**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BIRLA INSTITUTE OF TECHNOLOGY**



PRIVACY PRESERVING

DATA MINING

A project report

Submitted in the fulfillment of the requirement for

The award of the Degree of

Bachelors of Engineering In

**COMPUTER SCIENCE**

BY: PROJECT GUIDE:

**SAGAR BAVER** **DR. SHASHANK PUSHKAR**

BE/10413/2012

**SURAJ KAMAL GUPTA**

BE/10306/2012

**UJJWAL SACHDEVA**

BE/10305/2012

**DECLARATION CERTIFICATE**

This is to certify that the contents of the project entitled “PRIVACY PRESERVING DATA MINING**”** is a bonafide work carried out by **Sagar Baver, Suraj Kamal Gupta and Ujjwal Sachdeva** under my supervision and guidance in fulfillment of the requirements for the degree of Bachelor of Engineering in Computer Science of Birla Institute of Technology Mesra, Ranchi.

The contents of this project report have not been submitted earlier for the award of any other degree or certificate. I hereby commend this work.

Date:

Dr. Shashank Pushkar

Assistant Professor, Dept. of CSE

Birla Institute of Technology

Mesra, Ranchi-835215

Head Dean

Department of CSE (Undergraduate Studies)

BIT Mesra BIT Mesra

**CERTIFICATE OF APPROVAL**

The project work entitled “PRIVACY PRESERVING DATA MINING”, is carried out and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the project report for the purpose for which it is submitted.

Internal Examiner External Examiner

Head of the Department

Computer Science and Engineering

Birla Institute of Technology, Mesra

Ranchi-835215

**ACKNOWLEDGEMENTS**

We would like to thank all the people who helped and supported us in writing the research project.

We would like to express our gratitude to our project guide,Dr. Shashank Pushkar, for his constant motivation and material references for working on this project. We are grateful to have shared his experience in this field.

We would also want to thank all the other faculties, who have been our lecturers on various fields crucial to this project.

Last but not the least, we are also grateful to our friends and all the long discussions with them which has contributed to our project’s development.

**ABSTRACT**

In the research of privacy-preserving data mining, we address issues related to extracting knowledge from large amounts of data without violating the privacy of the data owners.

In this study, we first introduce an integrated baseline architecture, design principles, and implementation techniques for privacy-preserving data mining systems. We then discuss the key components of privacy-preserving data mining systems which include three protocols: data collection, inference control, and information sharing.

In privacy-preserving data mining (PPDM), a widely used method for achieving data mining goals while preserving privacy is based on k-anonymity. This method, which protects subject-specific sensitive data by anonymizing it before it is released for data mining, demands that every tuple in the released table should be indistinguishable from no fewer than k subjects.

We present and compare strategies for realizing these protocols. Theoretical analysis and experimental evaluation show that our protocols can generate accurate data mining models while protecting the privacy of the data being mined.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TOPIC** | **PAGE NO.** |
| 1. Introduction    1. Baseline Architecture    2. Design Principle    3. Basic Strategy | 9 |
| 1. Privacy Preserving Data Mining (PPDM)   2.1 Sample Case | 12 |
| 1. Suppression | 13 |
| 4. Generalization | 14 |
| 5. Feature set partitioning | 15 |
| 6. Genetic algorithm based search  6.1 Advantages and Drawbacks of GA | 16 |
| 7. Data Mining Privacy By Decomposition (DMPD)  7.1 DMPD Terminology  7.1.1 Classification Problem  7.1.2 Features  7.1.3 Quasi Identifiers  7.1.4 Fitness Value Function  7.1.5 Crossover Operation  7.1.6 Mutation Operation | 17 |
| 8. K – Anonymity Property  8.1 Sample Case | 20 |
| 9. Cuckoo Search  9.1 Characteristics of Cuckoo Search  9.2 Three Idealized Rules | 23 |
| 10. Feature Set Selection Using Cuckoo Search  10.1 The Algorithm | 24 |
| 11. Results and Analysis  11.1 GA based DMPD Performance  11.2 Performance of CS Algorithm  11.3 Comparative study | 25 |
| 12. Conclusion | 31 |
| 13. Bibliography | 32 |

**LIST OF FIGURES AND TABLES**

|  |  |  |
| --- | --- | --- |
| **SL NO.** | **DESCRIPTION** | **PAGE NO.** |
| 1. | Baseline Architecture Figure | 10 |
| 2. | K – anonymity Model | 12 |
| 3. | Suppression technique Table | 14 |
| 4. | GA Process Diagram | 16 |
| 5. | Non – anonymized database | 20 |
| 6. | Anonymized database of Table 5 | 22 |
| 7. | Generation of Initial Population in CS Algorithm | 24 |
| 8. | Creating the next population using levy flight in CS Algorithm | 24 |
| 9. | Comparing Fitness Values in CS Algorithm. | 25 |
| 10. | UCI Adult Repository Dataset | 26 |
| 11. | Table of Data | 27 |
| 12. | Output Snippet of GA based DMPD Implementation | 28 |
| 13. | Output Snippet of CS based Implementation | 29 |
| 14. | Comparative study of both Cuckoo Search and Genetic Algorithms. | 30 |

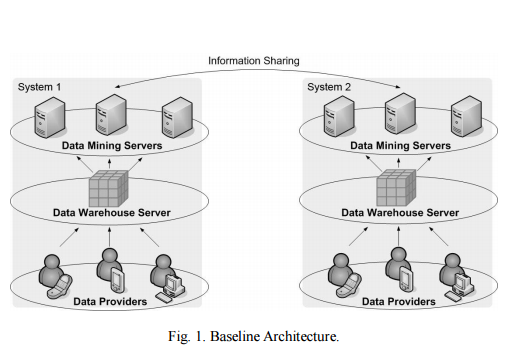
1. **INTRODUCTION**

Data mining is the process of extracting knowledge from large amounts of data. It has been widely and successfully used for more than ten years in various domains, such as marketing, weather forecasting, medical diagnostics, anti-terror measures, etc. Nonetheless, the challenge remains to conduct data mining over private data (e.g., health information) without violating the privacy of data owners (e.g., patients).

