SCIKD: Safeguarding Classified Information from Knowledge Discovery

Seunghyun Im[1]
[1]University of North Carolina
Department of Computer Science
Charlotte, N.C. 28223, USA
Email: sim@uncc.edu

Zbigniew W. Ras[1,2]
[2]Polish-Japanese Institute of IT
Department of Computer Science
Koszykowa 86, 02-008 Warsaw, Poland
Email: ras@uncc.edu

Agnieszka Dardzińska[3,1]
[3]Bialystok Technical University
Department of Mathematics
15-351 Bialystok, Poland
Email: adardzin@uncc.edu

***Abstract*— In this paper, we present a method that allows us to reduce disclosure risk of confidential data from knowledge discovery. In particular, the proposed method protects confidential data against the null value imputation algorithm Chase with minimum amount of additional data hiding. The method is designed to identify the set of values that can remain unchanged without examining all possible combinations of the values. The concept introduced in this paper can be used to protect multiple confidential attributes in information systems.**

I. Introduction

The knowledge discovered by data mining has provided powerful solutions to non-trivial problems in a wide variety of domains. The distinctive capability of extracting hidden knowledge from large volumes of data, however, brought up security issues as the concern about privacy and data security grows. In particular, disclosure of confidential data via hidden value reconstruction by knowledge discovery has received increasing attention in recent years. The notion of the hidden attribute reconstruction by knowledge discovery was introduced in [3] where the rules in Knowledge Base (*KB*) can be used to reconstruct sensitive data hidden from an information system as shown in Figure 1.

When trade-offs are not considered, protection of confidential data against knowledge discovery is relatively simple. In general, some degree of data loss is almost inevitable to block reconstruction of confidential data by knowledge inferences, and the chances of reconstruction drops as the amount of data loss continues to grow. In reality, however, additional requirements are typically imposed on data protection schemes, such as maximum preservation of original data and valid rules, to enhance information availability.

In this paper, we assume that one or more attributes in an information system contain confidential data that have to be protected, and the system is part of a Knowledge Discovery System (*KDS*) which provides a set of rules as a *KB*. Clearly, we have to be certain that the values of confidential attribute can not be predicted from the available data and *KB* by *Chase*, distributed *Chase* [4] or any other null value imputation method while minimizing the changes in the original information system. In pursue of such requirements, we propose a protection method named as *SCIKD*. The method identifies transitive closure of attribute values involved in confidential data reconstruction, and uses the result to identify the maximum number of attribute values that can remain unchanged.

II. Hidden Value Reconstruction by Null Value Imputation

We briefly provide some additional background on a null value imputation algorithm *Chase* [4]. Assume that $S = (X, A, V)$ is an information system [4], where X is a set of objects, A is a finite set of attributes, and V is a finite set of their values. In particular, we say that $S = (X, A, V)$ is an incomplete information system of type λ if the following three conditions hold:

- $a_S(x)$ is defined as $\{(a_i, p_i) : a_i \in V_a \wedge 1 \leq i \leq m\}$ for any $x \in X$, $a \in A$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(a_i, p_i) : 1 \leq i \leq m\}) \rightarrow \sum_{i=1}^{m} p_i = 1]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(a_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\forall i)(p_i \geq \lambda)]$.

Data incompleteness is understood by allowing a set of weighted attribute values to be a value of an attribute. Suppose an attribute d in S contains confidential data, and it is to be hidden. For this purpose, we construct $S_d = (X, A, V)$ to replace S, where:

- $a_S(x) = a_{S_d}(x)$, for any $a \in A - \{d\}, x \in X$,
- $d_{S_d}(x)$ is undefined, for any $x \in X$,
- $d_S(x) \in V_d$,

and user queries are responded by S_d (see table 2). The question is whether hiding the attribute d is enough. The definition of d may already be available in local *KB*, or a request can be sent to some of its remote sites if the *KDS* is a Distributed Information System containing S. Now, assume that such definitions (or rules) are defined as $R_d = \{r = [t \rightarrow d_c] : c \in d\} \subseteq KB$. (In this paper we assume that rules are generated by $ERID(S, \lambda_1, \lambda_2)$, where λ_1, λ_2 are thresholds for minimum support and minimum confidence, correspondingly. *ERID* is the algorithm for discovering rules from incomplete information systems, presented by Dardzińska and Raś in [5] and used as a part of *Chase* algorithm in [7]. The definitions

