A Privacy-Preserving Framework for Distributed Clinical Decision Support

George Mathew, Zoran Obradovic
Center for Information Science and Technology
Temple University
Philadelphia, PA, USA
{George.Mathew, Zoran.Obradovic}@temple.edu

***Abstract*— We propose a framework for distributed knowledge-mining that results in a useful clinical decision support tool in the form of a decision tree. This framework facilitates knowledge building using statistics based on patient data from multiple sites that satisfy a certain filtering condition, without the need for actual data to leave the participating sites. Our information retrieval and diagnostics supporting tool accommodates heterogeneous data schemas associated with participating sites. It also supports prevention of personally identifiable information leakage and preservation of privacy, which are important security concerns in management of clinical data transactions. Results of experiments conducted on 8 and 16 sites with a small number of patients per site (if any) satisfying specific partial diagnostics criteria are presented. The experiments coupled with restricting a fraction of attributes from sharing statistics as well as applying different constraints on privacy at various sites demonstrate the usefulness of the tool.**

Keywords - medical informatics; graph data mining; clinical decision support systems.

I. INTRODUCTION

One of the five recommendations made for Clinical Decision Support (CDS) Systems in connection with the practice of Evidence-based Medicine was to "develop maintainable technical and methodological foundations for computer-based decision support" [1]. Also, medical domain is "characterized by much judgmental knowledge" [2]. Consequently, a Clinical Decision Support (CDS) system that can provide suggestive knowledge representations based on data sets with patient attributes that are similar to the attributes of the patient in context is valuable to a medical practitioner. Invariably, there are situations when the number of local samples to draw conclusions from, is none or few. Aggregating similar samples from other distributed (off-site databases) would help in making better decisions. As an example, for diagnosis and treatment of a specific patient that is an outlier in his/her medical practice, a physician would like to obtain information that goes beyond what is seen at that office and what might be available of a more general patient population externally (as this patient is quite different from a typical case). To support diagnostics in such situations we are providing a framework that can:

1. Query other locations to retrieve summary diagnosis statistics for patients filtered based on specific properties (e.g. females in their twenties with normal body mass index who are type 2 diabetics and have a specific overall cardiac diagnosis based on Single Proton Emission Computed Tomography (SPECT) images); and
2. Allow learning a diagnostics model (classifier) based on statistics on other attributes obtained from multiple sites for patients that satisfy the specific query of interest. (e.g. other attributes could be some of 23 partial diagnoses obtained from SPECT images obtained from sites that allow sharing statistics on some of these additional features)

Due to legal and regulatory implications, dynamically acquiring patient data directly from other sites is difficult. Even if two sites agree to collaborate, the data schema in the two sites could be different. It is also vital to protect the confidentiality of medical data in clinical transactions [3]. Given these obstacles, it is advantageous to model a CDS system that makes use of statistics about the samples from distributed sites that are structurally similar to the current patient context, rather than model a CDS system that makes use of the actual data from other sites. We propose a framework for a CDS system that does not require the actual patient data from distributed sites. Also, it does not require the participating sites to have identical data schema and can accommodate local site policies that may prevent in divulging specific attributes.

In our decision tree building method, the site-wide data schema (of attributes and classes) is dynamically generated with no apriori knowledge of local data schema. In addition, our work outlines a framework that encompasses the lifecycle of a distributed CDS system activity. This includes the query process, a clearinghouse channel, dynamic schema generation (with flexibility for local sites to enforce their attribute policies) and analytics among the sites. A typical starting point for a clinical transaction in such a system is a query issued by a medical practitioner for suggestions towards decision-making.

We use a graph-based approach since data represented by graphs can capture the expressive power of the structure of data without requiring a specific relational data schema at multi-site situations which is often very difficult or infeasible to achieve. Representing transaction data in graph format allows graph data mining techniques [4,5] to be applied in enforcing business rules at local sites. A common theme that emerges from healthgrid (eg: MammoGrid [6], caGrid [7])

initiatives, RHIOs, HealthSystem Consortiums and HL7-based web service initiatives [8] is that of a distributed network of sites. Graphs provide the theoretical formalism for modeling distributed networks. Hence we use graphs as the foundation for building a framework.

