#samples} + (1 - \alpha) \cdot \frac{\|A\|}{\#samples} \quad (6)$$

where  $B$  and  $A$  ( $\|A\| = \#Misclassification$  in Eq.5) are the set of samples of serious misclassification and minor misclassification errors, respectively, and  $\alpha$  is a parameter controlling the penalties between these two errors. In order to achieve the goal of a minimum number of serious

misclassification errors as well as a decent number of minor misclassification errors, we set  $\alpha$  to a value which is larger than 0.5, e.g.,  $\alpha = 0.9$  after the experiments in MdtDSS.

Moreover, the penalty for each misclassified sample can be automatically adjusted according to the degree of disagreement in the voting results from doctors. The greater the degree of disagreement, the more controversial the case is, and therefore the punishment should be reduced for misclassification. We define the degree of disagreement of the sample  $D_i$ ' voting result  $V_i$  based on the information entropy (IE) [39] as follows:

$$H(V_i) = - \sum_{c \in C} P_{ic} \cdot \log(P_{ic}) \quad (7)$$

where  $P_{ic}$  is the proportion of class  $c$  in voting results  $V_i$ . The value of penalty ranging from 0 to 1 can be converted by a  $H(V_i)$  using the following formula:

$$\beta_i = \frac{1}{1 + H(V_i)} \quad (8)$$

Then, the fitness function is as follows:

$$\mathcal{L}_{GA3} = \alpha \cdot \frac{\sum_{D_i \in B} \beta_i}{\#samples} + (1 - \alpha) \cdot \frac{\sum_{D_i \in A} \beta_i}{\#samples} \quad (9)$$

After the weights are automatically tuned through the genetic algorithm according to the fitness function discussed above, we can calculate the final similarities between the target and historical cases, and assign a class label to the target case according to its similarities to its neighbors. Instead of simply assigning the majority of class labels in its  $k$  nearest neighbors to the target case, we treat the similarity as a weight for each neighbor and assign a class label with the largest sum of weights of its neighbors.

$$\hat{y} = \arg \max_{c \in C} \sum_{D_i \in N(\vec{x}, k)} \mathbb{I}(y^{(i)} = c) \cdot \text{Sim}(\vec{x}, \vec{x}^{(i)}) \quad (10)$$

where  $N(\vec{x}, k)$  denotes the set of  $\vec{x}$ 's  $k$ -nearest neighbors, and  $\mathbb{I}(x)$  is an indicator function. Note that, concept-drift occurs in medical diagnosis fields for the reasons discussed. Therefore, the  $N(\vec{x}, k)$  is formed by considering the time factor. For example, when searching  $k$  nearest neighbors, we only consider the most recent samples. There are some other approaches to deal with concept-drift, such as FLORA [40].

### 3.3.4. Rule-based Method

Figure 3 shows the fragment of the NCCN guidelines. In addition the NCCN guideline, there are many other medical guidelines, such as ESMO (European Society for Medical Oncology) [41]. Some advanced medical institutions have put forward their own medical guidelines, e.g., the Ruijin guidelines for breast cancer (RJBG) by Ruijin Hospital, China. The purpose of rule-based methods is to

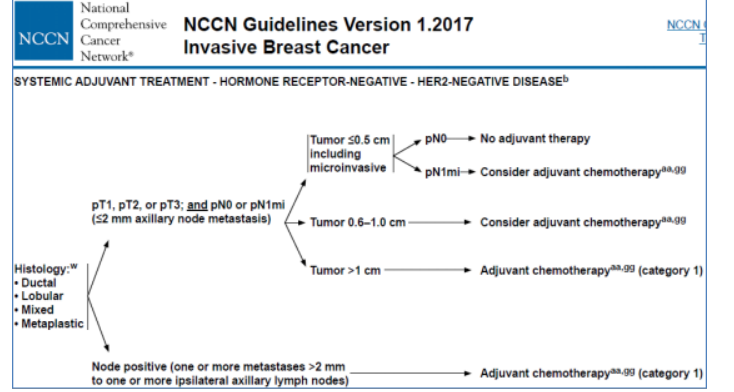


Figure 3: A fragment of the NCCN guidelines.

Table 2: The statistical information about the rules adopted in MdtDSS.

	CT	RT	TT	ET	Sum
NCCN	29 <sup>1</sup>	6	9	6	50
RJBG	18	12	8	12	50
Sum	47	18	17	18	100

<sup>1</sup> 29 represents that the NCCN guidelines have 29 rules with respect to chemotherapies (CT), and so on.

transform the medical guidelines into a form that the system can understand, and then to show doctors the rules on which the decision depends. It would be better to show the evidence from where the rule comes along with the rules.

