Information Sciences 180 (2010) 2696–2720

Contents lists available at ScienceDirect

Information Sciences

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

Privacy-preserving data mining: A feature set partitioning approach

Nissim Matatov [a], Lior Rokach [b,c,*], Oded Maimon [a]

[a] Department of Industrial Engineering, Tel-Aviv University, Israel
[b] Department of Information System Engineering, Ben Gurion University of the Negev, Be'er Sheva 84105, Israel
[c] Duetsche Telekom Laboratories at Ben Gurion University of the Negev, Be'er Sheva 84105, Israel

ARTICLE INFO

Article history:
Received 12 December 2008
Received in revised form 17 January 2010
Accepted 12 March 2010

Keywords:
Data mining
Privacy
Genetic algorithms
k-Anonymity
Feature set partitioning

ABSTRACT

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 paper 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 unlabelled instance is classified by combining the classifications of all classifiers.

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. Ten separate datasets were evaluated with DMPD in order to compare its classification performance with other *k*-anonymity-based methods. The results suggest that DMPD performs better than existing *k*-anonymity-based algorithms and there is no necessity for applying domain dependent knowledge. Using multiobjective optimization methods, we also examine the tradeoff between the two conflicting objectives in PPDM: privacy and predictive performance.

1. Introduction

Knowledge discovery in databases (KDD) is the "process of identifying valid, novel and potentially useful and ultimately understandable patterns in data" [15]. Data mining, which lies at the core of the KDD process, is based on algorithms that explore data and extract valuable information from the patterns that emerge. Data mining has emerged as a key tool for a wide variety of applications in such fields as diverse as astronomy and marketing [63].

Some applications involve mining sensitive data about individuals or corporations and this has led to a growing concern that data mining can violate individual privacy and attempts to limits its implementation [30,9]. For example, in 2003 the Data-Mining Moratorium Act [16] imposed a freeze on data mining by the Department of Defense and the Department of Homeland Security until Congress had thoroughly reviewed the Terrorism Information Awareness Program.

Commercial concerns are also concerned with the privacy issue. Most organizations collect information about individuals for their own specific needs. Very frequently, however, different units within an organization or even different organizations

* Corresponding author at: Department of Information System Engineering, Ben Gurion University of the Negev, Be'er Sheva 84105, Israel.
E-mail addresses: matatov.n@gmail.com (N. Matatov), liorrk@bgu.ac.il (L. Rokach), maimon@eng.tau.ac.il (O. Maimon).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.03.011

themselves may find it necessary to share information. In such cases, each organization or unit must be sure that the privacy of the individual is not violated or that sensitive business information is not revealed [7].

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.

While various means of preserving individual information have been developed, there are ways for circumventing these methods. In our example, in order to preserve privacy, passenger information records can be de-identified before the records are shared with anyone who is not permitted directly to access the relevant data. This can be accomplished by deleting from the dataset unique identity fields, such as name and passport number. However, even if this information is deleted, there are still other kinds of information, personal or behavioral (e.g. date of birth, zip code, gender, number of children, number of calls, number of accounts) that, when linked with other available datasets, could potentially identify subjects.

To avoid these types of violations, various countries have sought to introduce privacy laws. The OECD (Organization for Economic Co-operation and Development) Privacy Guidelines, established in 1980 include general principles regarding the use of personal data. EU member countries subsequently adopted the Data Protection Directive 95/46/EC [13], based on previous OECD Guidelines [50]. This directive, which was implemented in the form of national data protection legislation, imposes obligations on the data controller and data processor.

"The Healthcare Information Portability and Accountability Act" (HIPAA) enacted by the US Congress in 1996 is another legislative attempt to regulate data mining. This act seeks to create national standards for protecting medical records and other personal health information [60].

These two bodies of laws, HIPAA and the OECD Privacy Guidelines, have had a significant impact on PPDM essentially because of their definitions of privacy. These two sets of regulations consider individual privacy to be protected as long as any information cannot be traced to a specific individual. For example, within the OECD, Data Protection Directive 95/46/EC regulates using personal data which is defined as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity". The US's HIPAA is directed to protecting "individually identifiable health information", which means the information "that identifies an individual and with respect to which there is no reasonable basis to believe that the information can be used to identify an individual" [60].