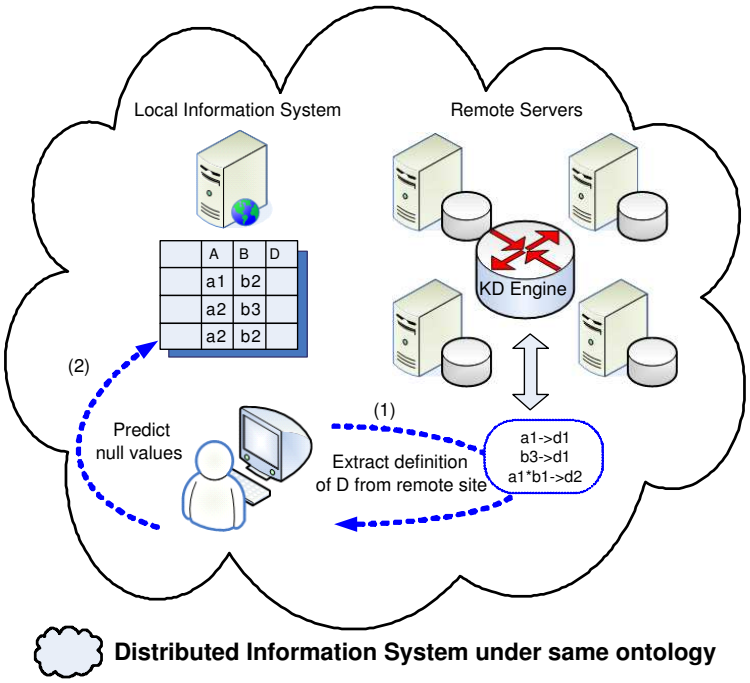

Fig. 1. An example of confidential data reconstruction

stored in the knowledge base *KB* can be used by *Chase* algorithm to replace missing values for objects at S_d using the following method.

Suppose that $d(x) = d_1$ for x in S. If x does not support any rule predicting d_1, it is ruled out because d_1 cannot be reconstructed precisely. If the object supports a set of rules, it can be either $R_s(x) = \{r_1 = [t_1 \rightarrow d_1], r_2 = [t_2 \rightarrow d_1], ..., r_k = [t_k \rightarrow d_1]\}$ where the rules imply a single decision attribute value, or $\{r_1 = [t_1 \rightarrow d_1], r_2 = [t_2 \rightarrow d_2], ..., r_k = [t_k \rightarrow d_k]\}$ where the rules imply multiple decision values. If we denote support and confidence of rule r_i as $[s_i, c_i]$, and the weight of each attribute value in a slot as p_i, the confidence for attribute value $d' \in V_d$ for x driven by KB is defined as [2]

$$Conf_{S_d}(d', x, KB) = \frac{\sum\{[p_i] \cdot s_i \cdot c_i : [1 \leq i \leq k] \wedge [d' = d_i]\}}{\sum\{[p_i] \cdot s_i \cdot c_i : 1 \leq i \leq k\}}$$

When $d(x) = d_j$ and λ is the threshold for minimal confidence in attribute values describing objects in S_d, there are three cases in hiding attribute values.

1) if $Conf_{S_d}(d_j, x, KB) \geq \lambda$ and $(\exists d \neq d_j)[Conf_{S_d}(d, x, KB) \geq \lambda]$, we do not have to hide any additional slots for x.
2) if $Conf_{S_d}(d_j, x, KB) \geq \lambda$ and $(\forall d \neq d_j)[Conf_{S_d}(d, x, KB) < \lambda]$, we have to hide additional slots for x.
3) if $Conf_{S_d}(d_j, x, KB) < \lambda$ and $(\exists d \neq d_j)[Conf_{S_d}(d, x, KB) \geq \lambda]$, we do not have to hide additional slots for x.