The problem is that the current medical guidelines can not cover all situations, i.e., they are incomplete. For example, assume that each patient record has 50 categorical variables, and each variable has 2 values, then the size of a complete rule set is  $2^{50}$ , but in practice, the number of rules is much smaller than this value. With the advent of technology, the guidelines are updated gradually. Therefore, the hard-coded method is obviously inappropriate for implementing rule-based methods. We develop a rule-based system using a rule engine Drools<sup>3</sup>, which is a business rules management system (BRMS) written in Java. The rule engine links the main system through the network. We model medical guidelines as a set of rules which are stored in a file. Then, the guidelines in the DSS can be maintained by modifying the rule file directly. The rule itself (the decision-making process) and the evidence of the rule are both attached to the recommendation results, as shown in Figure 4(c). The statistical information about the rules adopted in our MdtDSS for breast cancer is shown in Table 2. There are 100 rules in our implementation. In practice, they can be applied to 76% of 3340 test samples. The left 24% of samples cannot be given the recommendation programs as they may lack the key value of rules or they cannot be covered by the current rules.

<sup>3</sup><http://drools.org/>

### 3.3.5. Recommendation Results Integration

The recommended treatment programs in terms of rule-based guidelines are commonly coarse-grained, such as *Need chemotherapy*, while the recommendation decisions from the  $k$ -NN classifier can be fine-grained, i.e., *the type of chemotherapy*. Since  $k$ -NN classifiers and rule-based methods have their own advantages and disadvantages, it is natural to integrate them together. There are different ways to combine them and we adopt a simple method, i.e., the rule-based approach as a filter. If the target case can be associated with a treatment program according to rules, we simply restrict the search scope of the  $k$ -NN classifier to cases whose programs are consistent with the one given by the rule. The final recommendation decision given by the  $k$ -NN classifier is a more informative one. The reason is that the guidelines are generally more authoritative than data-driven approaches. Unfortunately, sometimes rule-based methods can not produce a program. In this case, we directly set the results of the  $k$ -NN classifier as the final recommendations.

### 3.4. Prediction of the Degree of Difficulty of Decision Making for a Treatment Plan

To allow different process models to be adopted for different cases, we estimate the degree of difficulty of decision making for a treatment program for each case. As previously mentioned, the degree of disagreement of the voting results can reflect the degree of difficulty of decision making for the case to some degree. Thus, we utilize the feature vectors of historical cases along with the degree of disagreement of the voting results to learn a regression prediction model [42]. The loss function is defined as follows:

$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{i=1}^N \{H(V_i) - \theta^T \vec{x}^{(i)}\}^2 + \frac{\lambda_\theta}{2} \|\theta\|_2 \quad (11)$$

where  $N$  is the number of samples,  $H(V_i)$  is the degree of disagreement of sample  $D_i$ 's voting result, which is defined in Eq.(7), and  $\frac{\lambda_\theta}{2} \|\theta\|_2$  is the  $\ell_2$ -norm with controlling parameter  $\lambda_\theta$ . Then, the model parameter  $\theta$  can be determined by the stochastic gradient descent method, and with a learning step size  $\eta$ , the updated formula is as follows:

$$\theta := \theta + \eta((H(V_i) - \theta^T \vec{x}^{(i)})\vec{x}^{(i)} - \lambda_\theta \theta) \quad (12)$$

Once the  $\theta$  is determined, we can estimate the degree of disagreement of the voting result by calculating the value of  $F(x^{(i)}) = \theta^T \vec{x}^{(i)}$ . We conduct five-fold cross validation on dataset  $D$ , and choose the parameters leading to the smallest root mean squared error (**RMSE**), i.e., 0.84, and the  $H(V_i)$  is calculated according to the first-round voting results.

On the other hand, the information entropy of  $k$  nearest neighbors' class labels named  $I(x^{(i)})$  can reflect the uncertainty of the target case's recommendation results. Therefore, the disagreement degree and the uncertainty

can work together to predict the degree of difficulty of the decision by a linear combination as follows:

$$\hat{H} = \mu F(x^{(i)}) + (1 - \mu)I(x^{(i)}) \quad (13)$$

where  $\mu$  is the balance parameter and is set to 0.5 in MdtDSS. The value of  $\hat{H}$  can be used to select different process models. i.e., when  $\hat{H}$  is lower than a predefined threshold value (which is decided based on the considerations of risk and efficiency), a simplified workflow model (similar to the one in Figure 2(b)) is applied to this case, otherwise, the full workflow model (similar to the one in Figure 2(a)) is applied.

## 4. Implementations and Experiments

We implemented a MdtDSS for breast cancer treatment. The detailed structure of MdtDSS is shown in Figure 4(a). It follows the aforementioned framework and has three modules. The data management system includes a database and a converter that transfers data from an external data source, i.e., BCDB<sup>4</sup> (the SJTU Breast Cancer Database). The MDT voting system has an inner database as well as two collaborative components: ActiveMQ<sup>5</sup> and a customized workflow engine. The recommender system has two types of recommendation algorithms, namely the  $k$ -NN classifier and the rule-based method. The rule-based method is built on the rule engine. We implement the body frame of MdtDSS based on Rails<sup>6</sup> which is a server-side web application framework written in Ruby under the MIT License, and design the interactive user interface based on the most advanced responsive front-end framework, namely Foundation<sup>7</sup>. The design style is shown in the Figures 4(b)-4(d). The two engines of MdtDSS: the workflow engine and the rule engine, are both a Java project. ActiveMQ is an open source message broker. It is a bridge between the main system and the workflow engine, which decouples the process controlling module from the main system. Different platforms connect with each other through a network protocol, e.g., HTTP and STOMP.

