DISTRIBUTED DATA MINING AND MINING MULTI-AGENT DATA

Vuda Sreenivasa Rao[1], Dr. S Vidyavathi[2]
Research Scholar CSIT Department[1], Associate Professor CSIT Department[2],
JNT University , Hyderabad Andhra Pradesh, India [1,2]
vudasrinivasarao@gmail.com[1] vidyasom@yahoo.co.in[2]

Abstract:- **The problem of distributed data mining is very important in network problems. Ina distributed environment (such as a sensor or IP network), one has distributed probes placed at strategic locations within the network. The problem here is to be able to correlate the data seen at the various probes, and discover patterns in the global data seen at all the different probes. There could be different models of distributed data mining here, but one could involve a NOC that collects data from the distributed sites, and another in which all sites are treated equally. The goal here obviously would be to minimize the amount of data shipped between the various sites — essentially, to reduce the communication overhead. In distributed mining, one problem is how to mine across multiple heterogeneous data sources: multi-database and multi-relational mining. Another important new area is *adversary data mining*. In a growing number of domains — email spam, counter-terrorism, intrusion detection/computer security, click spam, search engine spam, surveillance, fraud detection, shop bots, file sharing, etc. — data mining systems face adversaries that deliberately manipulate the data to sabotage them (e.g. make them produce false negatives). In this paper need to develop systems that explicitly take this into account, by combining data mining with game theory.**

Key words:- distributed data mining, NOC, multi-agent, multi-database, multi-relational mining, game theory.

1. INTRODUCTION:

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Distributed Data Mining (DDM) aims at extraction useful pattern from distributed heterogeneous data bases in order, for example, to compose them within a distributed knowledge base and use for the purposes of decision making. A lot of modern applications fall into the category of systems that need DDM supporting distributed decision making. Applications can be of different natures and from different scopes, for example, data and information fusion for situational awareness; scientific data mining in order to compose the results of diverse experiments and design a model of a phenomena, intrusion detection, analysis, prognosis and handling of natural and man-caused disaster to prevent their catastrophic development, Web mining ,etc. From practical point of view, DDM is of great concern and ultimate urgency.

A network operations center (or NOC, pronounced "knock") is one or more locations from which control is exercised over a computer, television broadcast, or telecommunications network. Large organizations may operate more than one NOC, either to manage different networks or to provide geographic redundancy in the event of one site being unavailable or offline. NOCs are responsible for monitoring the network for alarms or certain conditions that may require special attention to avoid impact on the network's performance. For example, in a telecommunications environment, NOCs are responsible for monitoring for power failures, communication line alarms (such as bit errors, framing errors, line coding errors, and circuits down) and other performance issues that may affect the network.

The increasing use of multi-database technology, such as computer communication The networks and distributed, federated and homogeneous multi-database systems, has led to the development of many multi-database systems for real world applications. For decision-making, large organizations need to mine the multiple databases distributed throughout their branches. The data of a company is referred to as internal data whereas the data collected from the Internet is referred to as external data. Although external data assists in improving the quality of decisions, it generates a significant challenge: how to efficiently identify quality knowledge from multi-databases [1], [2], [3]. Therefore, large companies may have to confront the multiple data-source problems

Multi-Relational Data Mining is inspired by the relational model [4, 5, 6]. This model presents a number of techniques to store, manipulate and retrieve complex and structured data in a database consisting of a collection of tables. It has been the dominant paradigm for industrial database applications during the last decades, and it is at the core of all major commercial database systems, commonly known as relational database management systems (RDBMS). A relational database consists of a

collection of named tables, often referred to as *relations* that individually behave as the single table that is the
subject of Propositional Data Mining. Data structures more complex than a single record are implemented by relating pairs of tables through so-called *foreign key relations*. Such a relation specifies how certain columns in one table can be used to look up information in corresponding columns in the other table, thus relating sets of records in the two tables. Structured individuals (graphs) are represented in a relational database in a distributed fashion. Each part of the individual (node) appears as a single record in one of the tables. All parts of the same class for all individuals appear in the same table. By following the foreign keys (edges), different parts can be joined in order to reconstruct an individual. In our search for patterns in the relational database, we will need to query individuals for certain structural properties. Relational database theory employs two popular languages for retrieving information from a relational database: relational algebra and the Structured Query Language (SQL). The former is primarily used in the theoretical settings, whereas the latter is primarily used in practical systems.

