IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 13, 2016, accepted February 14, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2672975

Semantic Inference on Clinical Documents: Combining Machine Learning Algorithms With an Inference Engine for Effective Clinical Diagnosis and Treatment

SHUO YANG[1], (Member, IEEE), RAN WEI[2], JINGZHI GUO[1], (Member, IEEE), AND LIDA XU[3], (Senior Member, IEEE)
[1]Faculty of Science and Technology, University of Macau, Taipa 999078, China
[2]Department of Microbiology, Rutgers University, Newark, NJ 07103 USA
[3]Department of Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA 23529 USA

Corresponding author: S. Yang (yb37416@umac.mo)

This work was supported by the University of Macau Research under Grant MYRG2015-00043-FST.

ABSTRACT Clinical practice calls for reliable diagnosis and optimized treatment. However, human errors in health care remain a severe issue even in industrialized countries. The application of clinical decision support systems (CDSS) casts light on this problem. However, given the great improvement in CDSS over the past several years, challenges to their wide-scale application are still present, including: 1) decision making of CDSS is complicated by the complexity of the data regarding human physiology and pathology, which could render the whole process more time-consuming by loading big data related to patients; and 2) information incompatibility among different health information systems (HIS) makes CDSS an information island, i.e., additional input work on patient information might be required, which would further increase the burden on clinicians. One popular strategy is the integration of CDSS in HIS to directly read electronic health records (EHRs) for analysis. However, gathering data from EHRs could constitute another problem, because EHR document standards are not unified. In addition, HIS could use different default clinical terminologies to define input data, which could cause additional misinterpretation. Several proposals have been published thus far to allow CDSS access to EHRs via the redefinition of data terminologies according to the standards used by the recipients of the data flow, but they mostly aim at specific versions of CDSS guidelines. This paper views these problems in a different way. Compared with conventional approaches, we suggest more fundamental changes; specifically, uniform and updatable clinical terminology and document syntax should be used by EHRs, HIS, and their integrated CDSS. Facilitated data exchange will increase the overall data loading efficacy, enabling CDSS to read more information for analysis at a given time. Furthermore, a proposed CDSS should be based on self-learning, which dynamically updates a knowledge model according to the data-stream-based upcoming data set. The experiment results show that our system increases the accuracy of the diagnosis and treatment strategy designs.

INDEX TERMS Big data, case-based reasoning, clinical diagnosis, decision tree, data stream mining, disease detection, electronic health record, medical record, semantic integration.

I. INTRODUCTION

Given the recent dramatic progress that global endeavors have made in health care, it is surprising that human errors remain the leading cause of death even in developed countries such as the US [1]. Among all of the contributing factors, errors related to medication are the most common category in medical practices [2]. A large number of adverse drug effects are reported annually [3], [4], with statistics revealing that approximately 50% are preventable [3]. Subgroups of this category of error include incorrect prescription, drug dose and administration. In addition, incorrect diagnosis is another typical human error, which causes fundamentally wrong medical decisions that lead to serious consequences [5]. To reduce the risk of human error as well as the workload of the medical staff, the application of medical software has long been suggested as a possible tool [6]–[8], [58]. According to [2], different strategies of software design are used to solve the two aforementioned problems. To prevent medication errors, the application should be designed as an automated database with historical and current medical records of a patient, as well as other key information, including all prescription and personal allergic reaction documents to prevent any inappropriate prescriptions and provide warnings on them. In addition, computer-assisted diagnosis software is used to increase the accuracy of the diagnosis and decrease the time that is needed for decision making. A few previous research studies have shown the feasibility of computer-assisted clinical information access and practice in terms of significantly reduced incidence of medical errors or improved accuracy of diagnosing multiple diseases [9]–[11]. Currently, one popular strategy in the implementation of automated clinical practices is the integration of clinical decision support systems (CDSS) and health information systems (HIS) to directly read electronic health records (EHR) for analysis [12], [13]. This method greatly reduces the cost of training and the time required for entering massive amounts of patient-related data during each visit [14], [15], and it dynamically reconstructs the diagnosis model according to the real-time fluctuations of the patient's condition [16].

However, several remaining problems still challenge the prospect of CDSS applications:

(1) **Rapid growth of data**. With the enormous number of patients who are exposed to all types of tests, the size of the medical information database is increasing rapidly in millions of multi-dimensional records (e.g., physical indicators of the human body collected by smart clothing [59], [60], [61]) every day, which makes classical data mining methodologies no longer sufficient [17]. Therefore, the low efficacy of data inquiry and mining becomes an emerging problem for future data analysis [18].

(2) **Cross-context interoperability problem**. Historically, multiple clinical document standards were selected by different versions of EHRs. Because no perfectly unified vocabulary and data format across all versions of EHRs have been established, misinterpretation of health information by heterogeneous EHRs could occur; similarly, different versions of HIS could use various default clinical terminologies to define input health data [16]. As a result, a series of information islands could be left behind, failing to be merged with other data sources for more systemic analyses.

(3) **Demand for personalization and profession**. As the name says, CDSS is still regarded as a ''clinical decision'' support system, with little effort being made toward improvement of the patient's experience [62]. Currently, it is very common for patients to search doctors or hospitals [57] according to their preference (e.g., distance from home) or symptoms (e.g., which hospital specializes in cardiovascular diseases), but too much information could be retrieved [19]. Unfortunately, at present, such personalized and professional demands cannot be completely satisfied; however, with the enormous amount of patient care data from different hospitals in EHRs, CDSS could be designed to analyze all of the historically hospitalized patient care data and make recommendations. This arrangement will help the patients to make optimal decisions according to their personalized demands.

To address these challenges, this article proposes a personalized and professional clinical diagnosis and treatment system (CDTS) that combines machine learning algorithms with an inference engine. Specifically, this article makes the following contributions:

- ➢ Devise a voted ensemble classification algorithm that is suitable for data stream mining to meet the big data computational demand;
- ➢ Implement clinical tabular document syntax (DocLang) to integrate heterogeneous clinical information for accurate disease diagnosis and treatment. This approach has two advantages: (1) maintaining semantic consistency among heterogeneous contexts; and (2) facilitating automatic document understanding and processing across different contexts, because DocLang is not only a document representation language but also a rule language;
- ➢ Support personalized and professional demands based on the patients' and doctors' features. Patients are able to receive suggestions (e.g., disease diagnosis or treatment suggestion) without going to hospitals according to their preference (e.g., choosing a clinician in another country).

The remainder of this article is organized as follows. Section II presents related studies on decision tree and case-based reasoning. In Section III, we introduce the framework of the clinical diagnosis and treatment system (CDTS). Section IV presents a clinical tabular document model (CTDM) that is used for clinical document representation in CDTS. Section V discusses a new decision-tree-based data stream mining algorithm to provide rules for the semantic inference algorithm proposed in Section VI. Section VII shows the performance of our proposal via experiment results and analysis. Finally, we conclude this article in Section VIII.

II. RELATED WORK

A. DECISION TREE

The theoretical foundation of disease diagnosis in this paper is a decision tree-based classification algorithm. A decision tree (DT) is a typical classification algorithm. Let DT be a function of examples. The class label for an example e is obtained by passing the example from the root down to a leaf, to be tested on a different attribute at each non-leaf node, and then, following the branch that corresponds to the attribute's value in e [20]. It is also feasible to improve traditional DT models by integrating complex tests on internal nodes and complex classification rules on leaf nodes [20].

Classical decision tree learners (e.g., ID3, C4.5 [21], and CART [22]) require all of the training datasets to be loaded into memory. When the distributions of the datasets change, these learners must reconstruct new tree models rather than updating old ones. In particular, these classical algorithms are not applicable to data stream mining, where potentially there is no upper bound on the number of instances that arrive sequentially [20]. In 2000, Domingos and Hulten [23] developed the Hoeffding tree algorithm, which is an incremental decision tree induction methodology that enables learning from successive data streams. In 2001, Oza and Russel designed the Ensemble Hoeffding Tree [24], an online bagging method that integrates some very fast decision tree classifiers. Bifet, Holmes, Pfahringer, Kirkby and Gavalda [25] developed the Adaptive Size Hoeffding Tree approach, which was derived in 2009 and was a very fast decision tree that additionally sets a maximum number (θ) of split nodes: if the number of splitting nodes is larger than θ, the algorithm will delete some nodes to minimize the tree size. In addition, it can handle concept-drift data streams.

B. CASE-BASED REASONING

The theoretical foundation of disease treatment in this paper is case-based reasoning. Ocampo *et al.* in [26], conclude that case-based reasoning (CBR) is an approach in which people's current problem solution is referred from their past experience. According to [27], a case is a contextualized piece of knowledge that represents a certain experience, and it contains previous lessons and context in which that lesson can be applied. It can also be defined as a complete description of a problem, with its respective solution and also an assessment of the solution's efficiency [28]. Thus, the CBR strategy is simple: when confronting a new problem, it first reviews past cases to identify the most similar one(s), and then, it adopts corresponding solution(s) instead of building a complex and explicit model.

Technically, CBR is commonly described as an iterative procedure that can be divided into four steps [26]:

Step 1 (Case Recovery): This step includes three tasks [26]: (1) identify the characteristics that describe a new problem; (2) locate the relevant cases; and (3) choose the best candidate(s) among the relevant cases. Two of the most currently used techniques are the recovery of the closest neighbor and inductive recovery [31], [32].

Step 2 (Solution Suggestion): Usually, when a case is recovered, an analysis is conducted to determine its similarity with the current problem. This step identifies the differences between the current case and the recovered cases and, afterward, applies constraints (e.g., formulas, rules) to those differences for determination of the final solution [26]. In general, there are two types of solution suggestions: (i) structural adaptation, which applies rules and formulas directly to the stored solutions, and (ii) derived suggestion, which generates new solutions by reutilizing the original rules and formulas for the recovered solution [27].

Step 3 (Solution Revision and Confirmation): After the solution is suggested, it is necessary to evaluate the fitness of the solution to a new case. If a suggested solution to the new problem is not suitable, then it shall be updated, and the algorithm would learn from mistakes. Basically, two points are undertaken: (i) the applicability of the solution to the new case is determined by experts, and (ii) if there is a change in the solution, then the case will be updated before saving it [26].

Step 4 (Retention/Remember): This step incorporates the useful information in a new solution into a knowledge base [26]. It involves two key points: (i) select the specific information to be retained from the new case, and (ii) integrate it into the knowledge base [29], [30].

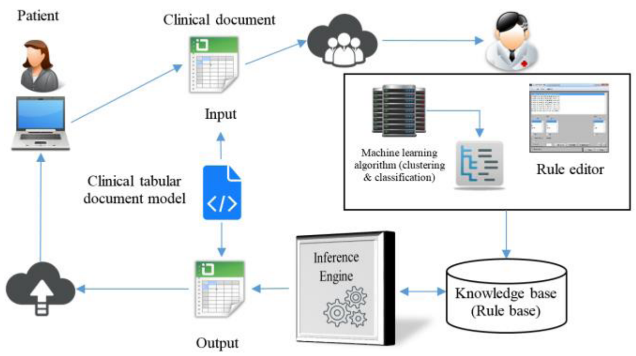

Patient
Clinical document
Input
Clinical tabular document model
Machine learning algorithm (clustering & classification)
Rule editor
Inference Engine
Knowledge base (Rule base)
Output

FIGURE 1. Framework of the CDTS.

III. PROPOSED SOLUTION

A. FRAMEWORK OF THE CLINICAL DIAGNOSIS AND TREATMENT SYSTEM (CDTS)

To facilitate the disease detection and treatment, this paper proposes a new clinical diagnosis and treatment system (CDTS), which is conceptually represented in Fig. 1. In CDTS, a new clinical tabular document model is provided as a standard for clinical document representation. After a patient chooses preferred doctors or hospitals, the clinical documents of the patient will be transferred to them through the network. The critical component from the perspective of the doctor or hospital is a semantic inference mechanism that consists of two stages: knowledge extraction and reasoning. Its objectives are as follows: (1) reading the input clinical document and transforming it into an instance for machine learning algorithms (e.g., clustering, classification) as well as a set of input facts and rules that are compatible with the inference engine; (2) collecting expert experience of professional physicians as expert rules and recording them through a Rule Editor; and (3) making semantic inferences and obtaining the results of the inference rules and transforming them into the output clinical document. In the following sections, we explain each of these components in detail.