II. Related Work

Epstein's [9] rendering of research methodologies underlines the interest in Clinical Data Mining. Due to recent worldwide interest in unifying medical data across the globe by various organizations, a great deal of research and prototyping has been done at the application level [10]. Data mining studies in specialties within medical domains (see [11], [12]) appear in the literature. The focus of this paper is a framework for CDS systems that will aid medical practitioners with knowledge mined from distributed systems using a graph-based approach.

MYCIN [13] was one of the first Rule-based CDS Systems. The system had more than 500 pre-defined rules and was constrained to the identification of infectious bacteria (and antibiotics recommendation). CADUCEUS [14] was a medical expert system developed in University of Pittsburgh based on the principles of MYCIN. Both MYCIN and CADUCEUS are self-contained systems. They cannot harness the "collective intelligence" of data available in distributed sites and cloud. These kinds of systems were not dynamic to accommodate ad hoc queries, which is one of the objectives of the framework explored in this study.

Decision Tree provides a data structure for representing paths of traversals in a classification problem. We use decision tree as the vehicle for knowledge conveyance in the framework. A Graph-based Decision Tree induction was detailed by Nguyen et al [15]. However, this is not a distributed model. Extending this model directly to distributed systems would require the data to be shipped to a centralized broker and so would violate privacy policies. The distributed decision tree generation process we developed was inspired by the theoretical sketch suggested by Caragea et al [16]. Bar-Or et al [17] also used the principles outlined by Caragea et al to introduce a distributed decision tree induction. However, both models (Cragea et al & Bar-Or et al) assume identical relational data schema (i.e., homogeneous data schema) in all the sites and the schema be known to the central broker. Our model can accommodate non-identical data schema (i.e., heterogeneous schema) at participating sites and the broker does not need prior knowledge of any data schema. This gives the flexibility for sites to participate or not participate at will and also maintain data schema independence at individual sites. We propose graph databases as data containers. The two data partitioning models for distributed data sets outlined by Caragea et al [16] are horizontal & vertical fragmentation. In horizontal data partitioning, all data instances have identical attribute sets and the instances corresponding to a specific value of an attribute will be located in different sites. In vertical partitioning, the attribute set is subdivided and each site holds values for the attributes in the subdivision assigned to it. In this case, to get a complete data instance, the sub-tuples are to be combined from the different sites. The graph-based model we propose allows for a flexible hybrid model (see Fig. 1). In the hybrid model, there is no predefined set of attributes for each data instance. Since a graph structure is flexible to store only the attributes that have values, when the data instances are arranged in an equivalent row-column format, there can be 'holes' in the cells. Graph databases have the flexibility of associating attributes only with nodes that require those attributes. This is an advantage compared to relational databases, where adding attributes (and hence more columns) usually result in sparse tables.

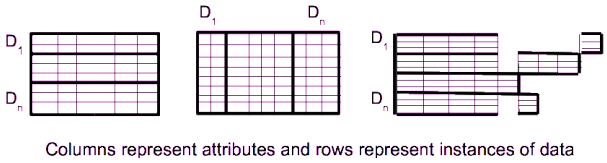

Figure 1. Horizontal Data Fragmentation (left), Vertical Data Fragmentation (middle) and Hybrid Data Fragmentation (right).

Due to the flexibility of graph databases, we do not require a single homogeneous schema to be enforced at each site. Our model can build the data attributes and classes on the fly for up to date information and does not restrict the local sites from making changes to their own database schemas.

Our framework allows for a flexible localizable constraint enforcement mechanism to be used to assure that local policies related to Personally Identifiable Information and privacy constraints are implemented. In reported experiments we use attribute-based constraints. However, this can be replaced by any other localized constraint filters (eg: differential privacy [18]). Attribute-based constraints specify what attributes at particular sites should be blocked from being disclosed.

III. Methodology

There are 2 stages in our methodology. The first stage involves generating a global schema of the distributed data sets relative to a query. The second stage involves generating the statistics and decision tree nodes based on the global schema constructed in the first stage. In the algorithm presented in this paper we handle categorical attributes; however, continuous attributes can be accommodated as well with some extensions to our algorithm. For brevity of discussion, we do not deal with decision tree pruning. In each of the 2 stages, a two-phase scheduling [19] (local processing at the sites and global synthesis at the common broker) is made use of.