Many data mining applications, both current and proposed are faced with an active adversary. Problems range from the annoyance of spam to the damage of computer hackers to the destruction of terrorists. In all of these cases, statistical classification techniques play an important role in distinguishing the legitimate from the destructive. There has been significant investment in the use of learned classifiers to address these issues, from commercial spam filters to research programs such as those on intrusion detection [8] These problems pose a significant new challenge not addressed in previous research: The behavior of a class (the adversary) may adapt to avoid detection. A classifier constructed by the data miner in a static environment won't maintain its optimal performance for long, when facing an active adversary.

An intuitive approach to fight the adversary is to let the classifier adapt to the adversary's actions, either manually or automatically. Such a classifier was proposed in [1], which left open the following issue. The problem is that this becomes a never-ending game between the classifier and the adversary. Or is it never-ending? Will we instead reach an equilibrium, where each party is doing the best it can and has no incentive to deviate from its current strategy? If so, does this equilibrium give a satisfactory result for those using the classifier? Or does the adversary win?

Our approach is *not* to develop a learning strategy for the classifier to stay ahead of the adversary. We instead predict the end state of the "game"- an equilibrium state. We model the problem as a two-player game, where the adversary tries to maximize its return and the data miner tries to minimize the amount of misclassification. We examine under which conditions an equilibrium would exist, and provide a method to estimate the classifier performance and the adversary's behavior at such an equilibrium point (e.g., the players' equilibrium strategies).

Spam filtering is one motivating application. There are many examples of spam e-mails where words are modified to avoid spam filters. We could see that those transformations the adversary makes to defeat the data miner come with a cost: lower response rates. Combining the fact that the reward to the adversary decreases as they try to defeat the data miner, with the data miner's interest in avoiding false positives as well as false negatives, can lead us to equilibrium where both are best served by maintaining the status quo.

A game is a formal description of a strategic situation. Game theory is the formal study of decision-making where several players must make choices that potentially affect the interests of the other players.

The remaining sections of the paper are organized as follows. In Section II we describe the distributed data mining. In Section III we describe Multi Data base Mining In Section IV we describe Agent-based distributed data mining and open problems Strategy Section V A game Theoretic Model Section VI concludes the paper.

2. DISTRIBUTED DATA MINING:

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: most tools operate on a principal of gathering all data into a central site, then running an algorithm against that data (Figure 1). There are a number of applications that are infeasible under such a methodology, leading to a need for distributed data mining.

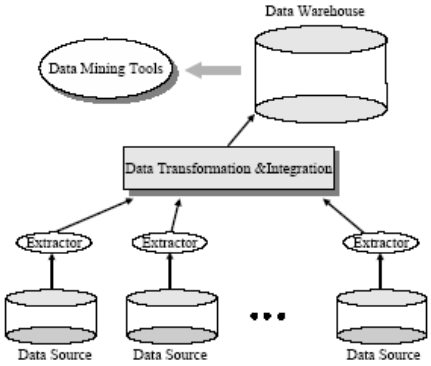
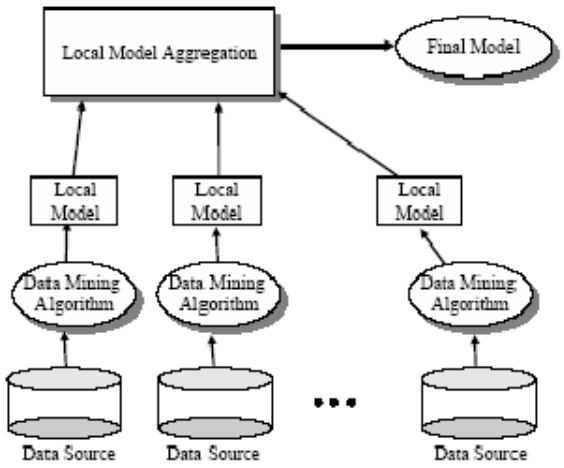


Figure 1. A data warehouse architecture

Distributed data mining (DDM) considers data mining in this broader context. As shown in figure(2), objective of DDM is to perform the data mining operations based on the type and availability of the distributed resources. It may choose to download the data sets to a single site and perform the data mining operations at a central location.