Clearly, $d_S(x)$ can be equal to $d_{Chase(S_d)}(x)$ for a number of objects in X. If this is the case, additional values of attributes for all these objects should be hidden.

III. Finding Minimum Number of Attribute Values to be Hidden

We present an algorithm which protects values of a hidden attribute over null value imputation *Chase*. Suppose we have an information system S as shown in Table I. S is transformed to S_d by hiding the confidential attribute d as shown in Table II. The rules in the knowledge base *KB* are summarized in Table III. For instance $r_1 = [b_1 \cdot c_1 \rightarrow a_1]$ is an example of a rule belonging to *KB*.

To describe the algorithm, first we define the following sets,

- $\alpha(x)$, the set of attribute values used to describe x in S_d
- $\alpha(t)$, the set of attribute values used in t, where t is their conjunction
- $R(x) = \{(t \rightarrow d) : \alpha(t) \subseteq \alpha(x)\} \subseteq KB$, the set of rules in *KB* where the attribute values used in t are contained in $\alpha(x)$
- $\beta(x) = \cup\{\alpha(t) \cup \{d\} : [t \rightarrow d] \in R(x)\}$.

In our example $R(x_1) = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}\}$, and $\beta(x_1) = \{a_1, b_1, c_1, d_1, e_1, f_1, g_1\}$. Clearly, by using the procedure described in section 2, d_1 replaces the hidden slots by rules from $\{r_8, r_9, r_{10}\}$. In addition, other rules from $R(x_1)$ also predict attribute values listed in $\{t_8, t_9, t_{10}\}$. These interconnections often build up a complex chain of inferences. The task of blocking such inference chains and identifying the minimal set of concealing values is not straightforward.

X	*A*	*B*	*C*	*D*	*E*	*F*	*G*
x_1	$(a_1,\frac{2}{3})(a_2,\frac{1}{3})$	b_1	c_1	d_1	e_1	f_1	g_1
x_2	$(a_2,\frac{2}{5})(a_3,\frac{3}{5})$	$(b_1,\frac{1}{3})(b_2,\frac{2}{3})$		d_2	e_1	f_2	
x_3	a_1	b_2	$(c_1,\frac{1}{2})(c_3,\frac{1}{2})$	d_1	e_3	f_2	
x_4	a_3		c_2	d_1	$(e_1,\frac{2}{3})(e2,\frac{1}{3})$	f_2	
x_5	$(a_1,\frac{2}{3})(a_3,\frac{1}{3})$	$(b_1,\frac{1}{2})(b_2,\frac{1}{2})$	c_2	d_1	e_1	f_2	g_1
x_6	a_2	b_2	c_3	d_1	$(e_2,\frac{1}{3})(e_3,\frac{2}{3})$	f_3	
x_7	a_2	b_1	$(c_1,\frac{1}{3})(c_2,\frac{2}{3})$		e_2	f_3	
			.				
			.				
x_i	$(a_3,\frac{1}{2})(a_4,\frac{1}{2})$	b_1	c_2		e_3	f_2	

TABLE I
Information System S

X	*A*	*B*	*C*	*D*	*E*	*F*	*G*
x_1	$(a_1,\frac{2}{3})(a_2,\frac{1}{3})$	b_1	c_1		e_1	f_1	g_1
x_2	$(a_2,\frac{2}{5})(a_3,\frac{3}{5})$	$(b_1,\frac{1}{3})(b_2,\frac{2}{3})$			e_1	f_2	
x_3	a_1	b_2	$(c_1,\frac{1}{2})(c_3,\frac{1}{2})$		e_3	f_2	
x_4	a_3		c_2		$(e_1,\frac{2}{3})(e2,\frac{1}{3})$	f_2	
x_5	$(a_1,\frac{2}{3})(a_3,\frac{1}{3})$	$(b_1,\frac{1}{2})(b_2,\frac{1}{2})$	c_2		e_1	f_2	g_1
x_6	a_2	b_2	c_3		$(e_2,\frac{1}{3})(e_3,\frac{2}{3})$	f_3	
x_7	a_2	b_1	$(c_1,\frac{1}{3})(c_2,\frac{2}{3})$		e_2	f_3	
			.				
			.				
x_i	$(a_3,\frac{1}{2})(a_4,\frac{1}{2})$	b_1	c_2		e_3	f_2	