Due to factors related to scalability, site registration, vetting and data privacy preservation, a query by a medical practitioner to mine decision-support information from distributed sites should be brokered through a Clearing House (CH). The query is channeled through an institutional

gateway to the CH as a query graph. A query graph is a graph data structure used to communicate the query. Essentially, a query graph is a prototype graph that can be used for pattern matching against the graph databases in individual sites. A query graph can be composed of constant value for attributes (eg: gender=male) and list of values for categorical attributes (eg: headache ~ {mild, severe, acute}).

A. Stage 1

At the discretion of CH, the query graph is forwarded to k participating sites. In order for all these processes to take place, there has to be agents in all sites as well as CH for distributed and coordinated communications. We do not cover the details here. However, there are publicly available JAVA-based agents developed as part of distributed data mining projects (see [20]). When a site i receives a query graph (from the CH), the following processing is done locally. The graph database is traversed and individual patient record graphs are checked for a subgraph isomorphic match with an instance of the query graph. Let G_{Mi} denote the set of all such matching graphs. Let τ_i denote the local constraints in terms of local policies and PII (personally identifiable information) specific attributes. Applying the constraints τ_i on G_{Mi} results in the set of graphs G_{Ti}. Based on G_{Ti}, the site will generate the set A_i of attributes, the set V^i_x of unique values for each $x \in A_i$ and the set C_i of classes. Let $n_i = |G_{Ti}|$. Each site i constructs the metadata tuple $<A_i,\{V^i_x \mid x \in A_i\},C_i,n_i>$ and sends it to the CH. The CH will aggregate the metadata tuples to generate global schema using the following:

$$< \bigcup_{i=1}^{k} A_i,\ \{ \bigcup_{i=1}^{k} V_x^i \mid x \in \bigcup_{j=1}^{k} A_j \},\ \bigcup_{i=1}^{k} C_i,\ \sum_{i=1}^{k} n_i > \quad .....(1)$$

This can be represented in the following simple format:

$$<\{a_1,a_2...,a_m\},\{\{v_1^1,v_2^1,..,v_{d_1}^1\},..,\{v_1^m,v_2^m,..,v_{d_m}^m\}\},\{c_0,..c_t\},n>$$

For example, let the metadata from the only 2 participating sites be as follows:

<{a1,a2},{{med,high},{0,1}},3>
<{a1,a3},{{low,med},{positive,negative}},4>

Then the global schema will be:

<{a1,a2,a3},{{low,med,high},{0,1},{positive,negative}},7>

B. Stage 2

The CH shares the global schema with the k sites selected in the previous stage. Each site generates *crosstable matrices* [17] for individual attributes in the global schema. For a given attribute u, the $(x,y)^{th}$ entry of the crosstable matrix represents the number of graphs in G_{Ti} for which the attribute u exists with value x and that the graph belong to class y. If an attribute u is not missing from any data graph, the sum of the elements of the crosstable matrix for u will be the same as $|G_{Ti}|$; otherwise, the sum of the elements of the crosstable matrix will be less than $|G_{Ti}|$. The crosstable matrices generated by the sites are sent to the CH. The CH will combine the site-specific crosstable matrices to create global crosstable matrices for each attribute. If there are k sites and CT^g_u represents the global crosstable matrix, while CT^l_u represents the local crosstable matrix for attribute u,

$$CT^g_u(x,y) = \sum_{l=1}^{k} CT^l_u(x,y)$$

Each node of the decision tree is constructed using node selection criteria on the global crosstable matrices. Let the crosstable matrix for an attribute u with m values $b_1, b_2, ..., b_m$ and n classes $c_1, c_2, ..., c_n$ be as follows:

$$\begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mn} \end{bmatrix}$$

Then, the weighted average impurity measure for u is:

$$\frac{-1}{\sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}} \left(\sum_{i=1}^{m} \left\{ \sum_{j=1}^{n} v_{ij} \log_2 \frac{v_{ij}}{\sum_{k=1}^{n} v_{ik}} \right\} \right) \quad ..... (2)$$

This is based on Quinlan's ID3 decision tree algorithm [21]. The attribute with the smallest value for the weighted average impurity measure is used for the split [22].