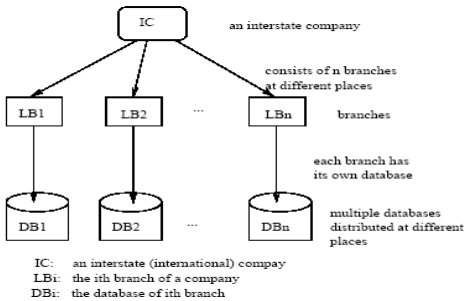
Distributed Data Mining Framework

Data mining is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers. Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. It discovers information within the data that queries and reports can't effectively reveal.

3. MULTI DATA BASE MINING:

Business, government and academic sectors have all implemented measures to computerize all, or part of, their daily functions [9]. An interstate (or international) company consists of multiple branches. The National Bank of Australia, for example, has many branches in different locations. Each branch has its own database, and the bank data is widely distributed and thus becomes a multi-database problem (see Fig. 2).

In Fig. 2, the top level is an interstate company (IC). This IC is responsible for the development and decision-making for the entire company. The middle level consists of n branches LB_1LB_2,----------LB_n. The bottom level consists of n local databases DB_1,DB_2,--------DB_n. of the n branches. Fig. 2 illustrates the structure of a two-level interstate company .In the real world, the structure of an interstate company is usually more complicated, and each branch may also have multi-level sub-branches. Many organizations have a pressing need to manipulate all the data from their different branches rapidly and reliably. This need is very difficult to satisfy when the data is stored in many independent databases, and the data is all of importance to an organization. Formulating and implementing queries requires data from more than one database. It requires knowledge of where all the data is stored, mastery of all the necessary interfaces and the ability to correctly combine partial results from individual queries into a single result. To respond to these demands, researchers and practitioners have intensified efforts on developing appropriate techniques for utilizing and managing multi-database systems. Hence, developing multi-database systems has become an important research area. Also, the computing environment is becoming increasingly widespread through the use of Internet and other computer communication networks. In this environment, it has become more critical to develop methods for building multi-database systems that combine relevant data from many sources and present the data in a form that is comprehensible for users, and provide tools that facilitate the efficient development and maintenance of information systems in a highly dynamic and distributed environment. One important technique within this environment is the development of multi-database systems. This includes managing and querying data from the collections of heterogeneous databases. While multi-database technology can support many multi database applications, it would be useful and necessary to mine these multi-databases to enable efficient utilization of the data. Thus, the development of multi-database mining is both a challenging and critical task.



4. AGENT-BASED DISTRIBUTED DATA MINING AND OPEN PROBLEMS STRATEGY:

Several systems have been developed for distributed data mining. These systems can be classified according to their strategy to three types; central learning, meta-learning, and hybrid learning.

4.1 Central learning strategy: is when all the data can be gathered at a central site and a single model can be build. The only requirement is to be able to move the data to a central location in order to merge them and then apply sequential DM algorithms. This strategy is used when the geographically distributed data is small. The strategy is generally very expansive but also more

accurate. The process of gathering data in general is not simply a merging step; it depends on the original distribution. For example, different records are placed in different sites, different attributes of the same records are distributed across different sites, or different tables can be placed at different sites, therefore when gathering data it is necessary to adopt the proper merging strategy. However, as pointed before this strategy in general is unfeasible [10]. Agent technology is not very preferred in such strategy.

4.2 Meta-learning strategy: it offers a way to mine classifiers from homogeneously distributed data. Meta-learning follows three main steps. The first is to generate base classifiers at each site using a classifier learning algorithms. The second step is to collect the base classifiers at a central site, and produce meta-level data from a separate validation set and predictions generated by the base classifier on it. The third step is to generate the final classifier (meta-classifier) from meta-level data via a combiner or an arbiter. Copies of classifier agent will exist or deployed on nodes in the network being used. Perhaps the most mature systems of agent-based meat-learning systems are: JAM system [11], and BODHI [11].