TABLE II
Information System S_d

Rule	*A*	*B*	C	D	E	F	G
r_1	(a_1)	b_1	c_1				
r_2	(a_1)		c_1			f_1	
r_3		(b_1)	c_1				
r_4		(b_1)			e_1		
r_5	a_1		(c_1)			f_1	
r_6	a_1		c_1		(e_1)		
r_7			(c_1)		e_1		g_1
r_8	a_1		c_1	(d_1)			
r_9		b_1	c_1	(d_1)			
r_{10}				(d_1)		f_1	

TABLE III
Rules contained in *KB*. Values in parenthesis are decision values

Let us consider the following example. Suppose we have $\{r_1 = [a_1 \cdot b_1 \rightarrow d_1], r_2 = [b_1 \cdot c_1 \rightarrow d_1], r_3 = [b_1 \cdot e_1 \rightarrow d_1]\}$, all inferencing d_1. In this case, b_1 is covered by 3 rules, and elimination of it will ensure the protection. However, if there were 3 other rules $\{h_1 \rightarrow b_1, i_1 \rightarrow h_1, k_1 \rightarrow j_1\}$, additional values $\{h_1, i_1, k_1\}$ have to be hidden, and b_1 was not the best choice. In general, if a large number of attributes and rules exist, overlap based approach often produces a large and complex graph as we try to trace connections from the top to the bottom. Another issue is that the order we eliminate values of attributes may have high impact on the final result. For example, hiding values in the order of $c_1 \Rightarrow f_1 \Rightarrow a_1$ and $c_1 \Rightarrow a_1 \Rightarrow f_1$ may produce different results because attribute value set $\{c_1, f_1\}$ removes the inference to d_1, while $\{c_1, a_1\}$ cannot remove it and we have to hide b_1 again.

To reduce the complexity and minimize the set of hidden values, a bottom up approach has been adapted. We check the values that will remain unchanged starting from a singleton set containing attribute value a by using transitive closure [11] (if $a \rightarrow b \wedge b \rightarrow c$ then $a \rightarrow c$, which gives us the set $\{a, b, c\}$), and

increase the initial set size as much as possible. This approach automatically rules out any superset of must-be-hidden values, and minimizes the computational cost. The justification of this is quite simple. Transitive closure has the property that the superset of a set s also contains s. Clearly, if a set of attribute values predicts d_1, then the set must be hidden regardless of the presence/abscence of other attribute values.

To outline the procedure, we start with a set $\beta(x)$ for the object x_1 which construction is supported by 10 rules from *KB*, and check the transitive closure of each singleton subset $\delta(x)$ of that set. If the transitive closure of $\delta(x)$ contains classified attribute value d_1, then $\delta(x)$ does not sustain, it is marked, and it is not considered in later steps. Otherwise, the set remains unmarked. In the second iteration of the algorithm, all two-element subsets of $\beta(x)$ built only from unmarked sets are considered. If the transitive closure of any of these sets does not contain d_1, then such a set remains unmarked and it is used in the later steps of the algorithm. Otherwise, the set is getting marked. If either all sets in a currently executed iteration step are marked or we have reached the set $\beta(x)$, then the algorithm stops. Since only subsets of $\beta(x)$ are considered, the number of iterations will be usually not large.