Privacy protection has become a necessary requirement in many data mining applications due to emerging privacy legislation and regulations, such as the U.S. Health Insurance Portability and Accountability Act (HIPAA) and the European Union's Privacy Directive. This dissertation seeks to design and compare strategies for protecting privacy in data mining.

* 1. **Baseline Architecture**

Data mining is usually carried out in multiple steps. First, the data being mined are collected from their sources, which we refer to as data providers. In many systems, data providers are physically distributed, forming the bottom tier of the baseline architecture of data mining systems, as shown in Figure 1. Data providers are the data owners, and are expected to submit their (private) data to the data warehouse server, which forms the middle tier of the architecture. For example, in an online survey system, the survey respondents are the data providers who submit their data to the survey analyzer, which holds the data warehouse server.



In the data warehouse server, data collected from the data providers are stored in well-disciplined physical structures (e.g., multi-dimensional data cube), and are aggregated and pre-computed in various forms (e.g., sum, max, min).

For example, in an online survey system, an aggregated data point may be the mean age of all survey respondents. The objective of data warehouse server is to support online analytical processing (OLAP) on the data, and to facilitate data mining. The actual data mining tasks are performed by the data mining servers, which form the top tier of the baseline architecture. When performing data mining tasks, the data mining servers are likely to use the aggregated data, which are pre-computed by the data warehouse server, rather than the rough data, which are directly collected from the data providers, in order to hasten the data mining process. Note that the data mining servers may not have the right to access all data stored in the data warehouse. For example, in a hospital where all patients’ information is stored in the data warehouse, the accounting department of the hospital (as a data mining server) is allowed to access patients’ financial data, but is prohibited from accessing patients' medical records per HIPAA requirements.

* 1. **Design Principle**

In order to introduce the design principle of privacy-preserving data mining systems, we need to define the term "privacy".

Privacy has been a central issue from a sociological standpoint. In the context of information privacy, information is considered to be private if its owner has the right to choose whether or not, to what extent, and for what purpose, to disclose the information to others.

In the literature on privacy preserving data mining, it is commonly (explicitly or tacitly) assumed that a data owner generally chooses not to disclose its private data unless the disclosure is necessary for the purpose of data mining. Based on this assumption, we can state the design principle of privacy-preserving data mining systems as follows.

* 1. **Basic Strategy**

Based on the system architecture and design principle, we now introduce the basic design strategies for privacy-preserving data mining systems.

Apparently, in a data mining system, privacy disclosure can occur when private data are transmitted from one entity to another. Thus, a commonly used privacy protection measure is to enforce privacy-preserving communication protocols between different entities, such that each entity may follow the protocol and thereby prevent private information disclosure during data communication.

Specifically, three kinds of protocols are needed:

• **Data Collection Protocol**, which protects privacy during data transmission from the data providers to the data warehouse server.

• **Inference Control Protocol**, which manages the privacy protection between the data warehouse server and data mining servers.

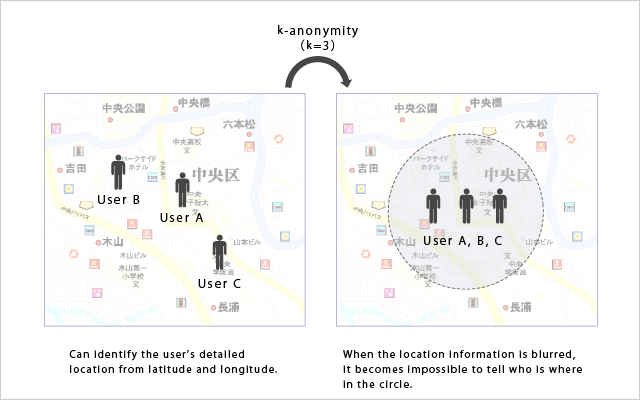
• **Information Sharing Protocol**, which controls the information shared between different data mining servers (of different systems).

1. **PRIVACY PRESERVING DATA MINING (PPDM)**

In privacy-preserving data mining (PPDM), a widely used method for achieving data mining goals while preserving privacy is based on k-anonymity. This method, which protects subject-specific sensitive data by anonymizing it before it is released for data mining, demands that every tuple in the released table should be indistinguishable from no fewer than k subjects.

The most common approach for achieving compliance with k-anonymity is to replace certain values with less specific but semantically consistent values. In this report we propose a different approach for achieving k-anonymity by partitioning the original dataset into several projections such that each one of them adheres to k-anonymity. Moreover, any attempt to rejoin the projections, results in a table that still complies with k-anonymity.

A classifier is trained on each projection and subsequently, an unlabeled instance is classified by combining the classifications of all classifiers.