4.3 Hybrid learning strategy: is a technique that combines local and centralized learning for model building [12]; for example, Papyrus [13] is designed to support both learning strategies. In contrast to JAM and BODHI, Papyrus can not only move models from site to site, but can also move data when that strategy is desired. Papyrus is a specialized system which is designed for clusters while JAM and BODHI are designed for data classification.

The major criticism of such systems is that it is not always possible to obtain an exact final result, i.e. the global knowledge model obtained may be different from the one obtained by applying the one model approach (if possible) to the same data.

Approximated results are not always a major concern, but it is important to be aware of that. Moreover, in these systems hardware resource usage is not optimized. If the heavy computational part is always executed locally to data, when the same data is accessed concurrently, the benefits coming from the distributed environment might vanish due to the possible strong performance degradation. Another drawback is that occasionally, these models are induced from databases that have different schemas and hence are incompatible.

4.4 Overview of ADDM systems:

Applications of distributed data mining include credit card fraud detection system, intrusion detection system, and health insurance, security-related applications, distributed Clustering, market segmentation, sensor networks, customer profiling, evaluation of retail promotions, credit risk analysis, etc. These DDM application can be further enhanced with agents. ADDM takes data mining as a basis foundation and is enhanced with agents; therefore, this novel data mining technique inherits all powerful properties of agents and, as a result, yields desirable characteristics.

In general, constructing an ADDM system concerns three key characteristics: interoperability, dynamic system configuration, and performance aspects, discussed as follows. Interoperability concerns, not only collaboration of agents in the system, but also external interaction which allow new agents to enter the system seamlessly. The architecture of the system must be open and flexible so that it can support the interaction including communication protocol, integration policy, and service directory. Communication protocol covers message encoding, encryption, and transportation between agents, nevertheless, these are standardized by the Foundation of Intelligent Physical Agents (FIPA) 1 and are available for public access. Most agent platforms, such as JADE2 and JACK3, are FIPA compliant therefore interoperability among them are possible. Integration policy specifies how a system behaves when an external component, such as an agent or a data site, requests to enter or leave.

In relation with the interoperability characteristic, dynamic system configuration, that tends to handle a dynamic configuration of the system, is a challenge issue due to the complexity of the planning and mining algorithms. A mining task may involve several agents and data sources, in which agents are configured to equip with an algorithm and deal with given data sets. Change in data affects the mining task as an agent may be still executing the algorithm.

Lastly, performance can be either improved or impaired because the distribution of data is a major constraint. In distributed environment, tasks can be executed in parallel, in exchange, concurrency issues arise. Quality of service control in performance of data mining and system perspectives is desired, however it can be derived from both data mining and agents fields.

Next, we are now looking at the overview of our point of focus. An ADDM system can be generalized into a set of components and viewed as depicted in figure 3.1.We may generalize activities of the system into request and response, each of which involves a different set of components. Basic components of an ADDM system are as follows.