Algorithm

The goal of our Distributed ID3-based Decision Tree (DIDT) algorithm is to derive the exact decision tree that would be obtained by applying ID3 decision tree on the combined set of all the graphs in the distributed sites when restricted to a subset of examples that satisfy a specific query filter. An important assumption is that G_{Ti} in each individual site i will remain the same throughout the processing of the algorithm (i.e., during the decision tree creation process). Another assumption is that all the distributed sites follow agreed upon attribute names, data types and class names (or database schema elements) from a common vocabulary (see Mathew & Obradovic [23]). An Interface Engine could mediate the standard naming conventions between the CH and the local site.

The proposed distributed knowledge generation algorithm for clinical decision support consists of the following steps:

01. A query by the medical practitioner in a clinical context is sent to local gateway.
02. The local gateway qualifies the query, transform the query into a query graph and sends to the Clearing House (CH).

Stage 1

03. CH passes along the query graph to a selected number of sites $S_1, ..., S_k$.
04. At site i, query graph is checked for subgraph isomorphism with the data graphs. The set G_{Mi} of matching graphs is generated by site i.
05. The site-specific constraints τ_i are applied on $G_{Mi.}$ Let the set of resulting graphs be G_{Ti}.
06. The set of all attributes A_i, the set of values for each attribute $\{V_x \mid x \in A_i\}$ and the set of all classes C_i for the graphs in G_{Ti} as well as $|G_{Ti}|$ are

generated by site *i*. The metadata tuple in the form $<A_i, \{V^i_x \mid x \in A_i\}, C_i, |G_{Ti}|>$ is sent to CH.

07. CH creates a global schema using the tuple expression given in (1), which is an aggregation of all attributes, their possible values, classes and number of matching graphs in the *k* sites.

Stage 2

08. CH communicates global schema to the *k* participating sites in the form:

$$<\{a_1,...a_m\},\{\{v_1^1,..v_{d_1}^1\},..,\{v_1^m,..v_{d_m}^m\}\},\{c_0,..c_t\}>$$

09. Based on the attributes in the global schema, each of the sites S_i generates uniform templates for the *crosstable matrices* for individual attributes. The crosstable matrix template for a_x takes the form:

	c_0 c_1 ⋯ c_t
v_1^x	
v_2^x	
⋮	
$v_{d_x}^x$	

The crosstable matrices for all attributes at each site are sent to the CH.

10. The CH adds site-specific crosstable matrices for each attribute a_x and creates global crosstable matrix for a_x.
11. The weighted average impurity measure for the attributes are calculated using (2) and the attribute for split is chosen based on smallest value of weighted average impurity measure (highest gain). Only attributes that span all the G_{Ti}'s are included in the impurity calculations. i.e., if the sum of the elements in the crosstable matrix of an attribute equals the total number of filtered instances from the *k* sites, it qualifies to be a candidate for the splitting node.
12. To proceed to the next level of the decision tree, updated query graphs are generated for each branch of the decision tree, using the values of the attribute selected. The number of query graphs generated depends on the number of values for the attribute chosen. The process repeats from step 03 with each of the query graphs until the classes are reached in the leaf nodes.
13. The CH sends the final Decision Tree to the query originator.

IV. EXPERIMENTS

Various experiments were performed to validate the algorithm using SPECT Heart data set [24] of patients from UCI machine learning data repository. The dataset summarizes features of 267 cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified as normal or abnormal based on the features. The SPECT Heart data set had 22 binary attributes and 2 classes. We labeled the attributes as a1, a2, …, a22. The two classes were labeled as c0 and c1. The data set had 80 training and 187 test instances, for a combined total of 267 instances. 11 duplicate data instances with all 0's and class c0 were eliminated, resulting in 256 data instances. These 256 data instances were shuffled and equal number of instances were distributed to the sites in all series of experiments. Cross-validation for our DIDT algorithm is done by reserving one of the sites for testing and the remaining sites for training.

Data graphs were distributed among simulated sites. The decision tree from the distributed data graphs generated by our algorithm is compared to the decision tree for the equivalent aggregate data set using public domain weka software [25] (for ID3 algorithm).

In all experiments, a formal representation of queries, using the boolean expression operators shown in Table 1 was made use of.