****

**Figure 2:** K anonymity Model

* 1. **Sample Case**

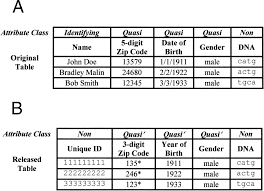
Consider, for example, a government, or more appropriately, one of its security branches interested in developing a system for determining, from passengers whose baggage has been checked, those who must be subjected to additional security measures.

The data indicating the necessity for further examination derives from a wide variety of sources such as police records, airports, banks, general government statistics and passenger information records that generally include personal information (such as name and passport number), demographic data (such as age and gender), flight information (such as departure, destination, and duration), and expenditure data (such as transfers, purchasing and bank transactions). In most countries, this information is regarded as private and to avoid intentionally or unintentionally exposing confidential information about an individual, it is against the law to make such information freely available.

1. **SUPPRESSION**

Suppression is one of the earlier techniques for creating a k-anonymous dataset. With suppression, the most common approach is to entirely replace a certain attribute value with a missing value that can take any other attribute’s domain value.

A major problem with suppression is that the technique can drastically reduce the quality of the data if it is not properly used. On the other hand suppression does not require any knowledge regarding the attributes’ domain values. Thus, this technique is still frequently used.



**Figure 3:** Suppression technique for quasi identifier Date of Birth

1. **GENERALISATION**

A more qualitative approach than suppression in creating a k-anonymous dataset is to generalize attributes which may be used to violate individual privacy. During generalization, original attribute values are substituted for the semantically consistent but less precise values.

For example, the zip code of an individual can be replaced by its two first figures. This is enough to provide sufficient geographical information for data mining. Due to the substitution, the value can be related to more individuals than the zip code in the original dataset.

Appropriate generalization maintains the mean of the data at the record level but an anonymized dataset can contain less information and this can affect the performance of data mining algorithms applied to the dataset. Different algorithms use various methods for selecting the attributes and records for generalization as well as the generalization technique.

**5. FEATURE SET PARTITIONING**

In feature set partitioning, the goal is to decompose the original set of features into several subsets in order to create a classification model for each subset. Subsequently, an unlabeled instance is classified by combining the classifications of all classifiers.

Feature set partitioning generalizes the task of feature selection which is extensively used in data mining. Feature selection provides a representative set of features from which a classifier is constructed. Moreover, feature set partitioning is regarded as specific case of ensemble methodology in which disjoint feature subsets are used, i.e. every classifier in the ensemble is trained on a different projection of the original training set.

In this report we implement feature set selection using first a genetic algorithm based approach followed by the popular cuckoo search.

* 1. **Sample Case**

The following example of feature set partitioning is based on a training dataset derived from health insurance policyholders. Each policyholder is characterized by four features: **Asset Ownership**, **Education (years)**, **Car Engine Volume** (in cubic centimeters) and **Employment Status**. The target feature describes whether a specific policyholder was willing to purchase complementary insurance and, if so, what type. A possible partitioning for resolving the question includes two decision trees. The first decision tree uses the features Asset Ownership and Volume, while the second utilizes Employment Status and Education.

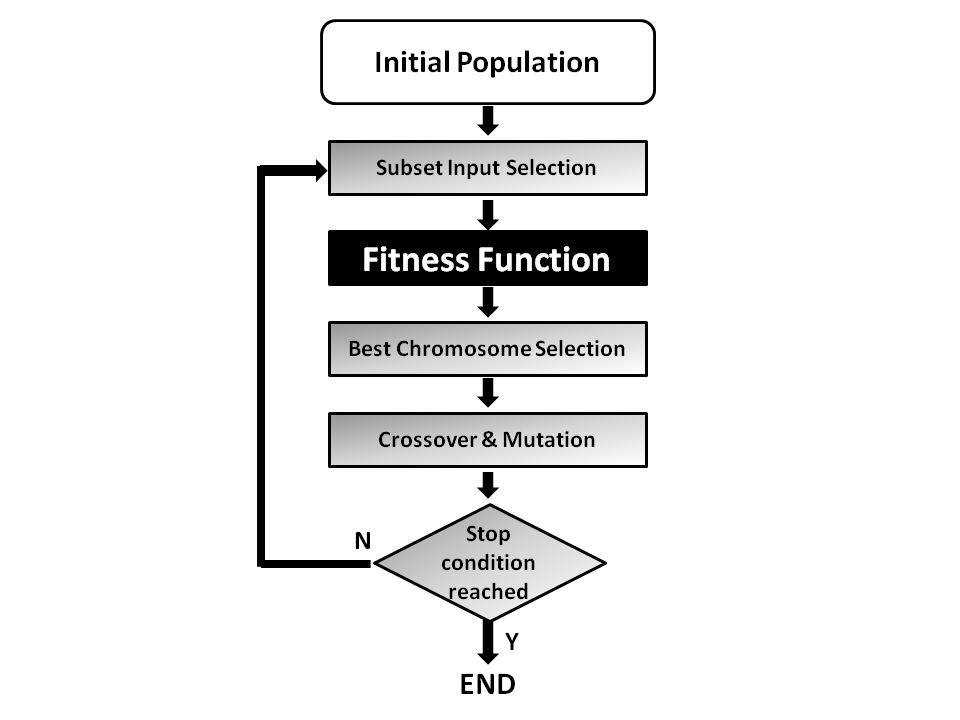
These empirical results point towards the superiority of feature set partitioning in learning tasks that contain a high number of features and a comparatively moderate number of tuples. One of the reasons for this superiority is the ability of feature set partitioning to deal with the ‘‘curse of dimensionality” associated with large feature spaces. The problem also arises in the context of data mining privacy.