IV. CLINICAL TABULAR DOCUMENT MODEL

Each patient medical record corresponds to a medical examination on a given date [33]. A formal definition of a medical record is given in Definition 4.1.

Definition 4.1 (Medical Record): Assume that $\sum = \{e_1, e_2, \ldots, e_k\}$ is the set of examinations and

$\Psi = \{p_1, p_2, \ldots, p_n\}$ is the set of patients; a medical record r models a list of examinations $\mathrm{E} = \{e_1, e_x, \ldots, e_y\} \subseteq \sum$ that are performed on a patient $p_j \in \Psi$ on a given date.

In the treatment space, each patient is represented by a vector, which represents the patient's treatment history. A vector element $\mathrm{c} = (p_i, E_i, t_i)$ corresponds to a treatment $\mathrm{t_i}$ to patient $\mathrm{p_i}$ under an examination $\mathrm{E_i}$. A formal definition of a patient's treatment history is as follows:

Definition 4.2 (Patient Treatment History): Assume that $\sum = \{e_1, e_2, \ldots, e_k\}$ is the set of examinations, $\Psi = \{p_1, p_2, \ldots, p_n\}$ is the set of patients and $\Gamma = \{\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m\}$ is a collection of treatment strategies. Each treatment history $T \subseteq \Gamma$ on patient $p_i \in \Psi$ is represented by a vector V with the number $\mathrm{N}(p_i)$ of elements. Each element $V^j_{p_i}$ of vector V reports a treatment $\mathrm{t_j}$ for patient $\mathrm{p_i}$ under the examination of $E_j = \{e_1, e_2, \ldots, e_x\} \subseteq \sum$. Thus, $T = V_{p_i} = [V^1_{p_i}, V^2_{p_i} \ldots, V^x_{p_i}]$.

Medical information is exchanged among different health information systems (HIS) in terms of a clinical document, or electronic health record (EHR), which is the combination of a medical record and patient treatment history. In this paper, a clinical document should be comprised of at least the following parts: (1) ***condition statement*** is the description of the patient's symptoms as clinical features; (2) ***solution*** is the disease diagnosis that is given by the doctor (or diagnosis system); (3) ***assessment*** indicates the accuracy of the diagnosis; and (4) ***treatment*** represents the treatment procedure.

To implement the semantic interoperability of clinical documents that are transferred among heterogeneous HIS, this paper designs a novel tabularized clinical document model, called the Clinical Tabular Document Model (CTDM). It is clear that the information model of a clinical document in the source context should be fully transformed to that of a target context without a semantic loss [34], [35]. In this paper, a semantic loss refers to missing or alternation of the term meaning and semantic relation when transferring documents across different contexts. For storage and communication, an information model must be expressed in terms of a logical data model (i.e., logical structure, e.g., relational, object-oriented or tree-based, in other words, implement-independent) and physical data model (e.g., HTML, XML, PHP or JSON, which is tool specific) [36]. A logical data model decides the organization of the elements for the data storage, while a physical data model focuses on the actual data transfer manner, storage format and presentation style. A logical data model and physical data model construct a syntactic document representation. For unambiguity at the semantic level of an overall information model, a conceptual model is introduced to implement the semantic document representation. The CTDM in this section, as a union of the syntactic document representation and semantic document representation, describes all of the characteristics of the clinical document exchanged among the HIS. Based on the CTDM, clinical documents (also as semantic documents [37]) are transferred from the document writer

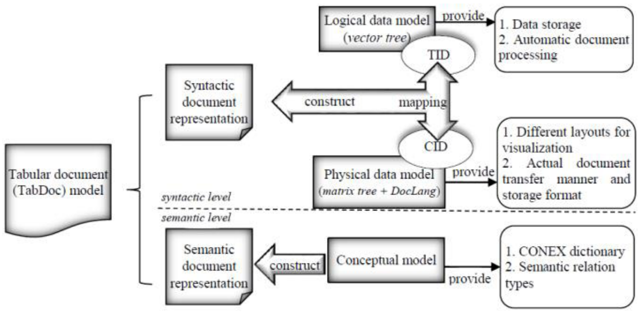

Logical data model
(vector tree)
provide
1. Data storage
2. Automatic document processing
TID
Syntactic document representation
construct
mapping
CID
1. Different layouts for visualization
2. Actual document transfer manner and storage format
Tabular document (TabDoc) model
Physical data model
(matrix tree + DocLang)
provide
syntactic level
semantic level
Semantic document representation
construct
Conceptual model
provide
1. CONEX dictionary
2. Semantic relation types

FIGURE 2. Syntactic and semantic document representation.

(Party A) to the document reader (Party B), and they are faithfully understood by the latter. The following sections show the syntactic representation and semantic representation of the CTDM.

A. SYNTACTIC CLINICAL DOCUMENT REPRESENTATION

Syntactic document representation (SDR) is a grammatical relationship among the terms in a document on how they are organized in a grammatical pattern [37]. This strategy can be applied to clinical documents. In this paper, we take the Vector Tree [38] (a tree structure) for the role of a logical data model. This choice is made because each clinical tabular document element (e.g., an empty placeholder or a single term) can be uniquely identified and hierarchically positioned, if its logical structure is a tree-based representation. A vector tree strengthens this point by assigning each tabular document element with a vector that represents its position from the root to its location in the tree, as shown in Fig. 3(4). For a physical data model, SDR contains a matrix tree [38] for presentation and a newly designed Clinical Tabular Document Language (DocLang, XML-based) for implementation. This arrangement is made because automatic analysis of an arbitrary document with a complex layout is an extremely difficult task. However, a critical component in facilitating the understanding of a document layout is its representation in memory [39]. Different research studies have represented document layouts in various ways [37], [40]. To achieve consistency of a layout, a clinical document is presented as a nested table, in other words, a nested matrix in mathematics. Nested matrixes construct a semantic document in another tree structure called a matrix tree. Thus, the vector tree, matrix tree and DocLang together constitute the syntactic document representation in this paper, as shown in Fig. 2. A vector tree can be alternatively constructed as a table by building its mapping to a matrix tree that shows positional relationships of cells in a table or embedded tables. This mapping is created by associating the position of each node in a vector tree with the position of each cell in a matrix tree, in such a way that any node in a vector tree has a unique corresponding position in a table. In a vector tree, the position is identified as the term identifier (TID), and in a matrix tree, it is identified by a cell identifier (CID) of a table.

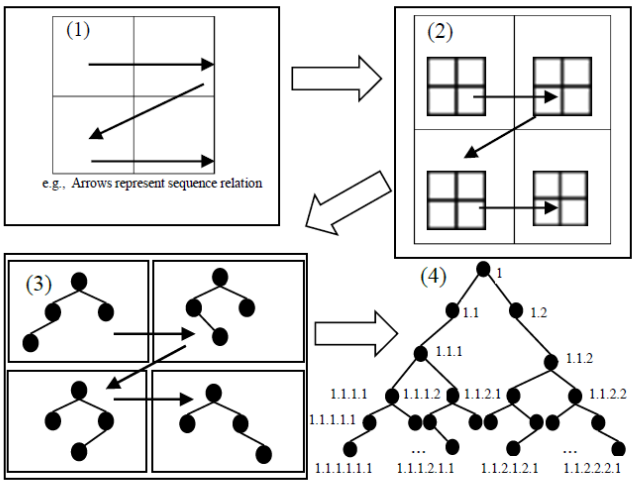
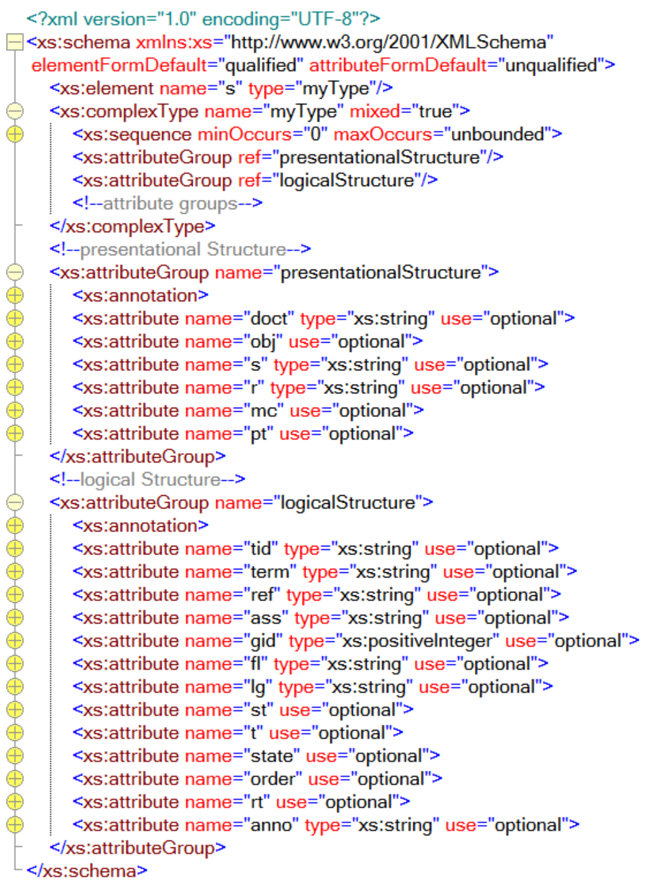


FIGURE 3. Procedures of constructing a clinical tabular document.

To facilitate clinical document exchange, DocLang integrates the vector tree for the logical structure and the matrix tree for presentations. Thus, DocLang guarantees syntactic consistency of the exchanged clinical documents in terms of a logical data model and physical data model among heterogeneous contexts. One advantage of DocLang is that clinical document designers do not need to consider the bottom implementation when designing a clinical document template. What layout they design in the front end will be automatically transformed to DocLang-based files at the back-end. When creating a table, the users first construct its basic table form, which is shown in Fig. 3(1). Second, we embed a basic sub-table form in each cell to create a complicated table if needed, and we iterate this step until reaching a desirable table format, as shown in Fig. 3(2). Because there are semantic relations between cells, embedded sub-table forms have semantic relations with one another. Because each type of basic sub-table forms maps to certain sub-tree structures (i.e., logical structure), semantic relations are also established among sub-trees (as shown in Fig. 3(3)), which constructs a large tree that represents the entire logical structure of the clinical tabular document (Fig. 3(4)).