TABLE I. BOOLEAN OPERATORS

symbol	operation	precedence
!	negation	high
∧	conjunction	medium
∨	disjunction	low

A. *Learning from multiple sites that share statistics on all attributes*

We initially ran our DIDT algorithm on the 256 data instances distributed among 8 sites without any query. This creates the decision tree for the complete data set. For reference purposes, we will call this tree the 'complete tree'. The result was compared to the complete tree generated by weka software against the combined 256 data instances.

TABLE II. ACCURACY FOR COMPLETE TREE

algorithm	cross-validation	correctly classified	accuracy
Weka ID3	8	183/256	71.48%
DIDT	8	181/256	70.70%

The values in table II are normal since the cross-validation sets in our algorithm need not match the cross-validation sets in Weka.

In the next 2 experiments, we used the query a3∧a5∧a8 to restrict only to patients with positive partial diagnosis based on a3, a5 and a8 criteria. The 256 data instances were shuffled and distributed to *x* number of sites (*x*=8,16). There were 37 instances matching the query and these were also spread among all sites such that no site had more than 4 such patients, which is insufficient for decision making according to local statistics. The following table shows the DIDT results and the results obtained by weka software on the combined data set.

TABLE III. ACCURACY FOR QUERY TREE

algorithm	number of sites	cross-validation	correctly classified	accuracy
Weka ID3	n/a	8	32/37	86.49%
DIDT	8	8	33/37	89.18%
Weka ID3	n/a	16	32/37	86.49%
DIDT	16	16	34/37	91.89%

As mentioned before, these values are close enough and since the total is small, the higher accuracy does not mean any improvement due to our algorithm.

To illustrate the working of our DIDT algorithm using the SPECT data set, consider the scenario of 16 sites. Shuffling records of 256 patients and distributing equal number of data instances to each site resulted in 16 data instances per site. Using the query a3∧a5∧a8 to filter data instances, following are sample statistics at some of the sites.

TABLE IV. SAMPLE RESULTS

site	attributes	classes	instances
1	a1,a2,a4,a6,a7,a9-a22	c1	2
7	<none>	<none>	0
14	a1,a2,a4,a6,a7,a9-a22	c0,c1	4

So the metadata for site 1 was:

<{a1,a2,a4,a6,a7,a9-a22},{{0,1}…,{0,1}},{c1},2>

Combining all metadata from the 16 sites, the global schema generated by the CH had the format:

<{a1,a2,a4,a6,a7,a9-a22},{{0,1}…,{0,1}},{c0,c1},37>

Using the global schema, each site generated the crosstable matrices for the attributes. At site 14, the crosstable matrices for a1, a2 & a4 were as follows:

$$\underset{a1}{\begin{pmatrix}1 & 3\\ 0 & 0\end{pmatrix}} \quad \underset{a2}{\begin{pmatrix}0 & 0\\ 1 & 3\end{pmatrix}} \quad \underset{a4}{\begin{pmatrix}0 & 3\\ 1 & 0\end{pmatrix}}$$

The rest of the crosstable matrices are omitted due to lack of space. Combining the crosstable matrices from all 16 sites resulted in the following sample matrices at the CH:

$$\underset{a1}{\begin{pmatrix}3 & 29\\ 0 & 5\end{pmatrix}} \quad \underset{a2}{\begin{pmatrix}1 & 12\\ 2 & 22\end{pmatrix}} \quad \underset{a4}{\begin{pmatrix}0 & 18\\ 3 & 16\end{pmatrix}}$$

Calculating the weighted average impurity measure for the attributes resulted in a13 having the lowest value. Hence a13 was chosen as the node for the split (see Fig. 2). Now, there were two paths to continue:

a3∧a5∧a8∧a13 and a3∧a5∧a8∧!a13

Successively following each of these and down-level paths resulted in the tree shown in Fig. 2.

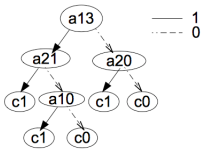

Figure 2. Decision Tree generated by the DIDT algorithm for the query a3∧a5∧a8

B. Learning from multiple sites when some sites constrain certain attributes

In these experiments, the number of sites was arbitrarily set at 16. The same query a3∧a5∧a8 was used for decision tree induction. The number of sites enforcing the constraints was set at 4 and 12. Enforcing the constraints was essentially blocking the attributes. In two sets of experiments, attributes a13 & a10 were blocked independently from being released. Note that a13 was the most dominant attribute in Fig 2.