**6. GENETIC ALGORITHM BASED SEARCH**

Genetic algorithms (GA), a type of evolutionary algorithm (EA), are computational abstractions, derived from biological evolution, for solving optimization problems through a series of genetic operations.

A GA requires a **fitness function** that assigns a score (fitness) to each candidate in the current population sample (generation). The fitness of a candidate depends on how well that candidate solves the problem at hand. Selection of candidates is performed randomly with a bias towards those with the highest fitness value. To avoid locally optimal solutions, crossover and mutation operators are introduced to produce new solutions along the whole search space.

Thanks to this capability in developing solutions, the GA is recognized today as a highly reliable global search procedure. Other issues involved in using genetic algorithms are the number of details to define in run settings, such as the size of the population and the probabilities of crossover and mutation, and the stop (convergence) criteria of the algorithm. Specific values often depend greatly on the GA’s application.



**Figure 4:** GA Process

**6.1 Advantages and Drawbacks of GA**

GA’s have found to be useful in many data mining tasks in general and in feature selection in particular. In general, these empirical comparisons show that GAs, with their associated global search in the solution space, usually obtain better results than local search-based feature selection methods. Inspired by these positive results, Rokach presented a GA based framework for solving feature set partitioning tasks. As in feature selection, GAs demonstrate a clear superiority over all other search methods when searching for accurate feature set partitions.

The main drawback of the GA approach is that it is computationally expensive compared to greedy search methods. The computational cost of GA might be controlled by appropriately choosing population size and stopping criterion.

**7. DATA MINING PRIVACY BY DECOMPOSITION (DMPD)**

Guided by classification accuracy and k-anonymity constraints, the proposed data mining privacy by decomposition (DMPD) algorithm uses a genetic algorithm to search for optimal feature set partitioning. The DMPD algorithm performs better than existing k-anonymity-based algorithms and there is no necessity for applying domain dependent knowledge. Using multi-objective optimization methods, we also examine the tradeoff between the two conflicting objectives in PPDM: **privacy and predictive performance**.

A generic single-objective GA can be modified to find a set of multiple, non-dominated solutions in a single run. The ability of the GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems.

We propose a privacy-preserving data mining algorithm that does not require any application do- main knowledge for the anonymization process. Instead, the DMPD effectively partitions the original feature to **comply with k-anonymity constraints** and to preserve **maximum information for classification**. As a result, the DMPD algorithm provides a better classification performance compared to other state-of-art generalization and suppression-based approaches. In addition, the algorithm has been extended to provide information about the tradeoff of k-anonymity level and classification accuracy.

**7.1 DMPD Terminology**

**7.1.1 Classification Problem**

In a typical classification problem, a train dataset of labeled examples is given. The train dataset can be described in a variety of ways, most commonly as a collection of records that may contain duplicates. A vector of feature values (sometimes referred to as attributes) describes each record. The notation A denotes the set of input features containing n features: A = {a1, ..., ai,... , an} and y represent the target feature (or class attribute).

**7.1.2 Features**

Features are typically one of two types: categorical and numeric. Categorical features provide qualitative information about the subject of the data and its domain values can be placed in a finite number of categories.

Numeric features provide quantitative information about the data’s subject. Numeric features can be attained by counting or measuring and can receive any value in a predefined range.

**7.1.3 Quasi Identifiers**

Quasi-identifiers are pieces of information that are not of themselves unique identifiers, but are sufficiently well correlated with an entity that they can be combined with other quasi-identifiers to create a unique identifier.

**7.1.4 Fitness Value Function**

A fitness function is a particular type of objective function that is used to summarize, as a single figure of merit, how close a given design solution is to achieving the set aims.

In particular, in the fields of genetic programming and genetic algorithms, each design solution is commonly represented as a string of numbers (referred to as a chromosome). After each round of testing, or simulation, the idea is to delete the 'n' worst design solutions, and to breed 'n' new ones from the best design solutions. Each design solution, therefore, needs to be awarded a figure of merit, to indicate how close it came to meeting the overall specification, and this is generated by applying the fitness function to the test, or simulation, results obtained from that solution.

**7.1.5 Crossover Operation**

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them.

**7.1.6 Mutation Operation**

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

**8. K – ANONYMITY PROPERTY**

K-anonymity is a property possessed by certain anonymized data. The concept of k-anonymity was first formulated by Latanya Sweeney in a paper published in 2002 as an attempt to solve the problem: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful.

A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appear in the release.

**8.1 Sample Case**

In the context of k-anonymization problems, a database is a table with n rows and m columns. Each row of the table represents a record relating to a specific member of a population and the entries in the various rows need not be unique. The values in the various columns are the values of attributes associated with the members of the population. The following table is a **non-anonymized database** consisting of the patient records of some fictitious hospital in Cochin.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Age** | **Gender** | **State of domicile** | **Religion** | **Disease** |
| Ramsha | 29 | Female | Tamil Nadu | Hindu | Cancer |
| Yadu | 24 | Female | Kerala | Hindu | Viral infection |
| Salima | 28 | Female | Tamil Nadu | Muslim | TB |
| sunny | 27 | Male | Karnataka | Parsi | No illness |
| Joan | 24 | Female | Kerala | Christian | Heart-related |
| Bahuksana | 23 | Male | Karnataka | Buddhist | TB |
| Rambha | 19 | Male | Kerala | Hindu | Cancer |
| Kishor | 29 | Male | Karnataka | Hindu | Heart-related |
| Johnson | 17 | Male | Kerala | Christian | Heart-related |
| John | 19 | Male | Kerala | Christian | Viral infection |

**Figure 5:** Non-anonymized database consisting of the patient records of some fictitious hospital in Cochin.

There are 6 attributes and 10 records in this data. There are two common methods for achieving k-anonymity for some value of k.