To evaluate the performance of MdtDSS, we conducted a set of experiments and the results are detailed in this section. The experiments focus on three aspects: 1) different ways to show the recommendation results, 2) the performance of recommender systems and 3) the efficiency of the system.

### 4.1. How to Recommend

We discuss two methods, i.e., **W1** and **W2**, to show the recommendation results in the subsection 3.2. To explore how these two methods influence doctors' decisions,

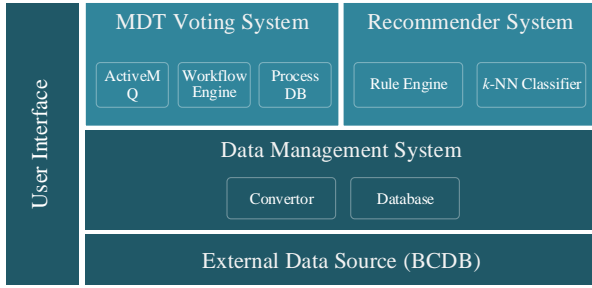
<sup>4</sup><http://bcd.b.learma.space:8080/>

<sup>5</sup><http://activemq.apache.org/>

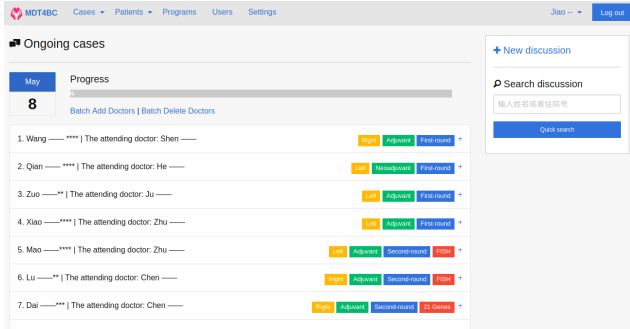
<sup>6</sup><http://rubyonrails.org/>

<sup>7</sup><https://foundation.zurb.com/>

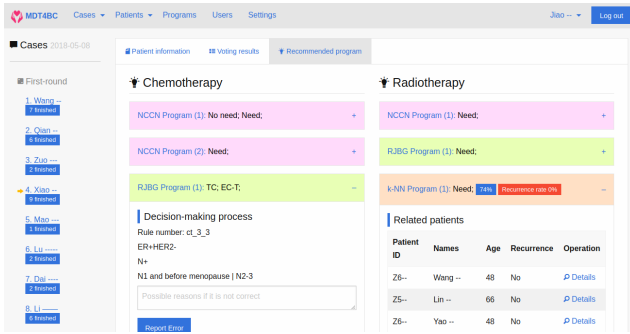




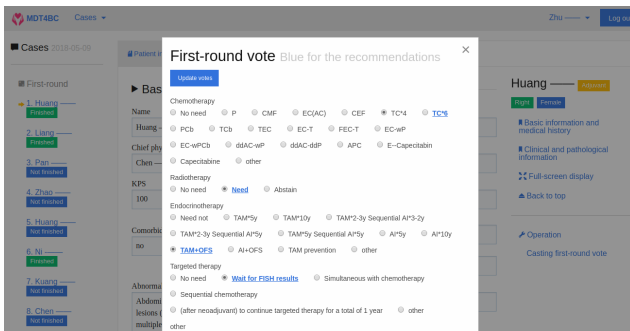
(a) MdtDSS Modules



(b) The homepage of MdtDSS.



(c) The MdtDSS's sub-recommendation system page.



(d) The voting page.

Figure 4: The implementations of MdtDSS

we conduct an A/B test on MdtDSS, i.e., an active method and a passive method. Each method runs online for two months (two different months for each method). Each month has approximately 130 new cases, and an average of 13 doctors participate in the discussion for each case.

#### 4.1.1. Evaluation Metrics

To better evaluate the system in different settings, we use two metrics for comparison.

**Mean information entropy (mIE).** mIE is defined as the mean of the  $H(V_i)$  values in Eq.(7) across all samples.

$$mIE = \frac{1}{\|T(t)\|} \sum_{D_i \in T(t)} H(V_i) \quad (14)$$

where  $T(t)$  is the set of samples being discussed in the time period  $t$ .

**Mean Click-through rate (mCTR).** CTR (click-through rate) is the ratio of doctors who click on the recommendation page to the total number of doctors involved. mCTR is defined as the mean of the CTR values across all samples.

$$mCTR = \frac{1}{\|T(t)\|} \sum_{D_i \in T(t)} CTR_i \quad (15)$$

$$CTR_i = \frac{1}{\|T(t)\|} \sum_{D_i \in T(t)} \frac{\#(\text{click-throughs of } D_i)}{\#(\text{doctors of } D_i)}$$

where  $\#(\text{click-throughs of } D_i)$  is the number of doctors who click on the recommendation page of sample  $D_i$ , and  $\#(\text{doctors of } D_i)$  is the number of doctors involved.