At this stage, we use XML to implement DocLang. Fig. 4 shows the XML schema of DocLang, which is designed as follows: (1) There is only one category of element <s> (or <sign>) with groups of predefined denoters (i.e., attributes) as basic grammatical elements. An instance of the XML schema is a reusable document template that includes only abstract elements <s> without reification. (2) An empty element (or a placeholder) is reified by the attribute 'term' using a set of atomic terms that refer to concepts in a commonly known dictionary (e.g., CONEX [37], [41]) by attribute 'ref'. (3) The attribute 'anno' is a concept definition of one element, whose value is also a set of atomic terms in the dictionary. (4) The attribute 'r' denotes the position of an element in matrix tree (or layout) structure by matrix position [38], while 'rt' denotes which element(s) the current element



```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:element name="s" type="myType"/>
   <xs:complexType name="myType" mixed="true">
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:attributeGroup ref="presentationalStructure"/>
      <xs:attributeGroup ref="logicalStructure"/>
      <!--attribute groups-->
   </xs:complexType>
   <!--presentational Structure-->
   <xs:attributeGroup name="presentationalStructure">
      <xs:annotation>
      <xs:attribute name="doct" type="xs:string" use="optional">
      <xs:attribute name="obj" use="optional">
      <xs:attribute name="s" type="xs:string" use="optional">
      <xs:attribute name="r" type="xs:string" use="optional">
      <xs:attribute name="mc" use="optional">
      <xs:attribute name="pt" use="optional">
   </xs:attributeGroup>
   <!--logical Structure-->
   <xs:attributeGroup name="logicalStructure">
      <xs:annotation>
      <xs:attribute name="tid" type="xs:string" use="optional">
      <xs:attribute name="term" type="xs:string" use="optional">
      <xs:attribute name="ref" type="xs:string" use="optional">
      <xs:attribute name="ass" type="xs:string" use="optional">
      <xs:attribute name="gid" type="xs:positiveInteger" use="optional">
      <xs:attribute name="fl" type="xs:string" use="optional">
      <xs:attribute name="lg" type="xs:string" use="optional">
      <xs:attribute name="st" use="optional">
      <xs:attribute name="t" use="optional">
      <xs:attribute name="state" use="optional">
      <xs:attribute name="order" use="optional">
      <xs:attribute name="rt" use="optional">
      <xs:attribute name="anno" type="xs:string" use="optional">
   </xs:attributeGroup>
</xs:schema>
```

FIGURE 4. XML schema of DocLang.

has a relation with. (5) The attribute 'tid' clarifies the position of an element in a vector tree (or logical) structure using a vector [38]. (6) The attribute 'obj' specifies whether a vector tree can be mapped into a table, list or cell. (7) The attribute 'doct' indicates the scenario for certain document templates to be applied, or the behavior of the template, e.g., an inquiry or an offer. (8) An element has its semantic relation defined by the attribute 'st', and its cell value type is defined by another attribute, 't'. (9) The attributes 's', 'pt' and 'mc' are designed for claiming the style format, position of the value in a cell and whether the cell is merged, respectively. (10) The attribute 'ass' specifies an association in which the document elements can form a minimum semantic unit that is identified by the attribute 'gid' (group identifier). (11) Specifically, when associated document elements form a mathematical relation or a logical relation, the formula expression should be defined with the attributes 'fl' or 'lg', respectively. (12) The attribute 'state' represents the status of a checkbox, while 'order' defines a superiority for preference. The XML schema is well-formed and valid under XMLSpy 2013 sp1.

B. SEMANTIC CLINICAL DOCUMENT REPRESENTATION

Semantic document representation is a conceptual relationship among the terms in a document on how they are organized in a semantic pattern [37]. Similarly, this strategy can also be applied to clinical documents. In previous sections, information models for representing the semantics of a document require a conceptual model [35]. However, knowledge manifests itself in various relations among concepts or entities [42], [43]. Thus, the conceptual model in this paper includes the CONEX dictionary [37], [41] as a common dictionary and semantic relation types for illustrating relationships among terms. It guarantees that all of the clinical tabular documents that are created, communicated and used are semantically consistent and inferable without ambiguity by two semantic chains: the term chain and semantic relation chain. The term chain ensures semantic consistency at the vocabulary level, which is defined as the "reified concept$_1$ (riid$_1$) ⇔ local concept$_1$ (liid$_1$) ⇔ mapping concept (liid$_1$, ciid) ⇔ common concept (ciid) ⇔ mapping conceptb (ciid, liid$_2$) ⇔ local concept$_2$ (liid$_2$) ⇔ reified concept$_2$ (riid$_2$)". The notation ⇔ represents a mapping procedure. The term chain has already been discussed in previous papers [37], [44]. A semantic relation chain guarantees semantic consistency at the data type level with "one semantic relation type (for Party A) ⇔ one type of sub-tree structure (logical structure) ⇔ one type-of attribute-value pair in DocLang ⇔ the same semantic relation type (for Party B)".

This subsection exemplifies eight types of semantic relations that were imported from information science [45] for semantic representation of clinical documents. They are *part-of relation*, *reference relation*, *calculation relation*, *parallel relation*, *progressive relation*, *sequence relation, instance relation* and *choice relation*. Each type of semantic relation is logically represented by an abstract sub-tree structure that corresponds to one type-of attribute-value pair in DocLang (XML-based). Each type of sub-tree structure represents an independent semantic unit, which can be mapped onto several types of sub-table forms for presentation. The links between the semantic units also adopt different semantic relations. All the semantic units, with the links in between, constitute the entire semantics of the clinical tabular document. Another expression for semantics is rules. In this way, a clinical tabular document as a semantic unit can be represented by a set of rules (i.e., a rulebase). The rule expression of the semantics will be discussed in Section VI. The presentational structure is a tabularized result of the logical data structure. It represents one display layout type of its corresponding physical data model. Because we use a tabular form to present documents and the display layout varies, one logical data structure corresponds to one or many tabular formats for visualization. As a semantic relation's logical structure is represented by an abstract sub-tree structure, all of the sub-tree structures of the semantic relations in a table construct a logical structure of the whole tabular document. Indeed, the logical structure of a tabular document is not necessarily a tree; it can also be a network in certain cases. In other words, it behaves like a net overall by allowing associations among any nodes in the tree.

The purpose of each type of semantic relation is discussed as follows. (1) *Sequence relation* defines a processing order (e.g., inference in an inference engine) among objects (e.g., cells) in a clinical tabular document. For any objects O1 and O2 that form a sequence relation in a clinical tabular document, if and only if O1 is processed before O2, then we determine that there is a sequence relation between them. The sequence relation indicates a dependency in the process of semantic interpretation of a clinical tabular document. Its abstract representation in DocLang documents is: seq $=:$ $\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n}$ where the notation $=:$ means that the elements $s_i (i = 1, \ldots, n)$ at the right side satisfy the semantic relation at the left side. (2) *Part-of relation* defines a composition relation among the objects in a clinical tabular document. This relation expresses the formation of an object that consists of simpler components, for example, a book that has many chapters that can be further divided into sections and subsections. Its abstract representation is: partOf $=:$ $\mathrm{s}_1(\mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n})$. (3) *Instance relation* defines that one object is an instance of another. Instance objects automatically inherit semantic relations from their instantiated objects. Its abstract representation is ins $=:\mathrm{s}_i(\mathrm{v}_\mathrm{i})$. (4) *Reference relation* expresses that a fact is complete if and only if the referred objects are regarded as further explanation of the reference objects. Its abstract representation is: ref $=:$ $\mathrm{s}_1@(\mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n})$. (5) *Calculation relation* defines a mathematical formula or a logical operation for objects. If needed, an arbitrary number of cells in a table could generate a calculation relation (cal). Its abstract representation is the following: cal $=:$ $\mathrm{Math}_\mathrm{x}(\mathrm{s}_\mathrm{i}|\mathrm{Math}_\mathrm{y}(\mathrm{s}_\mathrm{a}, \mathrm{s}_\mathrm{b}), \mathrm{s}_\mathrm{j}|\mathrm{Math}_\mathrm{z}(\mathrm{s}_\mathrm{p}, \mathrm{s}_\mathrm{q})^*)$. (6) *Choice relation* defines the availability status of selected items. Checkboxes are often used in clinical tabular documents to indicate a choice relation among items. Its abstract representation is: choice $=:$ $\{\mathrm{s}_\mathrm{j}|\mathrm{s}_\mathrm{j}|\ldots|\mathrm{s}_\mathrm{n}\}$. (7) *Progressive relation* defines hierarchical priorities among objects in ascending (ASC) or descending (DESC) order, while (8)*Parallel relation* defines a same priority among objects. Their respective abstract representations are: prog $=:$ $\mathrm{s}_1 << \mathrm{s}_2 << \ldots << \mathrm{s}_\mathrm{n}$ and para $=:$ $\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{n}$.

V. KNOWLEDGE EXTRACTION FOR RULES

The source of the rules in a knowledge base for inference can be divided into two categories: (1) decision tree models generated by classification algorithms; and (2) clinical experts (e.g., doctors) through a rule editor. This section mainly discusses the techniques of knowledge extraction for rules.

A. MACHINE LEARNING ALGORITHMS

For rules that are generated by classification algorithms, this paper proposes a voted ensemble multi-classification model that simultaneously integrates multiple classifier algorithms and votes for the best performance as the classification results. In this paper, we use state-of-art decision-tree-based classification algorithms that create rules in the form of a decision list. Each rule is followed by statistical output (e.g., accuracy rate) that is convenient for users (e.g., doctors) to consider.

To fulfill the reliability of classification, decision trees should be updated continuously. Instead of re-constructing a classification model each time when new instances arrive, the classification algorithms support incremental learning and, thus, reduce the cost in terms of the time and error rate. This section first introduces popular machine learning algorithms, including Hoeffding Tree, Hoeffding Adaptive Tree, Hoeffding Option Tree and iOVFDT, and then, it presents a new classification algorithm to perform the classification task.

1) HOEFFDING TREE

Hoeffding tree (HT) is an incremental learning algorithm [46]. It constructs a decision tree by using data streams as training datasets, provided that the statistical distribution of the instances always stays the same [23], [46]. The Hoeffding tree does not store any instance in the main memory, and thus, it requires only the space that is proportional to the size of the tree and the associated sufficient statistics. It updates the tree model whenever a new piece of data arrives [23].

HT guarantees that an optimal splitting attribute can be acquired based on a partial dataset rather than on a complete dataset. An obvious merit of HT is that its output is asymptotically nearly identical to that of a non-incremental learner using infinite many instances. This idea is mathematically supported by the Hoeffding bound (HB) [46], which selects attributes as test nodes by using a training dataset that is as small as possible, which produces the same result as the result that would be chosen using an infinite dataset with high confidence (e.g., the confidence in Weka is 1-1.0E-7). In fact, HB [47] states that with a probability of 1 $-\delta$, a random variable in the range R will not differ from the estimated mean after n observations by more than

$$HB = \sqrt{(1/2n)R^2 \ln(1/\delta)} \tag{1}$$

where R is the class distribution, and n is the number of instances that fall into a leaf. Note that the Hoeffding bound is a monotonically decreasing function of instance number n. Assume that X_i is the attribute that has the highest evaluation value E_i (e.g., the information gain or Gini Index), and X_j is the attribute that has the second highest evaluation value E_j. When $E_i - E_j > HB$, X_i is chosen as the best splitting attribute at the current node with a confidence of $(1 - \delta)$.

2) HOEFFDING ADAPTIVE TREE AND HOEFFDING OPTION TREE

The Hoeffding Adaptive Tree (HAT) uses the ADWIN algorithm [20] to monitor the performance of the branches during the period of tree construction. In HAT, when the accuracies of the old branches are lower than those of the new branches, they are replaced. This approach places an adaptive window of instances at each node, which raises an alert whenever a change in the attribute-class statistics is detected at the node [48]. The essence of the ADWIN algorithm is the following: whenever two "large enough" sub-windows of a sliding window W display "distinct enough" averages, it is confident that their corresponding expected values are different, and the older portion of the window must be discarded [20]. The Hoeffding bound is eventually updated as follows:

$$m = \frac{2}{1/|W_0| + 1/|W_1|} \tag{2}$$

$$HB = \sqrt{(1/2m) \cdot \ln(4\,|W|\,/\delta)} \tag{3}$$

where m is the harmonic mean of the length of subwindows $|W_0|$ and $|W_1|$, and δ is a confidence bound that indicates how confident we want to be in the algorithm's output.