So, in our example the following singleton sets are considered:

$\{a_1\}^+ = \{a_1\}$ unmarked
$\{b_1\}^+ = \{b_1,\}$ unmarked
$\{c_1\}^+ = \{a_1, b_1, c_1, e_1, d_1\} \supseteq \{d_1\}$ marked *
$\{e_1\}^+ = \{b_1, e_1\}$ unmarked
$\{f_1\}^+ = \{d_1, f_1\} \supseteq \{d_1\}$ marked *
$\{g_1\}^+ = \{g_1\}$ unmarked

Clearly, c_1 and f_1 have to be concealed. The next step is to build sets of length 2 and determine which of them can sustain. We take the union of two sets only if they are both unmarked and one of them is a singleton set.

$\{a_1, b_1\}^+ = \{a_1, b_1\}$ unmarked
$\{a_1, e_1\}^+ = \{a_1, b_1, e_1\}$ unmarked
$\{a_1, g_1\}^+ = \{a_1, g_1\}$ unmarked
$\{b_1, e_1\}^+ = \{b_1, e_1\}$ unmarked
$\{b_1, g_1\}^+ = \{b_1, g_1, e_1\}$ unmarked
$\{e_1, g_1\}^+ = \{a_1, b_1, d_1, d_1, e_1, g_1\} \supseteq \{d_1\}$ marked*

Now we build 3-element sets from previous sets that have not been marked.

$\{a_1, b_1, e_1\}^+ = \{a_1, b_1, e_1\}$ unmarked
$\{a_1, b_1, g_1\}^+ = \{a_1, b_1, g_1\}$ unmarked
$\{b_1, e_1, g_1\}^+$ is not considered as a superset of $\{e_1, g_1\}$ which was marked.

We have $\{a_1, b_1, e_1\}$ and $\{a_1, b_1, g_1\}$ as unmarked sets that contain the maximum number of elements and do not have the transitive closure containing d. In a similar way, we compute the maximal sets for any object x_i.

IV. Algorithm SCIKD: Safeguarding Classified Information from Knowledge Discovery

We are ready to present more precise description of the algorithm for identifying the minimal number of attribute values in S_D which have to be additionally hidden from users in order to guarantee that attribute D cannot be reconstructed through knowledge discovery. So, let us assume that *KB* is a knowledge base for S_D and that an attribute $D \in A$ needs to be hidden.

SCIKD(S_D, KB)

```
begin
  i := 1;
  while i ≤ l do
  begin
    for all v ∈ α(x_i) do Mark(v) := F;
    for all v ∈ α(x_i) do
    begin
      R(x_i) = {r ∈ KB : (∃d)[r = v → d]};
      γ(x_i) := D(x_i);
      α1(x_i, v) := {v};
      β(x_i, v) = α_1(x_i, v) ∪ {d : [v → d] ∈ R(x_i)};
      while γ(x_i) ∉ β(x_i, v) and α_1(x_i, v) ≠ β(x_i, v) do
      begin
        α_1(x_i, v) := β(x_i, v);
        R(x_i) = {r ∈ KB : (∃t ⊂ α_1(x_i, v))[r = t → d]};
        β(x_i, v) = α_1(x_i, v) ∪ {d : (∃t)([t → d] ∈ R(x_i))};
      end
      if γ(x_i) ∈ β(x_i, v) then Mark(v) := T;
    end
    j:= 2;
    while j ≤ k_i − 1 do
    begin
      for each w ⊂ α(x_i) such that [card(w) = j
      and all subsets of w are unmarked] do
      begin
        α_1(x_i, w) := w;
        β(x_i, w) = α_1(x_i, w) ∪ {d : (∃t ⊂ w)[t → d] ∈ R(x_i)};
        while γ(x_i) ∉ β(x_i, w) and α_1(x_i, w) ≠ β(x_i, w) do
        begin
          α_1(x_i, w) := β(x_i, w);
          R(x_i) = {r ∈ KB : (∃t ⊂ α_1(x_i, w))[r = t → d]};
          β(x_i, w) = α_1(x_i, w) ∪ {d : (∃t)([t → d] ∈ R(x_i))};
        end
        if γ(x_i) ∈ β(x_i, w) then Mark(w) := T;
      end
    end
    i := i + 1
  end
end
```