1. **Suppression:** In this method, certain values of the attributes are replaced by an asterisk '\*'. All or some values of a column may be replaced by '\*'. In the anonymized table below, we have replaced all the values in the 'Name' attribute and all the values in the 'Religion' attribute have been replaced by a '\*'.

2. **Generalization:** In this method, individual values of attributes are replaced by with a broader category. For example, the value '19' of the attribute 'Age' may be replaced by ' ≤ 20', the value '23' by '20 < Age ≤ 30' , etc.

The next table shows the **anonymized database**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Age** | **Gender** | **State of domicile** | **Religion** | **Disease** |
| \* | 20 < Age ≤ 30 | Female | Tamil Nadu | \* | Cancer |
| \* | 20 < Age ≤ 30 | Female | Kerala | \* | Viral infection |
| \* | 20 < Age ≤ 30 | Female | Tamil Nadu | \* | TB |
| \* | 20 < Age ≤ 30 | Male | Karnataka | \* | No illness |
| \* | 20 < Age ≤ 30 | Female | Kerala | \* | Heart-related |
| \* | 20 < Age ≤ 30 | Male | Karnataka | \* | TB |
| \* | Age ≤ 20 | Male | Kerala | \* | Cancer |
| \* | 20 < Age ≤ 30 | Male | Karnataka | \* | Heart-related |
| \* | Age ≤ 20 | Male | Kerala | \* | Heart-related |
| \* | Age ≤ 20 | Male | Kerala | \* | Viral infection |

**Figure 6:** Anonymized database consisting of the patient records of some fictitious hospital in Cochin.

This data has 2-anonymity with respect to the attributes 'Age', 'Gender' and 'State of domicile' since for any combination of these attributes found in any row of the table there are always at least 2 rows with those exact attributes. The attributes available to an adversary are called "quasi-identifiers". Each "quasi-identifier" tuple occurs in at least k records for a dataset with k-anonymity.

**9. CUCKOO SEARCH**

Cuckoo search (CS) is an optimization algorithm developed by Xin-she Yang and Suash Deb in 2009. It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). Some host birds can engage direct conflict with the intruding cuckoos.

For example, if a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Some cuckoo species such as the New World brood-parasitic Tapera have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colors and pattern of the eggs of a few chosen host species

Cuckoo search idealized such breeding behavior, and thus can be applied for various optimization problems. It seems that it can outperform other metaheuristic algorithms in applications.

**9.1 Characteristics of Cuckoo Search**

1. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution.

2. The aim is to employ the new and potentially better solutions (cuckoos) to replace not-so-good solutions in the nests.

3. In the simplest form, each nest has one egg.

4. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

**9.2 Three Idealized Rules**

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.

2. The best nests with high quality of eggs (solutions) will carry over to the next generations.

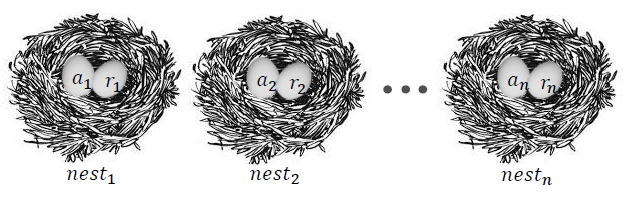
3. The number of available host nests is fixed, and a host can discover an alien egg with probability p ϵ [0,1]. In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

**10. FEATURE SET SELECTION USING CUCKOO SEARCH**

In the DMPD algorithm shown above, the feature set selection process used Genetic algorithm based search optimization technique to select the features. Here, we replace the genetic algorithm with Cuckoo search algorithm and compare both feature sets generated.

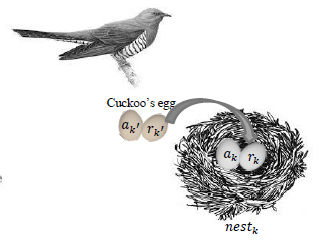
**10.1 The Algorithm**

Step 1: Generate initial population of n host nests.



**Figure 7:** Generation of Initial Population in CS Algorithm.

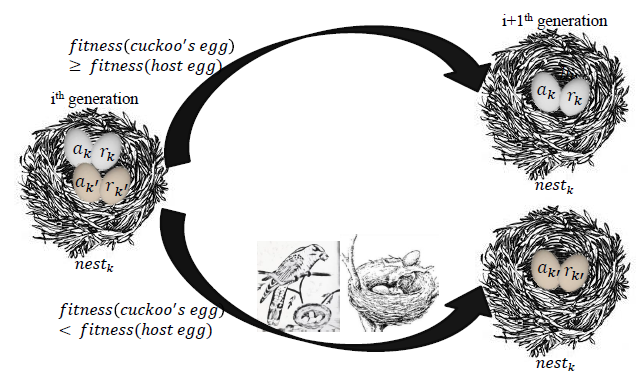
Step 2: Creating a new population using initial population (host eggs) using the technique of levy flight.



**Figure 8:** Creating the next population using levy flight.

Step 3: Compare the fitness of cuckoo’s egg with the fitness of the host egg.