Hoeffding Option Trees (HOT) are normal Hoeffding trees, which contain option nodes that apply different tests [49]. An instance travels down multiple paths of the decision tree and arrives at multiple leaves [50]. HOT is capable of simultaneously representing multiple trees in a single structure [25]. It introduces another parameter, $\delta^{'}$, for deciding when to add another split option beneath a node that has already been split [50]. A new option can be added if the best unused attribute looks better than the current best existing option according to the $\bar{G}$ criterion and a Hoeffding bound with confidence $\delta^{'}$ [50]. $\delta^{'}$ can be expressed in terms of a multiplication factor α, which specifies a fraction of the original Hoeffding bound:

$$\delta^{'} = e^{\alpha^2 \ln \delta} \tag{4}$$

$$HB = \sqrt{(1/2n) \cdot R^2 \cdot \ln(1/\delta^{'})} \tag{5}$$

3) iOVFDT

iOVFDT [51] optimizes the process of tree construction via functional tree leaf and incremental optimization to obtain a tradeoff between the accuracy and tree size. iOVFDT utilizes a weighted Naïve Bayes classifier to reduce the effect of having an imbalanced class distribution, where the classifier on the leaf can further enhance the prediction accuracy via the embedded classifier. It chooses the class that has the maximum possibility computed by the weighted Naïve Bayes [51] as the predictive class in a leaf:

$$p_{ijk} = \omega_{ijk} \frac{P(x_{ij}|y_k) \cdot P(y_k)}{P(x_{ij})} \quad where\ \omega_{ijk} = \frac{n_{ijk}}{\sum_{k=1}^{K} n_{ijk}} \tag{6}$$

where x_{ij} is the jth value of attribute X_i, and y_k is the kth class value. Here, n_{ijk} is the sufficient statistic that reflects the number of instances that have attribute X_i equal to x_{ij} and class value equal to y_k.

4) VOTED ENSEMBLE MULTI-CLASSIFICATION ALGORITHM

Ensemble methods merge several models, whose individual predictions are combined in a certain manner (e.g., averaging). The output of this method is a final prediction that has better accuracy and convenience to scale and parallelize compared with single classifier methods [25]. Among the current ensemble methods, component classifiers are merged into an ensemble that has a fixed size, and models are built from relatively small subsets of the data. Once the ensemble is full, new classifiers are added only if they satisfy some quality

Algorithm 1 Voted Ensemble Multi-Classification Algorithm

```
Input: data stream, different classifiers (e.g., Hoeffding Tree)
Output: a decision tree
Set classifier model T = null;
while more data points are available
  read $|X_i|$ instances between samples of the learning performance; // $|X_i|$ is the sample frequency
  for each classifier $C_j$ in ensemble $E$
    if ($T_{i-1} \in T \neq null$) //$T_{i-1}$ is a temporary classifier model in the last iteration.
        update temporary classifier model $T_{i-1}$ using $C_j$ on $X_i$ and acquire $T_i$; //update not re-construct model
    elseif (temporary model $T_{i-1} = null$)
      build a new model $T_i$ using $C_j$ on $X_i$;
    if (Quality($C_j$) > Quality($C_{j-1}$)) //modify votes
      $T_i = T_{ij}$; //$T_{ij}$ is created by the classifier $C_j$ with the best quality (e.g., accuracy or Kappa statistic in $E$)
  end_for
  save temporary classifier model $T_i$;
end_while
```

criterion (e.g., accuracy), which is based on their estimated ability to improve the ensemble's performance [52]. Due to the fixed size of the ensemble, one of the existing classifiers must be removed. Then, a tree is iteratively built upon upcoming data. Performance estimations are conducted by testing the new tree (and the existing ensemble) on the next chunk of data points [52].

In our voted ensemble multi-classification (VEMC) algorithm, as shown in Algorithm 1, decision trees are constructed by different classifiers that are designed for data stream mining. Initially, each classifier receives equally weighted votes. With upcoming instances, based on the error rate or prediction accuracy, the weighted vote for each classifier is adjusted dynamically based on the computing quality of the corresponding classifier. The basic idea of the VEMC algorithm is based on two assumptions: (1) the classification quality of an ensemble classifier is better than an ad hoc classifier, and (2) with successive upcoming instances, the performance of an ensemble classifier improves.

The two assumptions above can be demonstrated by a simple case. We use HT algorithms with different numbers of instances that a leaf should observe (i.e., grace period) between split attempts as different classifiers. Assume that HT_1 and HT_2 with 200 and 100 grace periods, respectively, are applied on two cancer datasets [53] with 34200 and 342000 instances each. From Table 1, in the first dataset (34200 instances), HT_2 is better than HT_1 in terms of accuracy and stability, but its performance becomes worse in the second dataset (342000 instances). Thus, it is advisable to use an ensemble classifier because the performance of different classifiers could be changed with different datasets.

TABLE 1. A simple case.

Classifier	Total Number of Instances	Correctly Classified Instances (accuracy)	Kappa statistic (stability)	Root mean squared error
HT_1	34200	73.0965 %	0.64	0.2869
HT_2	34200	77.8947 %	0.703	0.2554
HT_1	342000	87.7655 %	0.8341	0.1705
HT_2	342000	87.6778 %	0.8332	0.17

Moreover, the classifier quality is positively correlated with the number of instances.

B. EXPERT EXPERIENCE

In addition to the rules that are generated from the decision tree, it is also important to model the knowledge that medical experts use for clinical diagnosis [54]. This approach is important because of some specific cases, such that cases in which several diseases that are inferred by the rules in a decision list could have probabilities that are not significantly different from one another (i.e., $p \geq 0.05$). Medical experts are encouraged to enter new rules or edit existing rules to provide a solid decision. Rules that are directly provided by medical experts are given different priorities based on what stage in the diagnosis and treatment protocol the newly created/modified rule pertains to. This paper designs a rule editor to facilitate medical experts to enter rules.

VI. SEMANTIC INFERENCE

The whole procedure of semantic inference on clinical documents in the inference engine can be segmented into two stages. The first stage focuses on disease detection, which relies on knowledge that is learned from clinical document content and the rules transformed from decision trees as inputs to infer the type of disease. The second stage makes an inference on disease treatment, using the diagnosis output of the first stage and then comparing between the clinical document and historical medical records with similar symptoms. To implement these two stages, a semantic inference algorithm with defeasible logic as an inference strategy is proposed in this section.

In the following sections, we first discuss the inference strategy that is used in our semantic inference scheme, and then, we discuss how to translate different types of semantic relations in our clinical tabular document into rules. Finally, the semantic inference algorithm that is used in the inference engine is given.

A. INFERENCE STRATEGY

1) DEFEASIBLE LOGIC

In the semantic inference scheme, defeasible logic is applied for rule reasoning, which handles both strict and defeasible rules, especially the priority (e.g., a progressive relation in a document). It is a simple rule-based approach to reasoning with incomplete and inconsistent information.

TABLE 2. **Mapping strategy between SR and RS (partial).**

Semantic relation (SR) type	Abstract representation (SR)	Rule syntax (RS)
instance relation	instance $=: s_1(v_1)$	$<$ FACT $>$ $s_1 = v_1$
		$<$ QUERY $>$ { IF ?v1 THEN {Goto Database; Read ?v1; Where s1;}}
progressive relation	progressive $=: s_1 << s_2 << \cdots << s_n$	$<$ Superiority statement $>$ $\{s_n, \ldots, s_2, s_1\}$
sequence relation	seq $=: s_1, s_2, \ldots, s_n$	$<$ Superiority statement $>$ $\{s_1, s_2, \ldots, s_n\}$
reference relation	reference $=: s_1@(s_2, \ldots, s_n)$	$<$ FACT 1 $>$ $s_1(v_1, s_2)$ … $<$ FACT n $>$ $s_1(v_{n-1}, s_n)$
part-of	partOf $=: s_1(s_2, \ldots, s_n)$	$<$ FACT 1 $>$ $s_2(v_2, s_1)$
relation		… $<$ FACT n $>$ $s_n(v_n, s_1)$
calculation relation	calculation $=: Math_x(s_i \mid Math_y(s_a, s_b), s_j \mid Math_z(s_p, s_q)^*)$	<Rule> IF s_i != null and s_j != null THEN $Math_x(s_i \mid Math_y(s_a, s_b), s_j \mid Math_z(s_p, s_q)^*)$
choice relation	choice $=: \{ s_i \mid s_j \mid \ldots \mid s_n\}$	$<$ Rule i $>$ IF s_i.state = on THEN s_i.rt.v = s_i.v … $<$ Rule n $>$ IF s_n.state = on THEN s_n.rt.v = s_n.v

Based on different structures of rules, the Tabdoc rule syntax [55] is used to define the rules as different technical components, such as (i) *normal rule*: each normal rule has a *head* and a *body*. The *head* part proceeds to the keyword *THEN*, whereas the *body* part follows the keyword *IF* and precedes the *THEN* part. (ii) *fact*: a fact is a special rule that has no body part. When defining a fact, only the head part is given. (iii) *semantic relation*: different semantic relations stand for different functional relationships between the constituents in a document. Before performing semantic inference on a clinical tabular document, its semantic relations should be represented by different types of rules (e.g., facts, queries or normal rules). This mapping from semantic relations to rules is explicitly discussed in Table 2. (iv) *function*: functions used in the head of a Tabdoc rule are called responders, and they usually implement functionalities such as assignment or query. Functions used in the body of a Tabdoc rule are called testers, and they judge whether conditions are true or false. (v) *queries and answers*: queries are special rules without heads. When defining queries, only the body part of a Tabdoc rule is given. The answer to a query is often modeled as a set of facts. (vi) *processes*: a process is a conditional sequence of activities (or operations) and is heterogeneous in different parties [10]. Different parties are likely to design heterogeneous processes that might be applicable and adaptable for specific contexts. Activities can be divided into three categories: private, community and public. Private activities are internally defined operations of a party and cannot be understood by the outside world. Community activities are mutually understandable within a group of parties. Public activities are public operations that are understandable by all parties. In this paper, a clinical document template corresponds to an activity that is defined in the design phase of the document template. By utilizing the ConexNet concept representation [56], users can collaboratively create new concepts in vocabularies for activity definition when creating processes. For example, a process could be similar to the following: *IF* $activity_1(doc_1)$ *THEN* $activity_2(doc_2)$, where an activity contains a clinical document as its real parameter.

Having received a clinical tabular document (CTD), it is required to process the document and extract a series of components as the input to an inference engine. Table 2 partially summarizes the mapping strategy between each type of semantic relation (SR) in CTD and the rules used in the inference engine.

2) FUNCTIONALITY OF RULES

In an inference engine, the priority in the execution order of the rules is determined by their corresponding functionalities. In this paper, rules have the following functionalities: (i) basic conditions (R_c) transformed from a new medical record; (ii) decision rules (R_{dr}) to generate the temporary result set TRS; (iii) expert rules (R_{epr}) for the temporary result set update and decision; (iv) rules for a temporary treatment strategy (R_{ts}); and (v) rules for a treatment strategy (R_{emr}) update. The priority of rules with different functionalities is defined as $\{R_c > R_{dr} > R_{epr} > R_{ts} > R_{emr}\}$.

B. SEMANTIC INFERENCE ALGORITHM (SIA)

Based on the inference strategy in Section 6.1, this section describes a Semantic Inference Algorithm (SIA) that is used in the inference engine in the clinical diagnosis and treatment system (CDTS). The goal of the algorithm is to implement a diagnosis and determine a treatment strategy with an input clinical tabular document and the corresponding rules (e.g., decision rules in a decision list).

1) PRECONDITION OF SIA

SIA has preconditions as follows.

R_c : a set of rules about patient basic conditions (e.g., patient symptoms) that are transformed from a clinical tabular document (as shown in Table 2) in a logical format (e.g., facts, queries).

DS (Data source): historical medical records of patients, in the form of .arff or.csv.

R_{dr}*:* decision rules transformed from the decision tree generated by the voted ensemble multi-classification algorithm.

R_{epr}*:* expert rules created by medical experts for decision making.

R_{ts}/R_{emr}*:* a set of rules to query/update a treatment strategy.