TABLE V. RESULTS OF CONSTRAINING AN ATTRIBUTE

attribute blocked	# of sites blocking	cross-validation	correctly classified	accuracy
a13	4	16	35/37	94.59%
a13	12	16	35/37	94.59%
a13	16	16	35/37	94.59%
a10	4	16	30/37	81.08%
a10	12	16	30/37	81.08%
a10	16	16	30/37	81.08%

When no attribute was constrained from the distributed data sets, a13 was the dominant attribute in 13 of the 16 cross-validations for the DIDT run shown in table III (4th row). In all these 16 cross-validation runs, attribute a10 played a subdominant role in the structure of the decision tree. When attribute a13 was blocked (see table V), it was observed that a correlated attribute a20 took over as the dominant attribute and the trees for cross-validations stabilized. When a10 was blocked, the tree structure changed such that the node representing a10 was replaced by either a15 or a16 (with resultant down-level nodes). Both a15 & a16 were weakly correlated to a10 and this resulted in cross-validation trees with more errors (as shown in the latter half of table V).

In the next set of experiments, attributes a13 and a10 were blocked from being released simultaneously from different sites in exclusive and inclusive manner. In the exclusive experiments, at most one attribute was blocked from each site of interest. In the inclusive experiments, both attributes were blocked at the same time from sites of interest. For example, in an exclusive experiment of 8 sites of interest, we blocked attribute a13 from 4 sites and a10 from the other 4 sites. In inclusive experiment of 8 sites of interest, both a13 & a10 were blocked from all 8 sites at the same time.

TABLE VI. RESULTS OF COMBINED BLOCKING OF a13 & a10

type of blocking	# of sites blocking	cross-validation	correctly classified	accuracy
exclusive	8	16	32/37	86.49%
inclusive	8	16	32/37	86.49%

These results are not contradictory to the results in table V, since the number of sites chosen are dissimilar.

C. Verification of Heterogeneous Schema Accommodation

In order to verify the working of DIDT algorithm when sites have heterogeneous data schema, we used a set of experiments that parallel the ones in previous section *B*. In the experiments in section *B*, all the data instances were left intact and constraints were applied so that the attribute information does not leave the site. For the experiments in this section, we removed the attribute(s) under consideration from the data instances in the sites. This results in heterogeneous schema among sites.

For ease of comparison of results, the number of sites was fixed at 16 and the same query a3∧a5∧a8 was used for decision tree induction in all cases. In the first two experiments, attributes a13 & a10 were removed from x (x=4,12) sites. In the third experiment, attributes a13 and a10 were removed from 8 different sites in exclusive and inclusive manner. The results were consistent with those in tables V & VI, as expected.

V. Conclusion

We have outlined a flexible secure graph-based framework for a clinical decision support system that can protect patient personal identifiable information as well as assure data privacy. We demonstrated that a clinical decision support intelligence tool in the form of a decision tree can be constructed by using just the statistics related to the data distributed among the sites even when different sites have different restriction rules on release of statistics related to specific attributes of patients. In the proposed protocol, the data do not leave the sites and no rigid data schema structure is enforced on the collaborating sites. This makes it a viable option for building knowledge from sites that cannot disclose clinical data records due to privacy issues.

Acknowledgment

This project is funded in part under a grant with the Pennsylvania Department of Health. The Department specifically disclaims responsibility for any analyses, interpretations, or conclusions.

References

[1] I. Sim, P. Gorman, R. A. Greenes, R. B. Haynes, B. Kaplan, H. Lehman, and P. C. Tang, "Clinical Decision Support Systems for the Practice of Evidence-based Medicine", Journal of American Medical Informatics Association, Vol. 8, No. 6, Nov-Dec, 2001, pp. 527-534.

[2] W. van Melle, "MYCIN: A Knowledge-based Consultation Program for Infectious Disease Diagnosis", International Journal of Man-machine Stuides, Vol. 10, Issue 3, May 1978, pp. 313-322.