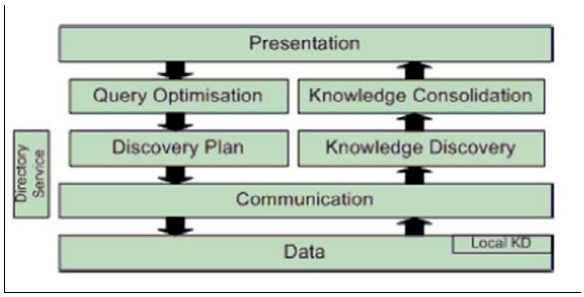


Fig. 3.1: Overview of ADDM systems

Data: Data is the foundation layer of our interest. In distributed environment, data can be hosted in various forms, such as online relational databases, data stream, web pages, etc., in which purpose of the data is varied.
Communication: The system chooses the related resources from the directory service, which maintains a list of data sources, mining algorithms, data schemas, data types, etc. The communication protocols may vary depending on implementation of the system, such as client-server, peer-to-peer, etc.
Presentation: The user interface (UI) interacts with the user as to receive and respond to the user. The interface simplifies complex distributed systems into user-friendly message such as network diagrams, visual reporting tools, etc. On the other hand, when a user requests for data mining through the UI, the following components are involved.
Query optimization: A query optimizer analyses the request as to determine type of mining tasks and chooses proper resources for the request. It also determines whether it is possible to parallelize the tasks, since the data is distributed and can be mined in parallel.
Discovery Plan: A planner allocates sub-tasks with related resources. At this stage, mediating agents play important roles as to coordinate multiple computing units since mining sub-tasks performed asynchronously as well as results from those tasks. On the other hand, when a mining task is done, the following components are taken place,
Local Knowledge Discovery (KD): In order to transform data into patterns which adequately represent the data and reasonable to be transferred over the network, at each data site, mining process may take place locally depending on the individual implementation.
Knowledge Discovery: Also known as mining, it execute the algorithm as required by the task to obtain knowledge from the specified data source.
Knowledge Consolidation: In order to present to the user with a compact and Meaningful mining result, it is necessary to normalize the knowledge obtained from various sources. The component involves a complex methodologies to combine knowledge/patterns from distributed sites. Consolidating homogeneous knowledge/ patterns is promising and yet difficult for heterogeneous case.

5. A GAME THEORETIC MODEL:

The adversarial learning scenario can be formulated as a two class problem, where class one (Π_1) is the "good" Class and class two (Π_2) is the "bad" class. n attributes would be measured from a subject coming from either classes. Denote the vector of attributes by
x = (x1; x2; : : : ; xn)/. Assume the attributes of a subject x would follow different distribution for different class. Let fi(x) be the probability density function of class
Π_i, i = 1; 2. The overall population is formed by combining the two classes. Let pi denote the proportion of class π_i in the overall population. Note p1 + p2 = 1. The distribution of the attributes x for the overall population could be considered as a mixture of the two distributions, with the density function written as
f(x) = p1f1(x) + p2f2(x).

Assume that the adversary can control the distribution of the "bad" class π_2. In other words, the adversary can modify the distribution by applying a transformation T to the attributes of a subject x that belong to π_2. Hence f2(x) would be changed into $f_2^T(x)$. Each such transformation would have a cost. At the same time, the adversary gains a profit when a "bad" instance (π_2) is classified as a "good" instance (π_1). We assume that the values of p1 and p2 will not be affected by the transformation, meaning that adversary would transform the distribution of π_2 but in a short time period would not significantly increase or decrease the amount of "bad" instances. Here we examine the case where a rational adversary and a rational data miner play the following game:

1. Given the initial distribution and density f(x), the adversary will choose a transformation T from the set of all feasible transformations S.

2. After observing the transformation T, the data miner Will create a classifier h.

Consider the case where data miner wants to minimize its (mis)classification cost. Define c_{ij} be the cost of classifying a subject x ϵ π_i given that x ϵ π_j . Given transformation T and the associated $f_2^T(x)$, the data miner uses a classifier h(x), and let L_i^h be the region where the instances are classified as π_i based on h(x), i = 1; 2. The expected cost of classification can be written as ([4]):

$$c(T,h) = \int_{L_1^h} \left[c_{11}p_1 f_1(x) + c_{12}p_2 f_2^T(x)\right] dx + \int_{L_2^h} \left[c_{21}p_1 f_1(x) + c_{22}p_2 f_2^T(x)\right] dx$$

Define the payoff function of data miner as

u2(T; h) = -c(T; h). Note that the value of c(T; h) is always positive assuming positive cij values. In order to maximize payoff u2, data miner needs to minimize c(T; h). Note that adversary will only profit from the "bad". Instances that are classified as "good". Also note that the transformation may change the adversary's profit of an instance that successfully passed the detection. Define

gT (x) as the profit function for a "bad". Instance x being classified as a "good" one, after the transformation T being applied. Define the adversary's payoff function of a transformation T given h as the following:

$$u_1(T,h) = \int_{L_1^h} \left(g^T(x) f_2^T(x) dx\right)$$

Within the vast literature of game theory, the *extensive game* provides a suitable framework for us to model the sequential structure of adversary and data miner's actions.