Algorithm 2 SIA

```
1 BEGIN
    2 $P_{cd}$ = Transform(*clinical document*) THEN /* transform received clinical document into a medical record in a compatible form with DS */
    3 $\Omega \leftarrow$ Cluster(DS $\Theta$ $P_{cd}$); /* cluster a new medical record into a group with similar historical medical records and form a new dataset $\Omega$*/
    4 TRS $\leftarrow$ Apply(r:$R_c \Theta$ r:$R_{dr}$); /* Apply decision rules ($R_{dr}$) on basic conditions ($R_c$) to get temporary result set (TRS) */
    5 IF |TRS| = 0
    6 THEN{ *send error message;*}
    7 ELSE {
        8 IF |TRS| = 1 /* *|TRS| = 1 means the number of final result is only one.* */
        9 THEN { $\Delta$TRS $\leftarrow$ TRS; }
        10 ELSE{
        11 IF ($R_{epr}$!=NULL)
        12 $\Delta$TRS $\leftarrow$ Apply(TRS $\Theta$ r: $R_{epr}$); /**Apply expert rules ($R_{epr}$) on TRS to update |TRS|;*/}
      }
13 FINAL { FRS $\leftarrow$ Query($\Omega\Theta$ $R_{ts}$); /**Apply treatment strategy rules $R_{ts}$on $\Omega$based on $\Delta$TRS and get disease treatment set (FRS);* */
        14 IF ($R_{emr}$!=NULL)
        15 $\Delta$FRS $\leftarrow$ FRS $\Theta$ r:$R_{emr}$; /**Use emergent rules $R_{emr}$to update the treatment strategy in FRS;* */}
16 END
```

2) POST-CONDITION OF SIA

The post-condition of SIA is the inferred results (e.g., disease diagnosis and treatment strategy).

3) SIA ALGORITHM

Based on the inference algorithm SIA, as shown in Algorithm 2, the whole inference procedure is mainly divided into three steps, as follows.

Step 1 (line 1-9): This step first uses a clustering technique to form a group Ω where the medical record of a new patient is clustered into the group together with similar historical medical records in terms of patient symptoms. This paper takes the COBWEB method as the clustering algorithm, which is based on a category utility (CU) function that measures the clustering quality. The definition of CU is

$$CU(C_1, C_2, \ldots, C_k) = \frac{\sum_l \Pr[C_l] \sum_i \sum_j (\Pr[a_i = v_{ij}|C_l]^2 - \Pr[a_i = v_{ij}]^2)}{k} \quad (7)$$

where $C_1, C_2, \ldots, C_k$ are the k clusters; the outer clusters summation is over these clusters; the next inner one sums over the attributes; and a_i is the i^th attribute, and it takes on the values $v_{i1}, v_{i2}, \ldots$, which are addressed by the sum over j.

TABLE 3. Groups of manifestations in lung cancer (lc).

Manifestations	Description
lc_topog	ICD-O-3 topography of lung cancer.
lc_loclhil	lc_loclhil = 1 means that the primary tumor is located in the left hilum.
lc_locllow	lc_locllow = 1 means that the primary tumor is located in the left lower lobe.
lc_loclup	lc_loclup = 1 means that the primary tumor is located in the left upper lobe.
lc_locmed	lc_locmed = 1 means that the primary tumor is located in the mediastinum.
lc_loclmsb	lc_loclmsb = 1 means that the primary tumor is located in the left main stem bronchus.
lc_loccar	lc_loccar = 1 means that the primary tumor is located in the carina.
lc_loclin	lc_loclin = 1 means that the primary tumor is located in the lingual.
lc_grade	ICD-O-3 grade of lung cancer.

Note that the probabilities are obtained by summing over all of the instances.

Then, the decision rules (R_{dr}) are created by our classification algorithm (see Section 5.1.4) and are applied to basic conditions (R_c) to infer possible diagnoses, generating a temporary result set TRS (e.g., TRS = {diabetes mellitus, breast mass}). If the number of inferred diseases in TRS is zero, then it sends an error message. If the number of components in TRS is one, then it takes the TRS as the final diagnosis (ΔTRS).

Step 2 (Line 10-12): This step is to exclude the unlikely inferences by using expert rules (R_{epr}), namely, deducing the number of components in TRS and finally outputting the updated result set (ΔTRS) as the diagnosis. This step can be ignored if R_{epr} is empty.

Step 3 (Line 13-16): The purpose of this step is to find suitable previous patient treatment records for a new patient. A query is executed on historical patient records Ω about the same disease in ΔTRS by using the rules R_{ts}, and then, it stores the treatment strategy of the best matching record in the disease treatment set (FRS). If there is a set of emergent rules (R_{emr}) that are designed by clinicians, it then uses R_{emr} to dynamically update FRS and thereby output the new treatment strategy ΔFRS. In this step, we must compute the similarity degree among the different medical records. Specifically, the similarity between the medical records is evaluated based on the symptoms. Table 3 shows some of the symptoms (i.e., manifestations) of lung cancer. Due to the complexity of the pathology, the symptoms of each patient can be regarded as a combination of specific manifestations.

Mathematically, the calculation of a similarity degree between the symptoms in two medical records is essentially the comparison of two arrays. We can use a matrix to represent the comparison result. For simplicity, we assume that different symptoms are independent of one another. Thus, when all of the symptoms are represented by symbols such as 'a, b, ..., z, aa, bb, ...' and sorted in alphabetical order, only

TABLE 4. Comparison of symptoms between two medical records.

		a new medical record					
		a	b	c	d	e	…
a historical medical record	a	S_a					
	b		S_b				
	c			S_c			
	d				S_d		
	e					S_e	
	…						…

the principal diagonal elements in the matrix are likely to be non-zero. Table 4 shows an example of a comparison between two medical records.

The similarity computation (*Sim or S*) between two medical records is based on the similarity measurement (S_{att}) for each pair of attributes ($att \in Attributes$). The following formula shows one example of a similarity calculation between two medical records.

$$\text{Sim}(\text{src}, \text{trg}) = \frac{1}{n} \sum\nolimits_{att \in Attributes} (W_a * S_{att}(src, trg)) \tag{8}$$

where *src* (source) is a historical medical record, and *trg* (target) is a new patient record. *Sim(src, trg)* represents the similarity degree for the symptoms in the two records (*src* and *trg*). Here, 'n' is the number of attributes (i.e., symptoms) in these two records. S_{att} represents the similarity degree for a certain attribute in the two records.W_a is the weight of attribute a, which represents the correlation degree between attribute a and the class attribute (i.e., the disease type).

$$W_a = GainRatio = \frac{InfoGain(a)}{\sum_{a \in Attributes} InfoGain(a)} \tag{9}$$

The higher the value of W_a, the more correlated that attribute a is with the class. Thus, it is necessary to consider the weight of the attribute when judging the similarity degree of two medical records. In a decision tree, an attribute that has a high correlation degree with the class attribute is close to the root node. *InfoGain* (a) is the information gain of attribute a.

$$InfoGain(a) = I(S_1, S_2, \ldots, S_m) - E(a) \tag{10}$$

where $I(S_1, S_2, \ldots, S_m)$ is the information amount that is required to split a dataset.

$$I(S_1, S_2, \ldots, S_m) = -\sum\nolimits_{i=1}^{m} p_i \log_2 p_i \tag{11}$$

where S_i is the number of samples in classes C_i ($i = 1, 2, \ldots, m$), and p_i means the probability of the instances in $S = \{S_1, S_1, \ldots, S_m\}$ that belong to class C_i. $E(a)$ is the information amount for splitting a sub-dataset by attribute a.

$$E(a) = \sum\nolimits_{j=1}^{v} \frac{(s_{1j} + s_{2j} + \ldots + s_{mj})}{s} \times I(s_{1j}, s_{2j}, \ldots, s_{mj}) \tag{12}$$

where s_{ij} is the set of instances whose class values are C_i in the subset of $\{s_j | a = a_j, j \in 1, 2, \ldots, v, s_j \in S\}$. $I(s_{1j}, s_{2j}, \ldots, s_{mj})$ means the average information amount that is required to identify the class labels for all of the instances in s_j.

$$I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum\nolimits_{i=1}^{m} p_{ij} \log_2(p_{ij}) \tag{13}$$

where p_{ij} is the probability that the instances in s_j belong to the class value C_i. The value of S_{att} depends on the data type of attribute *att*. If attribute *att* is a numeric attribute, then

$$S_{att}(x, y) = \frac{\min\{x_{att}, y_{att}\}}{B_a} \tag{14}$$

where B_a is the breadth of the range of attribute *att*, and x_{att}, y_{att}, B_a are all mapped into the interval [0, 1]. This approach is applicable because the range of every numerical attribute in this application is bounded. If attribute *att* is a nominal attribute, then

$$S_{att}(x, y) = \begin{cases} 1, & \text{iff } x = y \\ 0, & \text{iff } x \neq y \end{cases} \tag{15}$$

where x, y ∈ {true/*false*, yes/*no*, and so on}. Table 4 can be abstracted as a matrix called a similarity measurement matrix, as shown below.

$$\partial = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} S_a & 0 & 0 & 0 & 0 \\ 0 & S_b & 0 & 0 & 0 \\ 0 & 0 & S_c & 0 & 0 \\ 0 & 0 & 0 & S_d & 0 \\ 0 & 0 & 0 & 0 & S_e \end{bmatrix} \tag{16}$$

Similarity measurement matrix ∂ is a diagonal matrix, while the range of the value of the diagonal elements is in [0, 1] and the values of the non-diagonal elements are zero. 1 means that the values of the two medical records regarding the same attribute are equal, while 0 means that they are unequal. We take the historical medical record whose matrix ∂ has the maximum rank as the best matching case for the new medical record in the treatment inference phase. The rank is the number of non-zero rows in the reduced row echelon form of the matrix. The rank could vary, because different medical records could have different values on the same symptoms. Specifically, if any one value out of two medical records regarding the same symptom (e.g., attribute a) is null, then the diagonal element becomes 0. The diagonal element becomes 1 if the attribute a is nominal and the two values on a are the same. If attribute a is numerical, then a probability will be given to describe the similarity between the records.

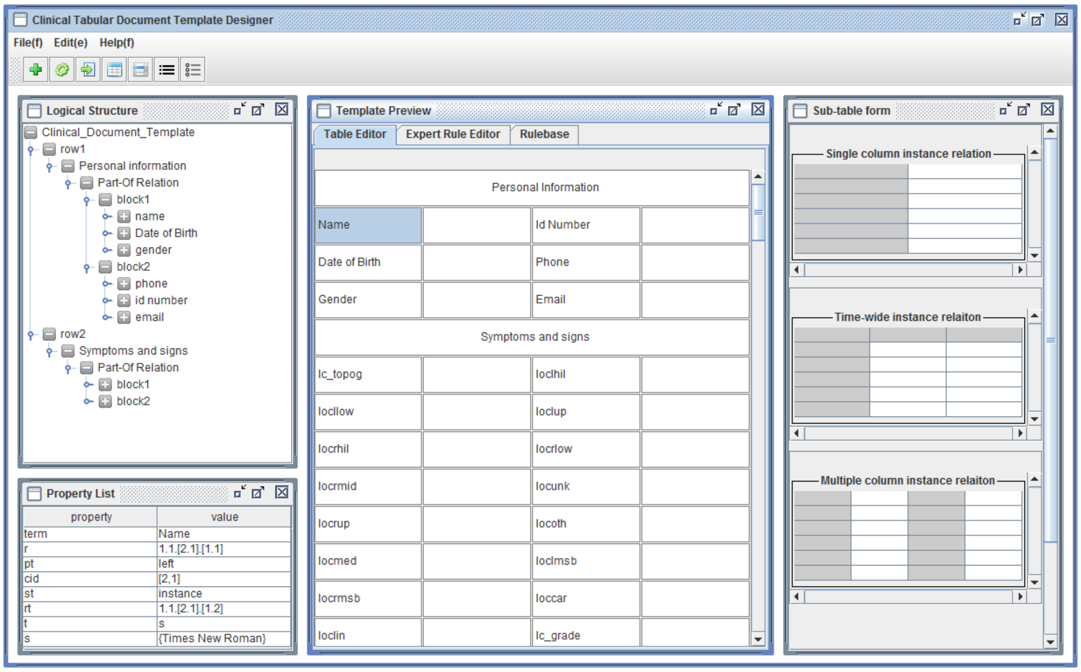
VII. EXPERIMENTS

Our aim is the evaluation of CDTS for disease detection and treatment suggestions with machine learning algorithms (e.g., clustering and classification) and an inference engine.