The algorithm presented here is a simplified version of the system *SCIKD* which is implemented and tested. Namely, its

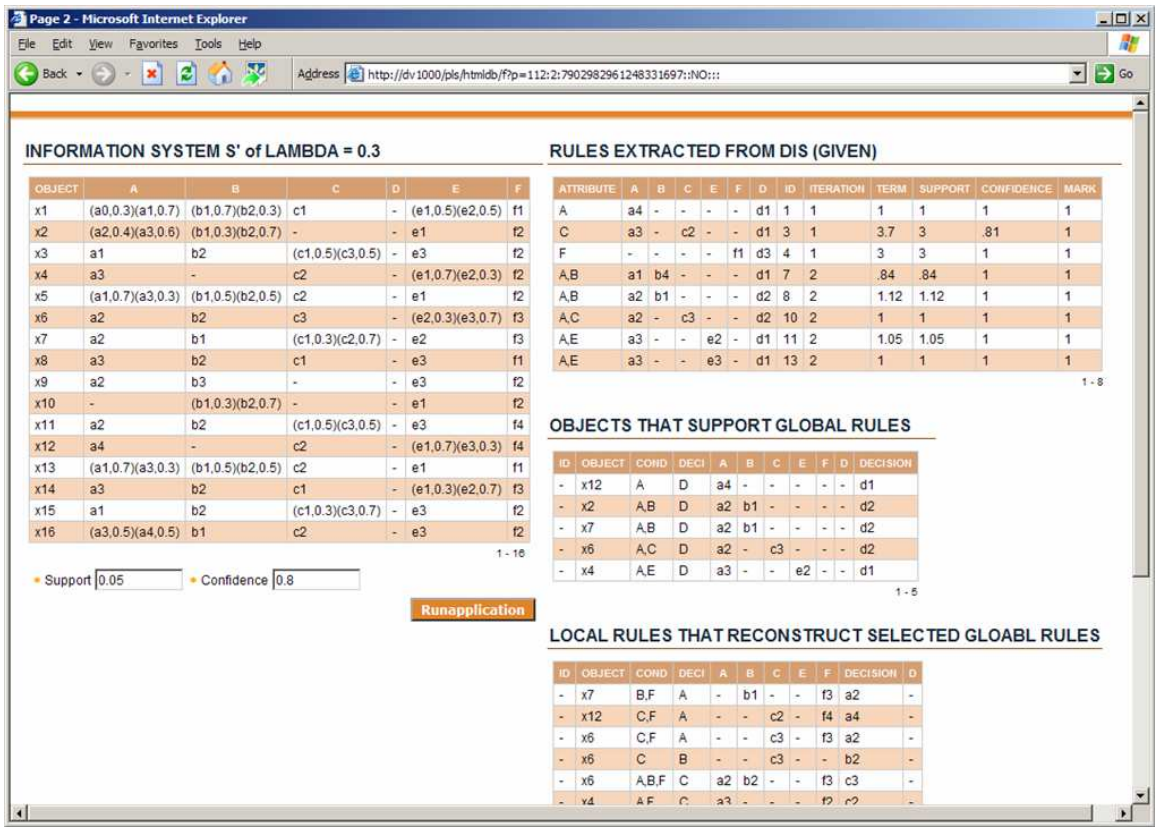

Fig. 2. Sample Interface for $SCIKD$

implemented version allows possible rules to be used in *KB*. Also, if one of the possible attribute values to be placed in a hidden slot has a confidence below a threshold value set up by a user, then this attribute value is not considered in further steps of the algorithm. This approach is similar to the one followed in the paper [4].

V. Experiment and Conclusion

We implemented the method on a PC running Windows XP and Oracle database version 10g. The code was written in PL/SQL language with PL/SQL Developer version 6. HTML DB and some additional Javascript have been used to create a graphical user interface.