#### 4.1.2. Experiment Comparisons

Based on **W1** and **W2** settings, we conduct experiments to compare the values of mIE and mCTR in different time periods  $t1$  and  $t2$ , where  $\|T(t1)\| = 266$  and  $\|T(t2)\| = 250$ . The results are shown in Figure 5. We find that the mCTR measures of both **W1** and **W2** are higher than 70%, which means that in the MdtDSS, the recommender system is accepted by the doctors. At least, they are willing to refer to the advice from the machine decision maker. Furthermore, an interesting observation is that **W2** does not only reduce the value of mIE, which means doctors probably change their votes if they are alerted by the recommender system but also improves the performance of mCTR. This may be because doctors check the details before they change their votes according to the recommended results. We argue that the setting of **W2** is better than that of **W1** if the recommender system has a very high recommendation accuracy since the setting of **W2** can promote interaction between humans and the system. However, if the recommendation is not accurate, the setting of **W2** will degrade the user experience. In such a case, **W1** would be a better choice.

#### 4.2. Recommendation Evaluation

To demonstrate the different effects of the proposed  $k$ -NN classifiers with three different fitness functions, i.e.,  $\mathcal{L}_{GA1}$ ,  $\mathcal{L}_{GA2}$ , and  $\mathcal{L}_{GA3}$  to the weight learning process, we compare them over the real dataset.

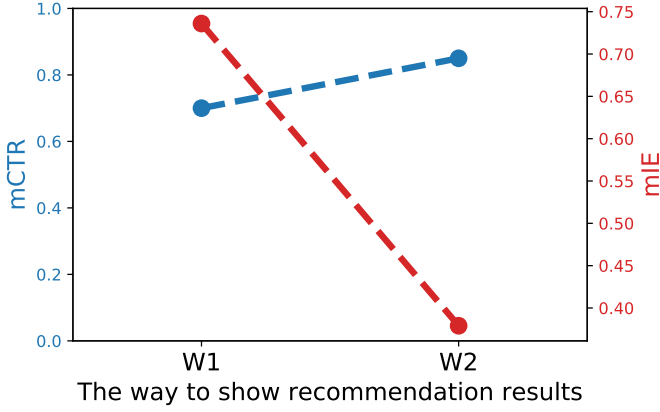
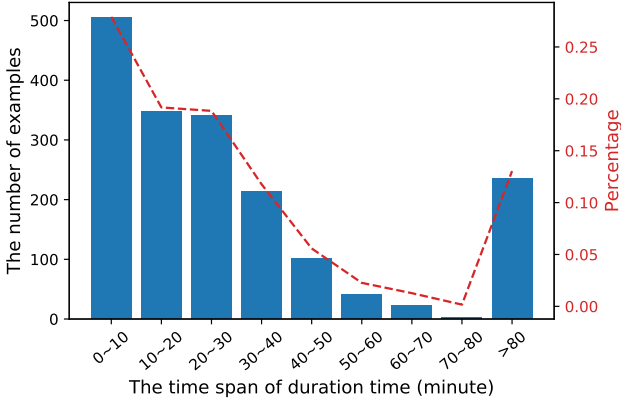
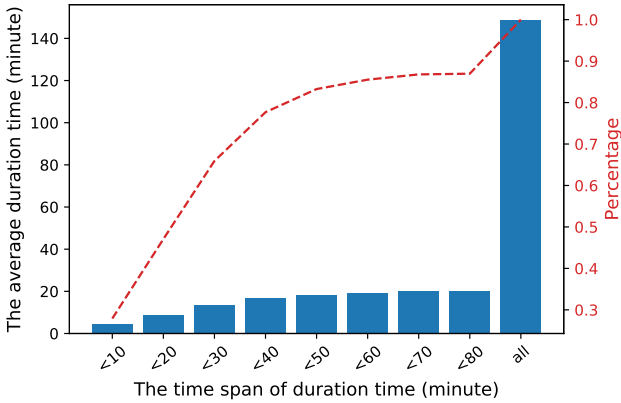


Figure 5: Performance comparisons of **W1** and **W2**.



(a) The number of cases in the second-round vote over different time spans.



(b) The average duration time of the second-round vote over different time spans.

Figure 6: The performance of the system efficiency.

#### 4.2.1. Dataset

The size of the real dataset from MdtDSS, which we use to examine the recommender system offline, is 3340 which is smaller than the number of patients in BCDB because

not all cases in BCDB are imported into MdtDSS. But the number is sufficient to train our  $k$ -NN classifiers. Each sample has 48 attributes and the main class and subclass information are shown in Table 1. Without loss of generality, here we only discuss the performance of our proposed methods on the chemotherapy classification problem.

#### 4.2.2. Evaluation Metrics

The recommendation quality is measured by the hit rate (**HR**) and the binary classification error rate (**BER**). **HR** is defined as follows:

$$HR = \frac{\#hits}{\#samples} \quad (16)$$

where  $\#samples$  is the total number of samples, and  $\#hits$  is the number of samples whose labeled class in the test set is recommended (i.e., hit) in the size- $N$  recommendation list. **BER** evaluates the error ratio of serious misclassification errors, which is defined as follows:

$$BER = \frac{\#ErrorHits}{\#samples} \quad (17)$$

where  $\#ErrorHits$  is the number of samples whose recommended result contains a subclass belonging to a main class that is different from the true main class of the target sample.