A. EXPERIMENTAL DATA AND EVALUATION MATRICES

1) DATA SOURCE

The dataset has been crawled from the website of the Cancer Data Access System (CDAS, https://biometry.nci.nih.gov/cdas/), which records data from the NLST (National Lung Screening Trial), PLCO (Prostate, Lung, Colorectal, and


Clinical Tabular Document Template Designer
File(f) Edit(e) Help(f)
Logical Structure
Template Preview
Table Editor
Expert Rule Editor
Rulebase
Sub-table form
Single column instance relation
Time-wide instance relaiton
Multiple column instance relaiton
Property List

```
Clinical_Document_Template
  row1
    Personal information
      Part-Of Relation
        block1
          name
          Date of Birth
          gender
        block2
          phone
          id number
          email
  row2
    Symptoms and signs
      Part-Of Relation
        block1
        block2
```

property	value
term	Name
r	1.1.[2.1].[1.1]
pt	left
cid	[2,1]
st	instance
rt	1.1.[2.1].[1.2]
t	s
s	{Times New Roman}

Personal Information			
Name		Id Number	
Date of Birth		Phone	
Gender		Email	
Symptoms and signs			
lc_topog		locIhil	
locllow		loclup	
locrhil		locrlow	
locrmid		locunk	
locrup		locoth	
locmed		loclmsb	
locrmsb		loccar	
loclin		lc_grade	

FIGURE 5. Snapshot for GUI-based clinical tabular document template designer.

Ovarian Cancer Screening Trial), and IDATA (Interactive Diet and Activity Tracking in AARP study) cancer studies. It includes 342 outpatient records with 31 attributes by integrating several datasets, such as Lung Cancer, Spiral CT Comparison Read Abnormalities and Diagnostic Procedures, through the patient's ID as a key. The attributes of each medical record include the lung cancer grade, cancer position, treatment type, and abnormality type.

2) EVALUATION MATRICES

This paper uses the accuracy and kappa statistic as the evaluation matrices that are most commonly used in machine learning algorithm evaluation. The accuracy (Acc) is defined as the number of true positives (T_P) over the number of true positives plus the number of false positives (F_P), as defined in formula (17).

$$Acc = \frac{T_p}{T_p + F_p} \tag{17}$$

The kappa statistic (κ) measures the agreement of a prediction with the true class, where 1.0 means complete agreement, as defined in formula (18).

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{18}$$

where p_o is the relative observed agreement among the raters, and p_e is the hypothetical probability of a chance agreement.

B. EXPERIMENTAL PROCEDURE

The experimental procedure includes the following steps:

Step 1 (Data Preprocessing): This step focuses on two types of data transformation. One type is to transform historical medical records (HMR) to historical patient instances in .arff or .csv format. The other type is to transform a new DocLang-based clinical tabular document into a patient instance in .arff or .csv format, where the language terms that are present in the template become attributes whose corresponding values are textual terms that are input from users (e.g., clinicians). Because the clinical document is based on the clinical tabular document model (CTDM), which guarantees semantic consistency through the term chain and semantic relation chain, the transformation of clinical documents among heterogeneous contexts does not cause semantic loss.

Fig. 5 shows the GUI of the clinical tabular document template designer (CTDTD). CTDTD generates customized clinical document templates that enable clinicians to design a variety of clinical tabular documents according to their particular needs. For example, a template that allows patients to enter their symptoms about lung cancer can be made, in such a way that new medical records can be created for online consultation. In this case, a reified clinical document corresponds to a record in a dataset (e.g., a row in an .arff or .csv file). Standard windows data entry elements (e.g., check boxes, radio buttons, input boxes and combo boxes) can also be inserted into the templates. To further facilitate the data input, a semantic input method (SIM) [37] is also integrated in CTDTD. Any term that is input through SIM is referred from the CONEX dictionary [37], [41] by selecting the exact meaning from a drop-down list. Each concept in the CONEX dictionary has a unique identifier (*iid*) that points to the same meaning regardless of the context. The information to

TABLE 5. Examples of patient records in dataset Ω.

Id	lc_topog	lc_locllow	lc_loclup	...	lc_loclmsb	lc_grade
100012	C34.1	0	0	...	0	2
100158	C34.1	0	0	...	0	3
100280	C34.3	1	0	...	0	1
100658	C34.3	1	0	...	0	9
104114	C34.8	0	1	...	1	4

exchange is the logical combination (i.e., the logical structure in the CTDM model) of *iid(s)* of different concepts.

CTDTD consists of several frames. The frame of the Subtable form lists the most-often used basic sub-table forms. The Template Preview frame is an operation platform for designing the layout structure of the clinical tabular documents. The users drag the required sub-table forms to the console in the Template Preview frame or manually create the sub-table forms by merging or splitting the table cells. Each type of sub-table form has to be mapped to at least a certain type of semantic relation managed by the attribute '*st*' in the *Property List frame*, which is identified by a group identifier GID. The semantic relations of manually created sub-table forms should be set in the Property List frame by the users themselves. In addition, clicking on any cell in the Template Preview frame will pop up the Property List frame that shows all of its denoters. All of the sub-table forms in the Template Preview frame construct a complete clinical tabular document template. The Logical Structure frame shows the logical data model in a tree-based pattern.

Step 2 (Clustering): Compare the new patient instance with historical ones to acquire a set of the same or similar instances and form a group. Two *assumptions* are defined for simplicity of discussion. The first is that patients with the same or similar attributes (i.e., physical examination parameters) and corresponding values are very likely to have the same disease. The second is that different hospitals have the same examination protocol on symptoms of the same disease. The purpose of these two assumptions is to ensure that all of the patients with the same class of diseases are clustered into the same group as much as possible. The group that the new patient instance falls into is taken as a new dataset.

Step 3 (Classification): Different diseases can share similar symptoms (e.g., fever, white blood cell counting increase, and so on). Thus, patients who have the same or similar symptoms could have different diseases. According to the new dataset Ω (i.e., the group) acquired in step 2, a decision tree is constructed via running the voted ensemble multi-classification algorithm on Ω.

For example, if the new patient being tested has lung cancer, the new dataset Ω as input for classification includes 342 outpatient records of the data source (see Section VII-A1). Each record in Ω contains 31 attributes with the attribute "lc_grade" (see Table 3) as a class. The class attribute has five categories which indicate the severity of lung cancer. Table 5 exemplifies five historical patient records in the dataset Ω.

TABLE 6. Decision tree vs. decision rules.

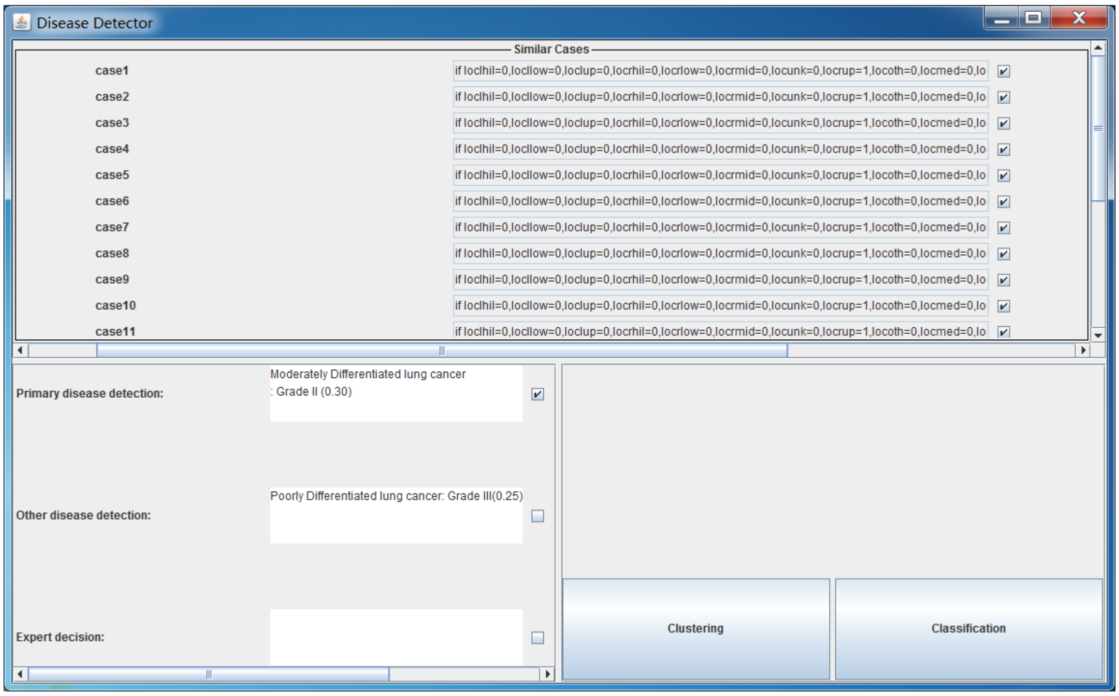
Decision tree	Decision rules

if [att 16:locunk] <= 0.09090909090909091:
if [att 6:visible_days] <= -147.27272727272725:
if [att 7:sct_ab_code2] <= 5.7272727272727275:
Leaf [class:lc_grade] = <class 2:2> weights: {0 | 4,204 | 0 | 0 | 0}
if [att 7:sct_ab_code2] > 5.7272727272727275:
Leaf [class:lc_grade] = <class 3:3> weights: {1,252 | 0 | 2,027 | 0 | 0}
if [att 6:visible_days] > -147.27272727272725:
if [att 13:locrhil] <= 0.09090909090909091:
if [att 20:loclmsb] <= 0.09090909090909091:
if [att 10:loclhil] <= 0.09090909090909091:
Leaf [class:lc_grade] = <class 2:2> weights: {36,543 | 96,355 | 78,881 | 10,490 | 8 8,028.404}
	(1) if locunk <= 0.09090909090909091 and visible_days <= -147.27272727272725 and sct_ab_code2 <= 5.7272727272727275 then lc_grade = class 2 (probabality = 1);
	(2) if locunk <= 0.09090909090909091 and visible_days <= -147.27272727272725 and sct_ab_code2> 5.7272727272727275 then lc_grade = class 3 (probability = 0.62);
	(3) if locunk <= 0.09090909090909091 and visible_days <= -147.27272727272725 and locrhil<= 0.09090909090909091 and loclmsb<= 0.09090909090909091 and loclhil<= 0.09090909090909091 then lc_grade=class 2 (probability = 0.31);

Step 4 (Rule Set Creation): Extract rules from the decision tree and form a decision list to be used as the rule base for inference. Then, the decision list will be the input of the inference engine. Table 6 shows an iOVFDT-generated decision tree and its decision rules.

Step 5 (Disease Detection): Predict the type of disease for the new patient instance by using the rules in the decision list, and then, store the inferred possible diseases in TRS. Fig. 6 shows the GUI of the disease detector to cluster similar cases and infer possible diseases. The top frame contains a detailed description of the recovered cases, while the left-bottom frame contains inferred diseases with probability values followed.

Once similar instances are recovered by clicking the clustering button at the bottom right, users (e.g., doctors) can select the most similar ones. As shown on the top frame in Fig. 6, each instance is followed by a multiple choice box. When the clustering button is clicked, multiple choice boxes of all instances are selected because the system by default considers all cluttering-generated instances to be similar to the new one. After that, the classification button is clicked to show the probabilities of inferred diseases. Then, the doctor is encouraged to judge whether the inferred diagnosis is correct or not. Initially, the check box followed by 'Primary disease detection' is selected. When the doctor considers that any disease in 'Other disease detection' has high probability, the



Similar Cases

case1	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case2	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case3	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case4	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case5	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case6	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case7	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case8	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case9	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case10	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓
case11	if loclhil=0,locllow=0,loclup=0,locrhil=0,locrlow=0,locrmid=0,locunk=0,locrup=1,locoth=0,locmed=0,lo	✓

Primary disease detection:	Moderately Differentiated lung cancer : Grade II (0.30)	✓
Other disease detection:	Poorly Differentiated lung cancer: Grade III(0.25)	☐
Expert decision:		☐

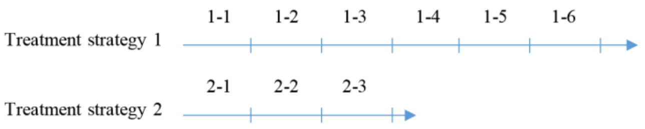
FIGURE 6. Snapshot for GUI-based disease detector.

corresponding check box can be selected. If diagnosis by the doctor does not appear in the inferred results, it can be manually typed using the 'Expert decision' input box with a probability. Once disease detection output is revised by the doctor, the new instance including the diagnoses will be stored to a TRS.