Step 4: If the fitness of cuckoo’s egg is better than host egg, replace the egg in nest k by cuckoo’s egg. If the fitness of the cuckoo’s egg is not better then there is a chance that the host eggs will not be selected in order to make the search more exhaustive.



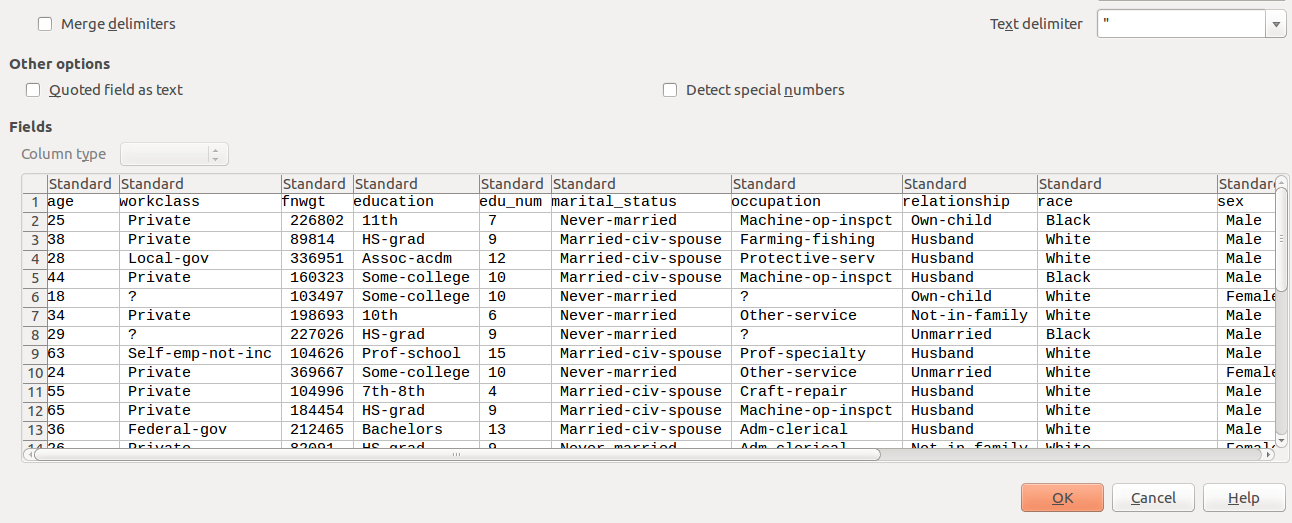
**Figure 9:** Comparing Fitness Values

**11. RESULTS AND ANALYSIS**

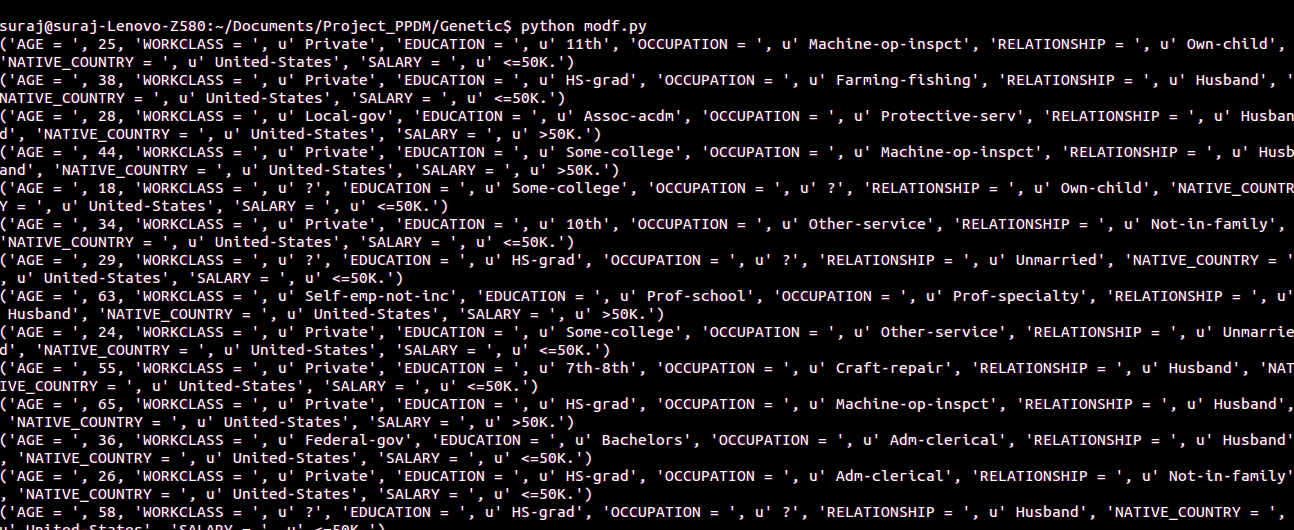
We adopt the adult dataset from the UCI machine learning repository for the experiments. The data set is then subjected to feature set partitioning and selection initially using the DMPD algorithm with genetic algorithm based search. The k value was initially set to three. The initial population is randomly generated as a sequence of bits equal to the number of quasi identifiers. The fitness value function uses a probabilistic sigmoid function to assign values to each population for comparison. The population is then subjected to the genetic algorithm search optimization technique and converged to a minimum threshold of fitness value difference.

The same data set was further subjected to feature selection using cuckoo search algorithm under k – anonymity value constraints. The initial population is randomly generated as a sequence of bits equal to the number of quasi identifiers. This population is then subjected to a random walk to generate closest set of values as compared to the initial population. Fitness values are generated for both parent and random populations generated with a probabilistic replacement of the parent population with the randomly generated population. The algorithm is then converged to the same minimum threshold of fitness value difference is successive iterations.