#### 4.2.3. Recommender Systems' Details

The dataset is first divided into two parts according to the decision completion time of samples and a cut-off point time, i.e., the set of samples with the decision completion time before the cut-off point time is considered as the repository which is used to find the  $k$  nearest neighbors of the target sample, and the remaining samples are further divided into two parts randomly based on the time to form the training set and the test set. The genetic algorithm is conducted on the training set along with the repository to find the best weights. Then, the recommendation systems including the  $k$ -NN classifier and the rule-based method are evaluated on the test set. Note that, to relieve the concept-drift problem, we only consider the most recent 500 samples in the repository when searching  $k$  nearest neighbors.

#### 4.2.4. Experiment Comparisons

We measure the **HR** and **BER** performances of  $k$ -NN classifiers as well as of the rule-based method, referred to as RULE in this section. Three  $k$ -NN classifiers with different fitness functions to the genetic algorithm are named  $k$ NN,  $k$ NNB and  $k$ NND, respectively, where  $k$ NN is a basic  $k$ -nearest neighbor classifier playing the role of baseline, using the fitness function mentioned in Eq.(5),  $k$ NNB considers the different risk between serious misclassification and minor misclassification and takes  $\mathcal{L}_{GA2}$  in Eq.(6) as the fitness function, and  $k$ NND automatically adjusts the penalty of each misclassification according to the degree of

Table 3: Performance comparisons of  $k$ -NN classifiers and the rule-based method

Learning weights by the <b>Top-1</b> recommendation						
Methods	Top-1		Top-2		Top-3	
	HR	BER	HR	BER	HR	BER
RULE <sup>1</sup>	0.81	0.18	0.81	0.18	0.81	0.18
$k$ NN	0.68	0.15	<b>0.87</b>	0.33	0.91	0.51
$k$ NNB	0.69	<b>0.12</b>	0.86	0.26	0.90	0.40
$k$ NND	<b>0.71</b>	0.15	0.86	<b>0.25</b>	<b>0.94</b>	0.41
$k$ NND-R	<b>0.71</b>	0.15	0.86	<b>0.25</b>	(0.90	<b>0.25)</b>
Learning weights by the <b>Top-2</b> recommendation						
Methods	Top-1		Top-2		Top-3	
	HR	BER	HR	BER	HR	BER
RULE	0.81	0.18	0.81	0.18	0.81	0.18
$k$ NN	<b>0.71</b>	<b>0.17</b>	<b>0.91</b>	0.39	0.93	0.54
$k$ NNB	0.67	<b>0.17</b>	0.82	<b>0.21</b>	0.88	0.32
$k$ NND	0.66	0.19	0.87	0.28	<b>0.96</b>	0.44
$k$ NND-R	0.66	0.19	0.87	0.28	0.90	<b>0.28</b>

<sup>1</sup> **Bold** values represent the best performance in the experiments, however, the results of the  $k$ -NN classifier are not compared with the results of the RULE method since the Top- $k$  setting has no effect on the results of the rule-based method even though the rule-based method also recommends multiple treatment programs. The treatment programs recommended by the rule-based method have the same priority.

disagreement of voting results and takes  $\mathcal{L}_{GA3}$  in Eq.(9) as the fitness function. We only show the results of the combination of the  $k$ NND and the RULE, named  $k$ NND-R, since  $k$ NND has the best performance in the experimental study. For the  $k$ NND-R, we only use the main classes of therapies (treatment programs) recommended by the RULE to filter the recommended lists of the  $k$ NND classifier. All the parameters are empirically tuned to the optimal values for  $k$ -NN classifiers, i.e., the trade-off parameter  $\alpha = 0.9$  and  $k = 25$  for the number of nearest neighbors.

We first consider the Top-1 recommendation when we learn the weights of the attributes, i.e., we measure the value of the fitness function  $\mathcal{L}_{GA}$  and each iteration is only based on the Top-1 recommendation. But in the test phase, three different settings, i.e., Top-1, Top-2, and Top-3 are explored. As for the rule-based method, it recommends one to multiple treatment programs which have the same priority, thus the performance of the rule-based method does not change in the three settings. It is reasonable to compare the results of RULE with the best performance of  $k$ -NN classifiers. The experiment results are shown on the top of Table 3. We observe that the values of the **HR** and **BER** both increase with the size of the recommendation lists, but the growth rate of the **BER** is higher than that of the **HR** if no larger value is set for the penalty of a serious misclassification problem. The **BER** performance of  $k$ NNB,  $k$ NND and  $k$ NND-R is all better than that of  $k$ NN. This verifies that it works if we make the penalties of serious misclassification and of the minor misclassification different when learning the weights. In