[3] E. D. Goldstein, "e-Healthcare", Aspen Publishers Inc., Gaitherberg, MD, USA, 2000.

[4] D. J. Cook, and L. B. Holder, "Mining Graph Data", Wiley-Interscience, Hoboken, NJ, USA, 2007.

[5] C. C. Aggarwal, and H. Wang, "Managing and Mining Graph Data", Springer, NY, USA, 2010.

[6] Mammogrid project, https://savannah.cern.ch/projects/mammogrid

[7] caGrid project, http://cagrid.org

[8] N. K. Janjua, M. Hussain, M. Afzal, and H. F. Ahmad, "Digital Health Care Ecosystem: SOA Compliant HL7 based Health Care Information Interchange", Proceedings of 3rd IEEE International Conference on Digital Ecosystems and Technologies, Istanbul, Turkey, May 2009, pp. 329-334.

[9] I. Epstein, "Clinical Data Mining, Integrating Practice and Research", Oxford University Press, New York, NY, USA, 2010.

[10] M. Cannataro, "Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare", Vols I & II. Medical Information Science Reference, Hershey, PA, USA, 2009.

[11] J. C. Prather, D. F. Lobach, L. K. Goodwin, J. W. Hales, M. J. Hage, and W. E. Hammond, "Medical data mining: knowledge discovery in a clinical data warehouse", Proceedings of AMIA Annual Fall Symposium, 1997, pp. 101-105.

[12] Q. Wang, E. Karamani-Liacouras, E. Miranda, U. S. Kanamala, and V. Megalooikonomou, "Classification of brain tumors using MRI and MRS", Proceedings of the SPIE Conference on Medical Imaging, 2007.

[13] B. G. Buchanan, and E. W. Shortliffe, "Rule Based Expert Systems: The MYCIN experiments in the Stanford Heuristic Programming Project", Addison-Wesley, Reading, MA, 1984.

[14] D. G. Bobrow, S. Mittal, and M. J. Stefik, "Expert Systems: perils and promise", Communications of the ACM, Vol 29, Issue 9, Sep 1986, pp. 880-894.

[15] P. C. Nguyen, K. Ohara, A. Mogi, H. Motoda, and T. Washio, "Constructing Decision Trees for Graph-Structured Data by Chunkingless Graph-Based Induction", Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Volume 3918, Springer Berlin, 2006, pp. 390-399.

[16] D. Caragea, A. Silvescu, and V. Honavar, "A Framework for Learning from Distributed Data Using Sufficient Statistics and its Application to Learning Decision Trees", International Journal on Hybrid Intelligent Systems, Vol 1, Issue 1-2, April 2004, pp. 80-89.

[17] A. Bar-Or, D. Keren, A. Schuster, and R. Wolff, "Hierarchical Decision Tree Induction in Distributed Genomic Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 8, Aug 2005, pp. 1138-1151.

[18] A. Friedman, and A. Schuster, "Data Mining with Differential Privacy", Proceedings of 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington D.C., July 2010, pp. 493-502.

[19] P. Luo, K. Lu, Z. Shi, and Q. He, "Distributed Data Mining in Grid Computing Environments", Future Generation Computer Systems, Vol. 23, Issue 1, Jan 2007, pp. 84-91.

[20] S. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, Fan, W. Dave, and P. K. Chan, "JAM: Java Agents for Meta-Learning over Distributed Databases", Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, ISBN 978-1-57735-027-9, 1997, pp. 74-81.

[21] J. R. Quinlan, "Induction of Decision Trees", Machine Learning, Vol. 1, 1986, pp. 81-106.

[22] P. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining", Pearson Addison Wesley, Boston, MA, 2006. pp. 160.

[23] G. Mathew, and Z. Obradovic, "Vocabularies in Collaboration Channels", Proceedings of the 6th International Conference on Collaborative Computing: Networking, Applications and Work Sharing, Chicago, IL, Oct 2010. ISBN: 978-963-9995-24-6

[24] SPECT Heart Data Set: available at – http://archive.ics.uci.edu/ml/datasets/SPECT+Heart

[25] I.H. Witten and E. Frank, "Data Mining: Pratical Machine Learning Tools and Techniques", Morgan Kaufmann, San Francisco, CA, 2nd edition, 2005.