Specifically, the *Stackelberg game* with two players suits our need. In a Stackelberg game, one of the two players chooses an action a1 first and the second player, after observing the action of the first one, chooses an action a2.The game ends with payoffs to each player based on their payoff functions u1, u2 and a1, a2. In our model, we assume all players act rationally throughout the game. For the Stackelberg game, this implies that the second player will respond with the action a2 that maximizes u2 given the action a1 of the first player. The assumption of acting rationally at every stage of the game eliminates the Nash equilibrium with non-credible threats and creates an equilibrium called *sub game perfect equilibrium*. Further more, we assume that each player has perfect information about the other. Here in this context, .perfect information. Means that each player knows the other player's utility function. Further more, player two observes the a1 before choosing an action. In applications such as spam filtering, this is a reasonable assumption due to publicly available data.

5.1 Adversarial Learning Stackelberg Game:

A game G = (N;H; P; ui) *is called an Adversarial Learning Stackelberg Game if* N ={1,2}*, set of sequences* H ={ϕ, (T); (T; h)} *s t.* T ϵ S *and* h ϵ C*, where* S *is the set of all admissible transformations for adversary, and* C *is the set of all possible classification rules given a certain type of classifier. Function* P *assigns player to each sequence in* H *where* P(ϕ) = 1; P((T)) = 2 *(i.e., there existsan corresponding function* A *that assigns action space to each sequence in* H *where* A(ϕ) = S;A((T)) = C,A((T; h)) = ϕ*). Payoff functions* u1 *and* u2 *are defined as above.*

We use the minimum cost Bayesian classifier as an example to illustrate how we would solve for the sub game perfect equilibrium. First we will find the best response function for data miner given a transformation T. Using the population proportion pi of each class as the prior probabilities, and after observing T being applied to the .bad. class ($f_2^T(x)$), the optimal classification rule becomes:

$$h_T(x) = \begin{cases} \pi_1 & (c_{12} - c_{22}) p_2 f_2^T(x) \le (c_{21} - c_{11}) p_1 f_1(x) \\ \pi_2 & \text{otherwise} \end{cases}$$

hT (x) is the decision rule that minimizes the expected classification cost of the data miner. Given T, hT is the best response of data miner, i.e., R2(T) = hT . Then the adversary would find the transformation T that belongs to S which maximizes its profit, given the data miner would use hT = R2(T) defined above as its classification rule. Let

$$L_1^{hT} = \{x : (c_{12} - c_{22}) p_2 f_2^T(x) \le (c_{21} - c_{11}) p_1 f_1(x)\}$$

Be the region where the instances are classified as π_1 given hT . The adversary gain of applying transformation T is:

$$g_e(T) = u_1(T, R_2(T)) = E_{f_2^T}(I_{\{L_1^{hT}\}}(x) \times g^T(x))$$

Which is the expected value of the profit generated by the "bad"? Instances that would pass detection under transformation T. Therefore we can write the sub game perfect equilibrium as (T^*; hT*(x)), where T^* = argmax Tϵ S (ge(T)): (1) *Game theory ([9]) established that the solution of the above maximization problem is a sub game perfect equilibrium. Furthermore if the action space* S *is compact and* ge(T) *is continuous, the maximization problem has a solution .*

Another important aspect of the Adversarial Learning Stackelberg game and its sub game perfect equilibrium is that once an equilibrium point is reached, even if the game is repeated, both parties will not have an incentive to change their actions.

Theorem 1. *Let us assume that the adversarial learning Stackelberg game is played* n *times for finite* n*. Let us also assume that current* f(x) = p1f1(x) + p2f2(x) *is reached After playing the game* k *times and after adversary used* T^**, the sub game perfect equilibrium strategy defined by Equation 1, in the* kith *game. Also assume that parties will change their actions if they increase their payoff. This implies that adversary will not change* f2(x) *in the* jth *round where* k < j < n*. Similarly, the data miner will not change* hT^* (x) *in the* jth *round where* k < j < n*.*

The above formulation could accommodate any well defined set of transformations S, any appropriate distributions with densities f1(x) and f2(x), and any meaningful profit function gT (x). Next we present how above equations can be solved in practice.