Step 6 (Disease Revision): Doctors need to make a definite clinical decision by designing the negative as failure in defeasible logic (i.e., negative rules) through a rule editor (see Fig. 7) to avoid impossible cases in TRS. For example, if score on one symptom is higher (or lower) than certain value, it is unlikely to be certain disease. Repeat this procedure until only one disease is left and we obtain updated temporary result set ΔTRS.

Step 7 (Treatment Strategy Reasoning): Different diseases have different treatment protocols. In addition, even for the same disease, because the physical conditions of a patient are ever-changing during the treatment period, the treatment strategies could vary frequently and cannot be determined dynamically by most traditional CDSS. Thus, we must define more general rules. In the following section, we first propose an assumption before giving two general rules as an example for treatment strategy reasoning.

Assumption: different patients have different treatment strategies. A treatment strategy can be divided into different phases, as shown below.



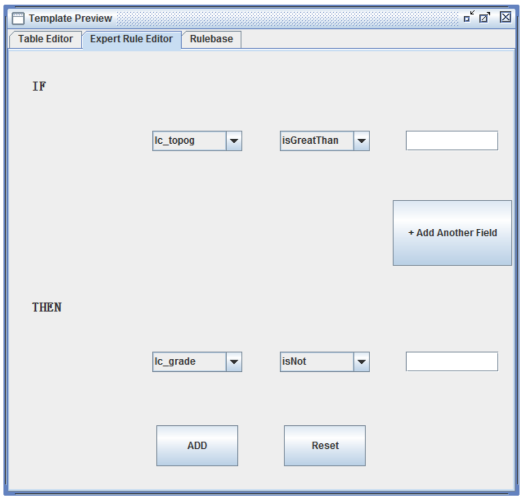


FIGURE 7. Rule editor.

Rule 1: If the symptoms of a patient at the current stage of treatment match those of a historical medical record during the same period, then the patient takes the corresponding treatment from that old record as the strategy in the current phase.

Rule 2: If no historical medical records match the new medical record in terms of symptoms at the current treatment phase, then, according to the symptoms of the new patient in all previous treatment phases Υ, we collect a group Ω with records that share similar symptoms with Υ in previous phases. The next step is to compare the symptoms of a new patient at the current stage with phases of each record in Ω

TABLE 7. Comparison of different classifiers used in CDTS.

Classifier type	Training instances	Model serialized size (bytes)	Tree size (nodes)	Tree size (leaves)	Tree depth
HoeffdingTree	342,000	649,840	97	50	21
HoeffdingAdaptiveTree	342,000	981,312	116	58	23
Hoeffding Optio Tree	342,000	2,907,536	432	221	21
iOVFDT	342,000	94,248	13	7	5

and determine the one whose symptoms are the closest to those of the new patient at this stage, and its corresponding treatment strategy is hereby applied to the patient at this stage. It is of note that the treatment phase numbers of the old and new medical records might not be the same in this case. Afterward, the treatment phase number for the new patient is adjusted to the historical medical record's next treatment phase number. Repeat Rule1 until the patient fully recovers.

For example, a dataset Ω will be established when the symptoms of a new patient in the third phase do not match the historical records at the same time point. Then, we go through all of the records again, and if any two phases of one historical record share similar symptoms with the first two phases of the new patient, it will be kept in Ω. Given that any historically successful treatment η has m treatment phases on average, the comparison procedure will be conducted $C_m^2 = \mathrm{m(m-1)/2}$ times at most. If there are n cases, then the time complexity is $\mathrm{O(n^*m^2)}$. This method is used to capture extended similar historical cases. After the establishment of Ω, all of the phases of the cases in Ω are compared with the third phase of the new patient treatment on the symptoms to find the most similar one with the maximum rank at the similarity measurement matrix ∂, which determines the treatment strategy for the new patient at the third phase.

C. RESULTS AND ANALYSIS

To enlarge the training dataset, we randomly resample the dataset 1000 times without replacement by using the filter *Resample* in Weka. In Table 7, iOVFDT constructs the smallest tree model compared with the other three classifiers in terms of the nodes, leaves and depth. HT and HAT also perform well in the construction of a nice tree model with 97 and 116 nodes, 50 and 58 leaves, and 21 and 23 depths, respectively. HOT has the largest tree size, with more nodes and leaves, which implies training dataset overfitting.

As shown in Figs. 8 and 9, Hoeffding-based tree classifiers have better performance compared to iOVFDT in terms of the classification correctness (i.e., accuracy) and kappa statistic (i.e., stability). With continuous incoming instances, the accuracy and stability of Hoeffding-based tree classifiers rise with some fluctuation, while the performance of iOVFDT is kept at a horizontal level. Similar to HAT, the voted ensemble multi-classification algorithm (VEMCA) also performs well in terms of accuracy and stability and outperforms HAT when the number of instances is larger than 340000.

Figs. 10 and 11 compare the evaluation time of the CPU and the memory cost of the five classification algorithms. iOVFDT shows the fastest speed and the smallest memory cost among the five classification algorithms. HT gives better performance than HAT and HOT with 6.93 seconds in CPU occupation and a 0.91-Mb memory cost. VEMCA's evaluation time and memory are 20.73 seconds and 1.62 Mb, respectively.

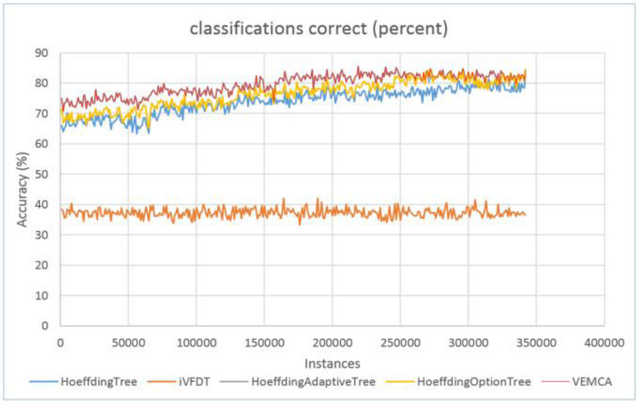


FIGURE 8. Comparison of classification correctness.

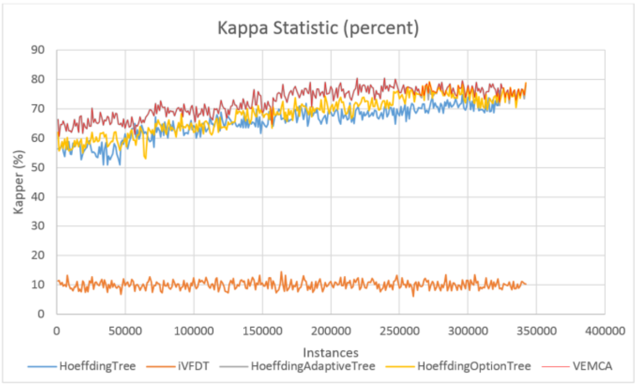


FIGURE 9. Comparison of the Kappa statistic.

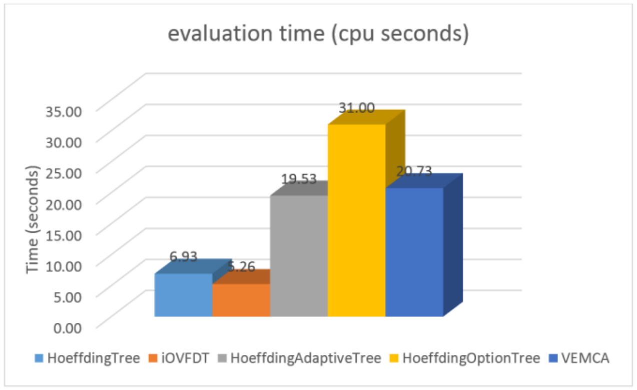


FIGURE 10. Comparison of the evaluation time of the CPU.

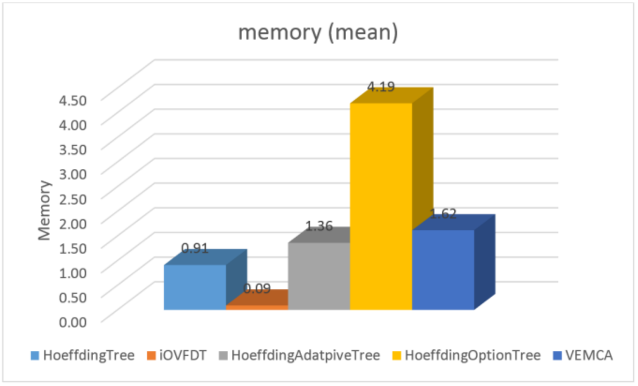


FIGURE 11. Comparison of the memory cost.

VIII. CONCLUSIONS

Modern techniques such as CDSS and HIS have substantially facilitated clinical diagnosis and treatment. However, some of the emergent problems remain to be considered. The first problem is challenging the traditional data analysis techniques due to rapidly increasing amounts of multi-dimensional clinical data. The second is the difficulty of clinical information integration. Clinical information from different contexts are heterogeneous in terms of structure and semantics. For the benefit of patients and long-term clinical development, it is very promising to integrate clinical information (e.g., outpatients' medical records) from different clinicians or hospitals to assist in clinical decision-making. The third problem is the call for personalized medicine and professional medical treatment, because different doctors or hospitals are good at diagnosing and treating different diseases, and the patients have the freedom to choose among them.

To solve the three problems above, this paper proposes a clinical diagnosis and treatment system (CDTS) that assists patients in choosing clinicians or hospitals according to their requirements (e.g., distance from home to hospital). The foundation for disease detection in CDTS consists of decision trees that are created by our voted ensemble multi-classification algorithm that enables decision-tree-based data stream mining. Furthermore, to integrate clinical documents and heterogeneous health information systems, a new clinical tabular document model is proposed, which represents clinical documents in tabular format and maintains consistent vocabulary terms and semantic relations among different contexts through the term chain and semantic relation chain. Additionally, a novel semantic inference algorithm is designed for disease detection and treatment suggestion, based on the rules that are generated by decision trees and similarity computations among medical records. The contribution of this paper can be summarized as follows:

- ➢ A unified clinical tabular document model (CTDM) implemented by a new clinical tabular document language (DocLang, XML based) facilitates the interoperability among different clinical decision support systems (CDSS).
- ➢ A new voted ensemble multi-classification algorithm is proposed, in terms of running multiple decision tree-based classification algorithms simultaneously on the same data stream and voting for the best output. The results are associated with statistical information on the accuracy of classification, which provides proof of optimization.
- ➢ A novel clinical diagnosis and treatment system with a newly designed semantic inference algorithm supports clinicians in the decision making process as well as saving time and expense for the patients.

For future work, it is necessary to expand DocLang to become a more comprehensive markup language by importing more semantic relation types for more complex clinical documents. These studies are in progress and are expected to present valuable results later.

REFERENCES

[1] M. A. Makary and M. Daniel, "Medical error—The third leading cause of death in the US," *BMJ*, vol. 353, p. i2139, May 2016.

[2] D. Kopec, M. H. Kabir, D. Reinharth, O. Rothschild, and J. A. Castiglione, "Human errors in medical practice: Systematic classification and reduction with automated information systems," *J. Med. Syst.*, vol. 27, no. 4, pp. 297–313, Aug. 2003.

[3] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson, Eds., *To Err is Human: Building a Safer Health System*, Washington, DC, USA: National Academy Press, 2000.