The final selected feature set are then subjected to anonymization techniques of generalization and suppression techniques to obtain the final data set.



**Figure 10:** UCI Adult Repository Dataset



**Figure 11:** Table of Data

**11.1 GA based DMPD Performance**

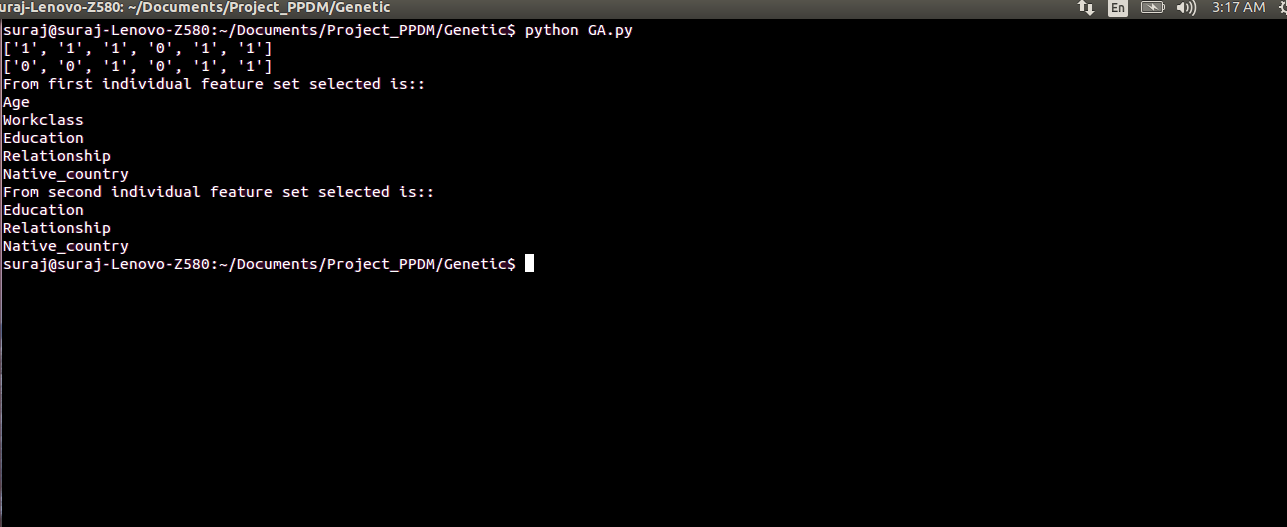
DMPD uses a general framework of feature set partitioning that can be applied prior to transforming feature values. In other words, DMPD can wrap any existing approach for anonymizing datasets and capture it for anonymizing projections following evaluation of partitioning fitness.

The DMPD is capable of dealing with different k-anonymity constraints and inducers without any significant effect on classification accuracy.

When compared to the state-of-the-art k-anonymity methods, DMPD classifiers provide an equivalent or slightly higher degree of accuracy.

DMPD, unlike other methods, does not use any prior knowledge. In TDS, TDR, Incognito and GA-based algorithms, the user is required to provide a taxonomy tree for categorical features. This makes these methods difficult to implement.

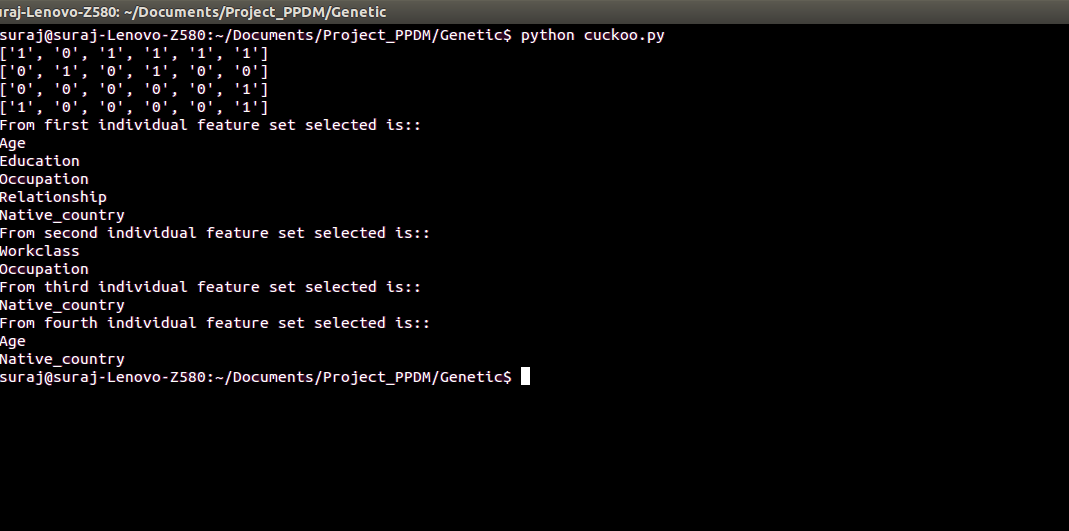
DMPD provide results without using any generalization of categorical feature values. Very possibly this feature results in deeper knowledge than the state-of-art generalization approaches.

DMPD can provide useful information regarding the tradeoff between k-anonymity constraints and classification performance.