5.2 Solving for the Equilibrium:

Since the domain of the integration L_1^{hr} for the adversary gain ge(T) is a function of the transformation T, finding an analytical solution to the maximization problem is very challenging. In addition, even calculating the integration analytically for a specific transformation is not possible for high dimensional data. Instead, we use

Monte Carlo integration technique that generally converts a given integration problem to computing an expected value. The adversary gain ge(T) can be written as:

$$g_e(T) = \int \left(I_{L_1^{hr}}(x) \times g^T(x) \right) f_2^T(x) dx$$

In the above formula, $I_{L1hr}(x)$ is the indicator function and returns 1 if x is classified into 1, else it returns 0. $f_2^T(x)$ is naturally a probability density function. Therefore $g_e(T)$ Could be calculated by sampling m points from $f_2^T(x)$, and taking the average of gT (x) for the sample points that satisfy

$$c_{12} - c_{22}) p_2 f_2^T(x) <= (c_{21} - c_{11}) p_1 f_1(x).$$

We consider stochastic search algorithms for finding an approximate solution for Equation 1. Especially, in our case, a stochastic search algorithm with the ability to converge to the global optimal solution is desired. To satisfy this goal, a simulated annealing algorithm is implemented to solve for the sub game perfect equilibrium. [2]

6. EXPERIMENTAL RESULTS:

It is interesting to see what the equilibrium strategies would become in response to different classification costs and transformation costs. Due to space limitations, we show only one set of experiments. In our setting a classifier changes when the classification cost matrix changes, and the adversary's gain is affected by the profit function under a transformation T. In this section we search for approximate equilibrium results under various classification cost matrices and profit functions. Table 1 contains the parameter values (rounded to 4 digits after the decimal point) for the Gaussian distributions. Notice there is no linear transformation T such that

$f2^T(x) = f1(x)$.

In our cost matrices, the correct classification costs are fixed to be 0, i.e., c11 = c22 = 0. We would modify the misclassification costs of classifying a ”bad” instance as “good” and a “good” instance as “bad” (Please note that cij is the cost of deciding x ϵ πi given that x ϵ πj . In our case, π2 is the “bad” class and π1 is the “good” class). Different profit reduction rates for the adversary are also considered.

Table 1. Mean and standard deviation for π1 and π2.

	π1		π_2	
Attribute	μ	σ	μ	σ
1	-0.7565	0.9597	-0.6461	0.7054
2	-0.7326	1.0415	-0.5403	0.8935
3	-1.6012	0.8545	-2.1507	0.7452
4	-2.8965	1.1542	-1.7248	0.9457
5	2.4552	0.9875	3.7255	1.2578
6	3.9976	1.4715	503245	1.2593

Table 2. Experiment Results

	a=0	a =0.2	a=0.7	Initial Gain
C_{21}/C_{12}=1	0.4950	0.2036	0.1958	0.1925
C_{21}/C_{12}=2	0.8420	0.3134	0.3134	0.3065
C_{21}/C_{12}=10	0.9820	0.6250	0.6235	0.6102

The adversary's gain is the expectation of the profit generated by a certain transformation T. Note that in the profit function, there are two parameters: the profit without transformation g, and the profit reduction rate a. In the experiments, without loss of generality, we fix g to be 1 and change the value of a. Combining the cost matrices and profit functions defined above, we performed nine experiments corresponding to combinations of the above. We restricted our search space to matrices with entries chosen from [-1; 1]. For each cost matrix of the data miner, the initial gain of the adversary (i.e., choosing the identity matrix as the transformation) and our experimental results are reported in Table 2.

The experiments show that for increasing profit reduction rate a > 0, simulated annealing cannot find a transformation within the search space that improves the gain of the adversary significantly better than the identity transformation. For a = 0, the adversary can increase its gain significantly by using transformation to defeat the filter.

The experiments identified two rather extreme equilibrium strategies. 1) The cost for misclassified .good. instances is much higher than for misclassified .bad. instances (i.e., c12p2 < c21p1), and there is no penalty for the adversary to perform transformations. The equilibrium strategy for the classifier is to pass most of the instances, good and bad alike; the adversary would transform its class (Π2) to have the similar distribution as the .good. class (Π1). Π2) Under equal misclassification costs, equal population size, and severe penalty for transformation, the classifier would minimize the total number of misclassified in instances; the adversary would not attempt to perform a transformation (i.e., perform the identity transformation). We could see when under more severe penalty, an adversary has less incentive to change.