In order to evaluate the efficiency of the proposed method, we compared the total number of additionally hidden attribute values with that obtained by the method described in [2]. The compared method uses a top-down approach that the order of elimination of attribute values in x_i is determined by the degree of overlap of supported rules. The sampling data table containing 4,000 objects with 10 attributes was extracted randomly from a complete database describing personal income reported in the Census data [1]. The data table was randomly partitioned into 4 tables that each have 1,000 tuples. One of these tables is called a client and the remaining 3 are called servers. Now, we hide all the values of one attribute that includes income data in the client. From the servers, 13 rules are extracted which are used to describe values of hidden attribute by *Distributed Chase* algorithm, and 75 rules are extracted from the client which are used to describe the values of remaining attributes by *Local Chase* algorithm. All these rules are generated using *ERID* and stored in KB of the client.

1014 attribute values (10.14% of the total number of attribute values in client table) are additionally hidden when overlap based method is applied, and 739 (7.39%) attribute values are additionally hidden by the method presented in this paper. Clearly, the level of improvement in the number of additional hidden value over previous method, as well as the percentage point of hidden values, will not be the same when different set of rules and information system are used. However, the proposed method reduces the number of hidden attribute values by identifying all combinations of values that predict the hidden data without examining all possible supersets. In addition, the method can easily be used to protect two or more confidential attributes in an information system. In this case, a set of attribute values in x_i should be hidden if the closure of the set contains any of the classified data.

Acknowledgment

This research was supported, in part, by the NIH Grant No. 5G11HD38486-06.

REFERENCES

[1] S. Hettich, C. L. Blake and, C. J. Merz, *UCI Repository of machine learning databases*, 3rd ed., *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[2] S. Im, Z. W. Raś, *Ensuring Data Security against Knowledge Discovery in Distributed Information System*, Proceedings of 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Springer-Verlag, 2005, 548-557.

[3] Z. W. Raś, A. Dardzińska, *Data security and null value imputation in distributed information systems*, Advances in Soft Computing, Springer-Verlag, 2005, 133-146.

[4] A. Dardzińska, Z. W. Raś, *CHASE-2: Rule based chase algorithm for information systems of type lambda*, Postproceedings of the Second International Workshop on Active Mining (AM'2003), Maebashi City, Japan, (Eds. S. Tsumoto et al.), LNAI, No. 3430, Springer, 2005, 258-270.

[5] A. Dardzińska, Z. W. Raś, *Extracting Rules from Incomplete Decision Systems: System ERID*, Foundations and Novel Approaches in Data Mining, (Eds. T.Y. Lin, S. Ohsuga, C.J. Liau, X. Hu), Advances in Soft Computing, Springer, 2005, 143-154.

[6] A. Dardzińska, Z. W. Raś, *Chasing Unknown Values in Incomplete Information Systems*, Proceedings of ICDM 03 Workshop on Foundations and New Directions of Data Mining, IEEE Computer Society, 2003, 24-30.

[7] Z. W. Raś, A. Dardzińska, *Query Answering based on Collaboration and Chase*, Proceedings of 6th International Conference On Flexible Query Answering Systems, LNAI, No. 3055, Springer, 2004, 125-136.

[8] Z. W. Raś, *Dictionaries in a distributed knowledge-based system*, in *Concurrent Engineering: Research and Applications*, Proceedings of CE'94 in Pittsburgh, Penn., Concurrent Technologies Corporation, 1994, 383-390.

[9] Z. W. Raś, A. Dardzińska, *Ontology Based Distributed Autonomous Knowledge Systems*, Information Systems International Journal, Vol. 29, No. 1, Elsevier, 2004, 47-58.

[10] Z. W. Raś, S. Joshi, *Query approximate answering system for an incomplete DKBS*, Fundamenta Informaticae Journal, Vol. 30, No. 3/4, IOS Press, 1997, 313-324.

[11] S. Abiteboul, R. Hull and, V. Vianu, *Foundations of Databases*, Addison Wesley, 1995.