**Figure 12:** Output Snippet of GA based DMPD Implementation.

**11.2 Performance of CS Algorithm**

The proposed algorithm has been validated and compared with other algorithms such as genetic algorithms and particle swarm optimization. Simulations and comparison show that CS is superior to these existing algorithms for multimodal objective functions. This is partly due to the fact that there are fewer parameters to be fine-tuned in CS than genetic algorithms. Furthermore, our simulations also indicate that the convergence rate is also quicker. This also means that we do not have to fine tune these parameters for a specific problem.

Subsequently, CS is more generic and robust for many optimization problems, comparing with other metaheuristic algorithms. This potentially powerful optimization strategy can easily be extended to study multiobjecitve optimization applications with various constraints, even to NP-hard problems. Further studies can focus on the sensitivity and parameter studies and their possible relationships with the convergence rate of the algorithm. Hybridization with other popular algorithms such as PSO will also be potentially fruitful.

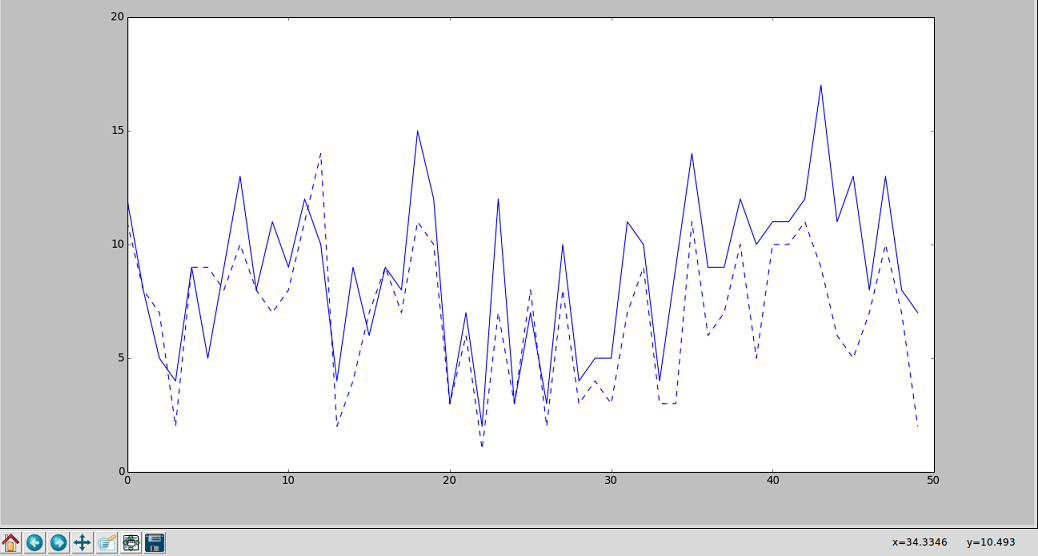
**Figure 13:** Output Snippet of CS based Implementation.

**11.3 Comparative Study**

The performance comparison of feature set partitioning and selection using both genetic algorithm (DMPD) and cuckoo search is carried out as follows.

The minimum fitness value difference between two successive iterations which forms the stopping condition was kept a constant for both the algorithms. Both the algorithms were run on the same dataset for the same number of times (in this case 50). Since both the algorithms select the next generation randomly, the output of each run is a probabilistic combination of features used. The k – anonymity constraint was also kept constant for both algorithms (k = 3).

For each algorithm the number of iterations it takes to converge were recorded and plotted against the number of runs of each algorithm.



**Figure 14:** Comparative study of both Cuckoo Search and Genetic Algorithms.

Note: Dotted line indicates number of iterations taken by Cuckoo search to converge.

Solid line indicates number of iterations taken by GA to converge.

X – Axis indicates the number of runs of both algorithms.

From the graph plot it is evident that the cuckoo search algorithm generally converges faster than the genetic algorithm based search.

**12. CONCLUSION**

Given a database both the feature set selection algorithms used in this report optimally select the set of features that satisfy the k – value constraints. The cuckoo search algorithm converges to a final set of values faster than the genetic algorithm based search owing to the method of generating the next closest population.

A new data anonymization method by randomizing the data records is introduced in this project. We arbitrarily replace part of the values in each record while providing the data relationships in the entire data set, which is dissimilar from most anonymization methods. This method not only accomplishes a higher level of privacy protection, but also preserves more knowledge than the other anonymization methods. We can say that, the valuable relations which are less susceptible can be exposed more precisely than the sensitive ones.

**13.** **BIBLIOGRAPHY**

**1.** Privacy-preserving data mining: A feature set partitioning approach

By: Nissim Matatov, Lior Rokach, Oded Maimon

Accepted 12 March 2010

Journal homepage: www.elsevier.com/locate/ins

**2.** A Binary Cuckoo Search algorithm for feature selection

By: D. Rodrigues ; Department of Computing, UNESP - Univ Estadual Paulista, Bauru, São Paulo, Brazil ; L. A. M. Pereira ; T. N. S. Almeida ; J. P. Papa

Dated: 19-23 May 2013

Published in: 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)

Print ISBN: 978-1-4673-5760-9

**3.** S. Hettich and S. D. Bay, The UCI KDD Archive,

http://kdd. ics. uci. edu, 1999.

**4.** C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin and

M. Y. Zhu, Tools for privacy preserving distributed data mining, ACM SIGKDD Explorations Newsletter, 4 (2): 28-34, 2002.

**5.** A. Hussain, N. Hameza and N. Hefny, Attacks on Anonymization-based privacy preserving: A survey for data mining and data publishing, Journal of Information Security, Vol. 4, No. 2, pp. 101-112, 2013.