the Top-3 setting,  $k$ NND has the best **HR** performance, i.e., 94%, however, the value of the **BER** is as large as 41%, which cannot be accepted. Fortunately, if we use the recommendation lists of RULE to filter the recommendation items of  $k$ NND, the value of the **BER** can decrease to 25%, while the value of the **HR** decreases a little, but it is still larger than 90%. The performance of  $k$ NND and  $k$ NND-R in the Top-1 and Top-2 settings is the same because the main classes of the Top-2 treatment programs recommended by the  $k$ NND are no different from those of RULE. Furthermore, the value of the **BER** of  $k$ NND-R is larger than that of RULE since the latter cannot provide recommendation lists for all samples as previously mentioned. The same phenomenon can be observed on the bottom of Table 3, which explores the performance when the Top-2 recommendations are considered in the process of learning the weights. Note that, as long as the size of the recommendation list of the samples, which are labeled *No need*, is larger than 2, this case must be a serious misclassification since the class *No need* has no subclass. In such a case, we measure the value of  $\mathcal{L}_{GA}$  according to the Top-1 recommendation. Then, we find all approaches have a similar performance with the results shown on the top of Table 3 but with a higher value. We argue that the pair of results in the parentheses is best in our experiment studies, i.e., the result with a lower **BER** but with a higher **HR**. An unexpected observation is that the value of **BER** of RULE is as high as 18%, which means, in practice, the medical guidelines are not always correct. This is still a challenge for the recommendation.

#### 4.3. System Efficiency

After using our MdtDSS, the medical institution can save **about half an hour** when casting their second-round votes each day compared to the situation without a DSS, according to the feedback from the doctors. We also explore system efficiency through the analysis of the workflow engine records, i.e., the duration time of the activity which is calculated by the end time minus the start time. Since the first-round vote is a non-centralized vote as previously mentioned, its duration time is not meaningful. Instead, the characteristic of the second-round vote reflects the quality of MdtDSS to some degree.

Figure 6(a) shows the number of cases in the second-round vote over different time spans as well as the percentage of cases over each time span. We find that the duration/cost time of most cases is smaller than 10 minutes, i.e., there are 500 cases, which is 25% of the total number of the cases (1810). The average duration time of these cases is 4.36 minutes which is shown in Figure 6(b). The number of cases decreases as the duration time in the second-round vote increases. Note that, approximately 13% of cases have a duration time larger than 80 minutes. However, this does not mean that doctors really spend this amount of time discussing the treatment program for these cases. Rather, this usually indicates that the voting process of the case is blocked in this meeting

for some reason, such as a lack of supplementary medical examinations. Therefore, the waiting time for the supplementary reports can be explained by the duration time being longer than usual. As Figure 6(b) shows, the average duration time is about 20 minutes when excluding the samples whose duration time is larger than 80 minutes. We argue that the real average duration time is much smaller than 20 minutes. Consider that multiple cases can start the second-round vote simultaneously, but they must be voted in order. In such a case, the longest duration time of the case is actually the sum of the duration times of all cases. Assume that the number of cases each day is the same, then the number of cases over each duration time span should be close to each other. This characteristic can be observed in Figure 6(a), i.e., the first four intervals have a close number of cases. Thus, the first case each day is very likely finished in 10 minutes, and the average duration time over these cases is 4.36 minutes, which is more reliable for representing the efficiency of MdtDSS.

Furthermore, since the simplified process is adopted for routine cases, the cost time of the first-round vote of these cases is saved. Assume that the average cost time of the first-round vote is also 4.36 minutes, then the total time saved by this simplified process for each doctor is  $4.36 * \bar{n}$  minutes, where  $\bar{n}$  is the average number of routine cases each day. In MdtDSS, those cases whose rating (degree of difficulty) is no more than 2 stars are considered routine cases since this setting can achieve the highest **HR** performance over the selected routine cases, which is about 85% in Top-1 recommendation. It makes  $\bar{n} = 2$  and thus each doctor would further save about 8.72 minutes each day compared to the situation where all medical cases follow the same workflow.

## 5. Discussion

After showing the details of MdtDSS, we would discuss two important issues as follows:

**Q1** How to speed up the medical treatment decision-making process?

In order to speed up the process, the first step is to streamline it through workflow technology, i.e., to coordinate the doctor's decision-making process automatically. In addition, we can integrate information from different sources to better support the doctor's decision making.

The next step is to provide recommendations to the doctor so that the doctor's decision is made more quickly and easily. However, if the recommendations are not precise enough, the doctor will not trust it and the recommendations can become a kind of disturbance. Therefore, an appropriate way to provide the recommendation results to doctors during the decision-making process needs to be found.

A further step is to optimize the process by providing different process models based on case classification and

recommendation results. In addition to the two process models discussed in this paper, an extreme process model is a system which gives the treatment program directly while only one doctor is responsible for the review.

Which step is the more appropriate is actually decided by recommendation accuracy, which is the next issue.

**Q2** How to improve the decision-making accuracy of the machine decision maker?

Different from traditional approaches that only focus on the **HR** metric, we take into consideration the **HR** and the **BER** simultaneously since the **HR** reflects the accuracy of the decision results while the **BER** represents the ratio of the mistakes that cannot be ignored by the doctors.