7. CONCLUSION:

This paper has shown that the problem of distributed data mining and mining multi-database is challenging and pressing. We have defined a new process of multi-database mining for our system with game theory. In domains ranging from spam detection to counter-terrorism, classifiers have to contend with adversaries manipulating the data to produce false negatives. Research in this direction has the potential to produce DDM systems that are more robust to adversary

manipulations and require less human intervention to keep up with them. Many classification problems operate in a setting with active adversaries: while one party tries to identify the members of a particular class, the other tries to reduce the effectiveness of the classifier. Although this may seem like a never-ending cycle, it is possible to reach a steady-state where the actions of both parties stabilize. The game has equilibrium because both parties facing costs: costs associated with misclassification on the one hand, and for defeating the classifier on the other. By incorporating such costs in modeling, we can determine where such equilibrium could be reached, and whether it is acceptable to the data miner.

8.REFERENCES:

[1] X.Wu and S. Zhang, Synthesizing High-Frequency Rules from Different Data Sources, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, March/April 2003: 353-367.

[2] C. Zhang and S. Zhang, *Association Rules Mining: Models and Algorithms*. Springer- Verlag Publishers in Lecture Notes on Computer Science, Volume 2307, p. 243, 2002.

[3] N. Zhong, Y. Yao, and S. Ohsuga, Peculiarity oriented multi-database mining. In: Proceedings of PKDD, 1999: 136-146.

[4]. Date, C. *An Introduction to Database Systems, Volume I*, The Systems Programming Series, Addison-Wesley, 1986.

[5]. Ullman, I.D. *Principles of Databases and Knowledge-Based Systems,* Volume I, Computer Science Press, 1988

[6]. Ullman, J., Widom, J., *A First Course in Database Systems*, Prentice Hall, 2001

[7] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A
scalable comparison-shopping agent for the World-Wide Web. In Proceedings of the First International Conference on Autonomous Agents, pages 39{48, Marina del Rey, CA, 1997. ACM Press.

[8] T. Fawcett. \In vivo" spam _ltering: A challenge problem for KDD. SIGKDD Explorations, 5(2):140{148, 2003.

5] T. Fawcett and F. Provost. Adaptive fraud detection.
Data Mining and Knowledge Discovery, 1(3):291{316,

[9] L. Guernsey. Retailers rise in Google rankings as rivals cry foul. New York Times, November 20, 2003.

[10] D. Jensen, M. Rattigan, and H. Blau. Information awareness: A prospective technical assessment. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 378{387, Washington, DC, 2003. ACM Press.

[11] B. Krebs. Online piracy spurs high-tech arms race.
Washington Post, June 26, 2003.

[12] B. Lloyd. Been gazumped by Google? Trying to make sense of the \Florida" update. Search Engine Guide,
November 25, 2003.

[13] M. V. Mahoney and P. K. Chan. Learning Non stationary models of normal network trac for detecting novel attacks. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 376{385, Edmonton, Canada, 2002. ACM Press.

[14] P. Robertson and J. M. Brady. Adaptive image analysis for aerial surveillance. IEEE Intelligent Systems, 14(3):30{36, 1999.

[15] T. Senator. Ongoing management and application of
Discovered knowledge in a large regulatory organization: A case study of the use and impact of NASD regulation's advanced detection system (ADS). In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 44{53, Boston, MA, 2000. ACM Press.

[16] A. Hurson, M. Bright, and S. Pakzad, Multidatabase systems: an advanced solution for global information sharing. IEEE Computer Society Press, 1994.

[17] H. Liu, H. Lu, and J. Yao, Identifying Relevant Databases for Multidatabase Mining. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1998: 210–221.

[18] J. Yao and H. Liu, Searching Multiple Databases for Interesting Complexes.
In: Proc. of PAKDD, 1997: 198-210.

Vuda .Srinivasarao received the M.Tech degree in Computer Science & Engg from the Satyabama University, in 2007. . He is research scholar in CSIT Department, JNT University Hyderabad Andhra Pradesh, India. His research interests include Network Security, Cryptography, and Data Mining & Artificial Intelligence.

Dr. S Vidyavathi received her PhD degree from IIT Mumbai, She is currently working as Associate Professor in CSIT Department, JNT University, Andhra Pradesh, India.