[4] D. W. Bates *et al.*, "The Costs of adverse drug events in hospitalized patients," *JAMA*, vol. 277, no. 4, pp. 307–311, 1997.

[5] H. Singh *et al.*, "Types and origins of diagnostic errors in primary care settings," *JAMA Internal Med.*, vol. 173, no. 6, pp. 418–425, Mar. 2013.

[6] W. Rogers, B. Ryack, and G. Moeller, "Computer-aided medical diagnosis: Literature review," *Int. J. Biomed. Comput.*, vol. 10, no. 4, pp. 267–289, Aug. 1979.

[7] D. Kopec and D. Michie, *Mismatch Between Machine Representations and Human Concepts: Dangers and Remedies*. Brussels, Belgium: Forecasting and Assessment in Science and Technology, 1992.

[8] D. Wallace and D. R. Kuhn, "Software quality lessons from medical device failure data," U.S. Dept. Commerce, Technol. Admin., Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR - 6407, 1999.

[9] D. L. Bates *et al.*, "Effect of computerized physician order entry and a team intervention on prevention of serious medication errors," *JAMA*, vol. 280, no. 15, pp. 1311–1316, 1998.

[10] W. G. Baxt, "Application of artificial neural networks to clinical medicine," *Lancet*, vol. 346, pp. 1135–1138, Oct. 1995.

[11] M. S. Hossain, "Cloud-supported cyber–physical localization framework for patients monitoring," *IEEE Syst. J.*, to be published. [Online]. Available: http://dx.doi.org/10.1109/JSYST.2015.2470644

[12] E. S. Berner, Ed., *Clinical Decision Support Systems: Theory and Practice*, 2nd ed. New York, NY, USA: Springer, 2007, pp. 3–22.

[13] B. Rothman, J. C. Leonard, and M. M. Vigoda, "Future of electronic health records: Implications for decision support," *Mount Sinai J. Med.*, vol. 79, no. 6, pp. 757–768, 2012.

[14] M. H. Trivedi *et al.*, "Barriers to implementation of a computerized decision support system for depression: An observational report on lessons learned in 'real world' clinical settings," *BMC Med. Inform. Decision Making*, vol. 9, p. 6, Jan. 2009.

[15] C. Sáez, A. Bresó, J. Vicente, M. Robles, and J. M. García-Gómez, "An HL7-CDA wrapper for facilitating semantic interoperability to rule-based clinical decision support systems," *Comput. Methods Programs Biomed.*, vol. 109, no. 3, pp. 239–249, Mar. 2013.

[16] B. Rothman, J. C. Leonard, and M. M. Vigoda, "Future of electronic health records: Implications for decision support," *Mount Sinai J. Med., J. Transl. Person. Med.*, vol. 79, no. 6, pp. 757–768, Nov./Dec. 2012.

[17] J. Gholap, V. P. Janeja, and Y. Yesha, "Unified framework for clinical data analytics (U-CDA)," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct./Nov. 2015, pp. 2939–2941.

[18] S. Hussain and S. Lee, "Semantic transformation model for clinical documents in big data to support healthcare analytics," in *Proc. 10th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Oct. 2015, pp. 99–102.

[19] M. Herland, T. M. Khoshgoftaar, and R. Wald, "Survey of clinical data mining applications on big data in health informatics," in *Proc. 12th Int. Conf. Mach. Learn. Appl. (ICMLA)*, vol. 2. Dec. 2013, pp. 465–472.

[20] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *Proc. Int. Symp. Intell. Data Anal.*, Aug. 2009, pp. 249–260.

[21] R. J. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann Series in Machine Learning). San Mateo, CA, USA: Morgan Kaufmann, Jan. 1993.

[22] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA, USA: Wadsworth, 1994.

[23] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2000, pp. 71–80.

[24] N. Oza and S. Russell, "Online bagging and boosting," in *Artificial Intelligence and Statistics*. San Mateo, CA, USA: Morgan Kaufmann, 2001, pp. 105–112.

[25] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda, "New ensemble methods for evolving data streams," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2009, pp. 139–147.

[26] E. Ocampo, M. Maceiras, S. Herrera, C. Maurente, D. Rodríguez, and M. A. Sicilia, "Comparing Bayesian inference and case-based reasoning as support techniques in the diagnosis of acute bacterial meningitis," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10343–10354, 2011.

[27] I. Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Mateo, CA, USA: Morgan Kaufmann, 1997.

[28] L. OtavioAlvares, "Raciocínio Baseadoem Casos," in *Informática UFRGS*, 2006. [Online]. Available: http://www.inf.ufsc.br/~luis.alvares/INE5633/RaciocinioBC.pdf

[29] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 2004.

[30] R. L. De Mantaras *et al.*, "Retrieval, reuse, revision and retention in case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 215–240, 2005.

[31] J. Kolodner, *Case-Based Reasoning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[32] S. K. Pal and S. C. K. Shiu, *Foundations of Soft Case-Based Reasoning*. Hoboken, NJ, USA: Wiley, 2004.

[33] T. Cerquitelli, S. Chiusano, and X. Xiao, "Exploiting clustering algorithms in a multiple-level fashion: A comparative study in the medical care scenario," *Expert Syst. Appl.*, vol. 55, pp. 297–312, Aug. 2016.

[34] M. Ben Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semantic interoperability in one M2M architecture," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 35–41, Dec. 2015.

[35] I. Lemmens, J. P. Koster, and S. Valera, "Achieving interoperability at semantic level," in *On the Move to Meaningful Internet Systems* (Lecture Notes in Computer Science), vol. 9416, I. Ciuciu *et al.*, Eds. Cham, Switzerland: Springer, 2015.

[36] A. O. Shigarov, "Table understanding using a rule engine," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 929–937, 2015.

[37] G. Xiao, "Semantic document exchange for electronic business through user-autonomous document sense-making," Ph.D. dissertation, Univ. Macau, Zhuhai, China, 2015.

[38] S. Yang and J. Guo, "A multi-viewed document representation for semantic document exchange," in *Proc. IEEE 12th Int. Conf. e-Bus. Eng. (ICEBE)*, Oct. 2015, pp. 154–159.

[39] A. M. Namboodiri and A. K. Jain, "Document structure and layout analysis," in *Digital Document Processing*. London, U.K.: Springer, 2007, pp. 29–48.

[40] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 294–308, Mar. 1998.

[41] J. Guo, "Collaborative conceptualisation: Towards a conceptual foundation of interoperable electronic product catalogue system design," *Enterprise Inf. Syst.*, vol. 3, no. 1, pp. 59–94, 2009.

[42] S. Nam, S. Lee, J. G. B. Kim, and H. G. Kim, "STEP: An ontology-based smart clinical document template editing and production system," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 3005–3015, 2014.

[43] Z. Gong, M. Muyeba, and J. Guo, "Business information query expansion through semantic network," *Enterprise Inf. Syst.*, vol. 4, no. 1, pp. 1–22, 2010.

[44] J. Guo, L. Xu, Z. Gong, C. P. Che, and S. S. Chaudhry, "Semantic inference on heterogeneous e-marketplace activities," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 2, pp. 316–330, Mar. 2012.

[45] C. S. G. Khoo and J.-C. Na, "Semantic relations in information science," *Annu. Rev. Inf. Sci. Technol.*, vol. 40, no. 1, p. 157, 2006.

[46] G. Hulten and L. Spencer, "PedroDomingos: Mining time-changing data streams," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 97–106.

[47] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2005.

[48] A. Bifet, "Adaptive learning and mining for data streams and frequent patterns," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 55–56, 2009.

[49] B. Pfahringer, G. Holmes, and R. Kirkby, "New options for hoeffding trees," in *Proc. AI*, 2007, pp. 90–99.

[50] B. Pfahringer, G. Holmes, and R. Kirkby, "New options for hoeffding trees," in *Proc. Austral. Joint Conf. Artif. Intell.*, Dec. 2007, pp. 90–99.

[51] H. Yang and S. Fong, "Incrementally optimized decision tree for noisy big data," in *Proc. 1st Int. Workshop Big Data, Streams Heterogeneous Source Mining, Algorithms, Syst., Program. Models Appl.*, Aug. 2012, pp. 36–44.

[52] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2001, pp. 377–382.

[53] *Cancer Data Access System of American National Cancer Institute*, accessed on May 25, 2016, [Online]. Available: https://biometry.nci.nih.gov/cdas/

[54] S. Tsumoto, "Automated extraction of hierarchical decision rules from clinical databases using rough set model," *Expert Syst. Appl.*, vol. 24, no. 2, pp. 189–197, 2003.

[55] S. Yang and J. Guo, "A novel approach for cross-context document reasoning in e-commerce," in *Proc. 6th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Sep. 2015, pp. 1018–1025.

[56] J. Guo, "SDF: A sign description framework for cross-context information resource representation and interchange," in *Proc. Enterprise Syst. Conf. (ES)*, Aug. 2014, pp. 255–260.

[57] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generat. Comput. Syst.*, vol. 66, pp. 30–35, Jan. 2016.

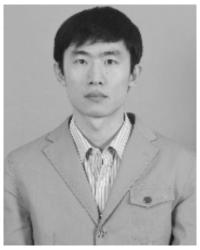
[58] Y. Zhang, D. Zhang, M. M. Hassan, A. Alamri, and L. Peng, "CADRE: Cloud-assisted drug recommendation service for online pharmacies," *Mobile Netw. Appl.*, vol. 20, no. 3, pp. 348–355, 2015.

[59] M. Chen, Y. Ma, J. Song, C.-F. Lai, and B. Hu, "Smart clothing: Connecting human with clouds and big data for sustainable health monitoring," *Mobile Netw. Appl.*, vol. 21, no. 5, pp. 825–845, 2016.

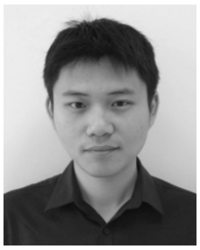
[60] M. Chen, "NDNC-BAN: Supporting rich media healthcare services via named data networking in cloud-assisted wireless body area networks," *Inf. Sci.*, vol. 284, pp. 142–156, Nov. 2014.

[61] J. Wan, C. Zou, S. Ullah, C.-F. Lai, M. Zhou, and X. Wang, "Cloud-enabled wireless body area networks for pervasive healthcare," *IEEE Netw.*, vol. 27, no. 5, pp. 56–61, Sep./Oct. 2013.

[62] X. Chen, L. Wang, J. Ding, and N. Thomas, "Patient flow scheduling and capacity planning in a smart hospital environment," *IEEE Access*, vol. 4, pp. 135–148, 2016.



SHUO YANG (M'15) received the master's degree in software engineering from Dalian Jiaotong University, Dalian, China, in 2013. He is currently pursuing the Ph.D. degree in E-commerce technology with the Department of Computer and Information Science, University of Macau. His research interests include document engineering and semantic inference, mainly applied to the field of E-commerce, E-marketplace, and clinical area.



RAN WEI received the master's degree in pharmaceutical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, in 2011. He is currently pursuing the Ph.D. degree in biomedical sciences with the Department of Microbiology, Rutgers University, Newark. His research interest focuses on the quantitative analysis of epigenetic pathways in vitamin D-regulated lung immune responses.

JINGZHI GUO (M'05) received the the B.Econ. degree in international business management from the University of International Business and Economics, Beijing, China, in 1988, the M.Sc. degree in computation from The University of Manchester, Manchester, U.K., in 2010, and the Ph.D. degree in internet computing and e-commerce from Griffith University, Brisbane, Australia, in 2005. He is currently an Associate Professor in e-Commerce Technology with the University of Macau, Macau, China. His research interests include concept representation, semantic integration, and collaboration systems, mainly applied to the fields of e-commerce, e-marketplace, e-banking, and the virtual world.

LIDA XU (M'86–SM'11) is currently an Academician of the Russian Academy of Engineering. He was recognized as a Highly Cited Researcher by Thomson Reuters (Clarivate) in 2016.