To achieve this goal, we devise  $k$ -NNB and  $k$ -NND with two different fitness functions which are used in the process of learning the weights of attributes. Furthermore, we combine the  $k$ -NN classifiers and the rule-based method to further decrease the value of **BER** without greatly violating the performance of **HR**.

The new samples are added into the repository continuously, which means the  $k$ -NN classifier is a self-learning recommender system. This helps us address the concept-drift problem if we set a window to filter the current samples when finding the most  $k$  nearest neighbors. The data quality is the key for  $k$ -NN classifiers and other approaches. Rare cases always appear and these cases decrease the recommendation accuracy if they are simply put them into the training set. A simple way to address this is to judge whether each case can be put into the training set or not. However, if rare cases are not put into the training set, this increases the risk that concept-drift is not reflected in time and similar rare cases cannot be processed properly.

## 6. Conclusion and Future Work

A real-world decision support system for multi-disciplinary treatment, namely MdtDSS, is introduced in this paper. Different from traditional decision support systems that only focus on the correctness of recommendation results, the MdtDSS considers accuracy and efficiency simultaneously. To achieve this goal, we apply a workflow to provide different process models to process different cases on the one hand, and on the other hand, we explore multiple recommendation approaches, including  $k$ -NN classifiers and rule-based methods to provide the machine decision results. The experiment results show that the MdtDSS has an advantages in both accuracy and efficiency. Moreover, MdtDSS has been put into practical use on a daily basis.

There are several research topics for potential future study. Firstly, a method combining the  $k$ -NN methods and the rule-based method can be explored further, e.g., we can auto-tune the number of nearest neighbors until the main class recommended by the  $k$ -NN classifier is consistent with the rule-based method. Secondly, we can utilize

the behavior information of doctors in the platform, such as the time cost incurred by doctors in designing a more informative fitness function. Thirdly, there are several other ways to show the recommendation results, such as showing the recommendation results only in the final round of the voting stage, which affects doctors in different ways.

## 7. Acknowledgments

This work is partially supported by China National Science Foundation (Granted Number 61272438,61472253), Research Funds of Science and Technology Commission of Shanghai Municipality (Granted Number 15411952502, 14511107702) and Cross Research Fund of Biomedical Engineering of Shanghai Jiao Tong University (YG2015MS61).

## References

- [1] C. Taylor, A. J. Munro, R. Glynne-Jones, C. Griffith, P. Trevatt, M. Richards, A. J. Ramirez, Multidisciplinary team working in cancer: what is the evidence?, *BMJ: British Medical Journal* (Online) 340.
- [2] D. J. Power, Decision support systems: a historical overview, in: *Handbook on Decision Support Systems 1*, Springer, 2008, pp. 121–140.
- [3] D. J. Power, R. Sharda, F. Burstein, *Decision support systems*, Wiley Online Library, 2015.
- [4] H. L. Kim, M. R. Puymon, M. Qin, K. Guru, J. L. Mohler, Nccn clinical practice guidelines in oncology.
- [5] Y. Zhang, M. Chen, D. Huang, D. Wu, Y. Li, idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization, *Future Generation Computer Systems* 66 (2017) 30–35.
- [6] M. Wang, M. Liu, J. Liu, S. Wang, G. Long, B. Qian, Safe medicine recommendation via medical knowledge graph embedding, *arXiv preprint arXiv:1710.05980*.
- [7] I. Zliobaitė, M. Pechenizkiy, J. Gama, An overview of concept drift applications, in: *Big Data Analysis: New Algorithms for a New Society*, Springer, 2016, pp. 91–114.
- [8] A. Jula, E. Sundararajan, Z. Othman, Cloud computing service composition: A systematic literature review, *Expert Systems with Applications* 41 (8) (2014) 3809–3824.
- [9] L. Chen, M. Li, J. Cao, Eca rule-based workflow modeling and implementation for service composition, *IEICE transactions on information and systems* 89 (2) (2006) 624–630.
- [10] N. Ploskas, I. Athanasiadis, J. Papathanasiou, N. Samaras, A collaborative spatial decision support system for the capacitated vehicle routing problem on a tabletop display, in: *Decision support systems IV-information and knowledge management in decision processes*, Springer, 2015, pp. 26–36.
- [11] J. I. Kim, J. Kim, M. Fischer, R. Orr, Bim-based decision-support method for master planning of sustainable large-scale developments, *Automation in Construction* 58 (2015) 95–108.
- [12] N. Sahebjamnia, R. Tavakkoli-Moghaddam, N. Ghorbani, Designing a fuzzy q-learning multi-agent quality control system for a continuous chemical production line—a case study, *Computers & Industrial Engineering* 93 (2016) 215–226.
- [13] E. S. Berner, T. J. La Lande, Overview of clinical decision support systems, in: *Clinical decision support systems*, Springer, 2016, pp. 1–17.
- [14] R. A. Miller, Medical diagnostic decision support systems past, present, and future: a threaded bibliography and brief commentary, *Journal of the American Medical Informatics Association* 1 (1) (1994) 8–27.
- [15] E. S. Berner, *Clinical decision support systems*, Vol. 233, Springer, 2007.
- [16] M. A. Musen, B. Middleton, R. A. Greenes, Clinical decision-support systems, in: *Biomedical informatics*, Springer, 2014, pp. 643–674.
- [17] A. X. Garg, N. K. Adhikari, H. McDonald, M. P. Rosas-Arellano, P. Devereaux, J. Beyene, J. Sam, R. B. Haynes, Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review, *Jama* 293 (10) (2005) 1223–1238.
- [18] D. L. Hunt, R. B. Haynes, S. E. Hanna, K. Smith, Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review, *Jama* 280 (15) (1998) 1339–1346.
- [19] L. Moja, K. H. Kwag, T. Lytras, L. Bertizzolo, L. Brandt, V. Pecoraro, G. Rigon, A. Vaona, F. Ruggiero, M. Mangia, et al., Effectiveness of computerized decision support systems linked to electronic health records: a systematic review and meta-analysis, *American journal of public health* 104 (12) (2014) e12–e22.
- [20] N. T. Thong, et al., Hifcf: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis, *Expert Systems with Applications* 42 (7) (2015) 3682–3701.
- [21] S. Pradhan, V. Gay, S. Nepal, An innovative approach to derive trust from social networks and to improve the matching in dental care recommendation systems., in: *PACIS*, 2016, p. 328.
- [22] Y. Chen, J. E. Argentinis, G. Weber, Ibm watson: how cognitive computing can be applied to big data challenges in life sciences research, *Clinical therapeutics* 38 (4) (2016) 688–701.
- [23] A. C. Chozas, S. Memeti, S. Pillana, Using cognitive computing for learning parallel programming: An ibm watson solution, *Procedia Computer Science* 108 (2017) 2121–2130.
- [24] M. G. Zauderer, A. Guclalp, A. S. Epstein, A. D. Seidman, A. Caroline, S. Granovsky, J. Fu, J. Keesing, S. Lewis, H. Co, et al., Piloting ibm watson oncology within memorial sloan ketterings regional network. (2014).
- [25] R. High, The era of cognitive systems: An inside look at ibm watson and how it works, IBM Corporation, Redbooks.
- [26] J. Powles, H. Hodson, Google deepmind and healthcare in an age of algorithms, *Health and technology* 7 (4) (2017) 351–367.
- [27] J. De Fauw, P. Keane, N. Tomasev, D. Visentin, G. van den Driessche, M. Johnson, C. O. Hughes, C. Chu, J. Ledsam, T. Back, et al., Automated analysis of retinal imaging using machine learning techniques for computer vision, *F1000Research* 5.
- [28] C. Chu, J. De Fauw, N. Tomasev, B. R. Paredes, C. Hughes, J. Ledsam, T. Back, H. Montgomery, G. Rees, R. Raine, et al., Applying machine learning to automated segmentation of head and neck tumour volumes and organs at risk on radiotherapy planning ct and mri scans, *F1000Research* 5.
- [29] E. M. Kesson, G. M. Allardice, W. D. George, H. J. Burns, D. S. Morrison, Effects of multidisciplinary team working on breast cancer survival: retrospective, comparative, interventional cohort study of 13 722 women, *bmj* 344 (2012) e2718.
- [30] R. H. Miller, I. Sim, Physicians use of electronic medical records: barriers and solutions, *Health affairs* 23 (2) (2004) 116–126.
- [31] Y. Xu, N. Li, M. Lu, R. Myers, E. Dixon, R. Walker, H. Quan, Developing and validating electronic medical record based case definitions for liver diseases and comorbidities, *International Journal for Population Data Science* 1 (1).
- [32] W. M. Van Der Aalst, M. La Rosa, F. M. Santoro, *Business process management* (2016).
- [33] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician* 46 (3) (1992) 175–185.
- [34] M. T. Hagan, H. B. Demuth, M. H. Beale, et al., *Neural network design*, Vol. 20, Pws Pub. Boston, 1996.
- [35] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, *IEEE Intelligent Systems and their applications* 13 (4) (1998) 18–28.
- [36] P. C. Cheeseman, M. Self, J. Kelly, W. Taylor, D. Freeman, J. C. Stutz, Bayesian classification., in: *AAAI*, Vol. 88, 1988, pp. 607–611.

- [37] S. Boriah, V. Chandola, V. Kumar, Similarity measures for categorical data: A comparative evaluation, in: Proceedings of the 2008 SIAM International Conference on Data Mining, SIAM, 2008, pp. 243–254.
- [38] D. Whitley, A genetic algorithm tutorial, *Statistics and computing* 4 (2) (1994) 65–85.
- [39] R. M. Gray, *Entropy and information theory*, Springer Science & Business Media, 2011.
- [40] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, *Machine learning* 23 (1) (1996) 69–101.
- [41] E. Van Cutsem, B. Nordlinger, A. Cervantes, E. G. W. Group, Advanced colorectal cancer: Esmo clinical practice guidelines for treatment, *Annals of oncology* 21 (suppl\_5) (2010) v93–v97.
- [42] D. C. Montgomery, E. A. Peck, G. G. Vining, *Introduction to linear regression analysis*, Vol. 821, John Wiley & Sons, 2012.