Individual data privacy definitions involve addressing complex individual privacy issues. For example, inferences based on different types of individual data and aimed at achieving sensitive information about an individual can lead to serious privacy violations. In other words, the individual must be protected not only against exposure of sensitive information but also against attempts to evaluate this data more accurately [30].

In an effort to analyze the implications of these principles in data mining, Meints and Moller [43] propose establishing absolute principles in privacy-preserving data mining, including standardizing the definitions of data mining, privacy, privacy violation, and policies. The standardization would enforce privacy safeguards and control how personal data is shared and used.

Several researchers have proposed methods for incorporating privacy-preserving requirements in various data mining tasks, such as classification [58]; frequent itemsets [69]; and sequential patterns [31]. In this paper we focus on classification tasks and the k-anonymity model as proposed by Sweeny [62]. A dataset complies with k-anonymity constraints if for each individual, the data stored in the released dataset cannot be distinguished from at least $k-1$ individuals whose data also appears in the dataset. Or more generally, each original individual record can only be reconstructed based on released data with a probability that does not exceed $1/k$, given knowledge based on information available from external sources.

Suppression is one of the earlier techniques for creating a k-anonymous dataset [56]. 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 [61]. On the other hand suppression does not require any knowledge regarding the attributes' domain values. Thus, this technique is still frequently used [8,70,32].

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 [8].

The main difficulty of existing generalization algorithms is the need for domain hierarchy trees for every quasi-identifier attribute [28,65,22,37,38,19,21]. These attribute hierarchies are created manually and the process requires prior knowledge regarding the problem domain.

We propose a new method for achieving *k*-anonymity, – data mining privacy by decomposition (DMPD). The basic idea is to divide the original dataset into several disjoint projections such that each one of them adheres to *k*-anonymity. It is easier to make a projection comply with *k*-anonymity if the projection does not contain all quasi identifier features. Moreover, our procedure ensures that even if the attacker attempts to rejoin the projections, the *k*-anonymity is still preserved. A classifier is trained on each projection and subsequently, an unlabelled instance is classified by combining the classifications of all classifiers. Because DMPD preserves the original values and only partitions the dataset, we assume that it has a minimal impact on classification accuracy. DMPD does not require domain trees for every attribute nor does it fall into the difficulties encountered by existing algorithms which reduce the validity of the dataset (such as suppression). DMPD supports classification, but can be extended to support other data mining tasks by incorporating various types of decomposition [10].

DMPD employs a genetic algorithm for searching for optimal feature set partitioning. The search is guided by *k*-anonymity level constraint and classification accuracy. Both are incorporated into the fitness function. We show that our new approach significantly outperforms existing suppression-based and generalization-based methods that require manually defined generalization trees. In addition, DMPD can assist the data owner in choosing the appropriate anonymity level. Using the Pareto efficiency principle, we offer a better way to understand the tradeoff between classification accuracy and privacy level [49].

The rest of the paper is organized as follows: in Section 2 we present an overview of related work and methodologies. In Section 3 we describe the goals of our study and present appropriate definitions while in Section 4 we outline the DMPD method. In Section 5 we evaluate and discuss different aspects of the new method for providing *k*-anonymity. Conclusions and future research issues are presented in Section 6.

2. Related work

In this section we briefly review some of the topics that were mentioned in the previous section. These issues fall into four categories:

- Privacy-preserving data mining
- The feature set partitioning approach
- Genetic algorithm-based search
- Genetic algorithms for multiobjective optimization

2.1. Privacy-preserving data mining

Privacy-preserving data mining (PPDM), a relatively new research area, is focused on preventing privacy violations that might arise during data mining operations [64,30,2]. In implementing this goal, PPDM algorithms modify original datasets in order to preserve privacy even after the mining process is activated. The aim is to ensure minimal data loss and to obtain qualitative data mining results.

Verykios et al. [64] describe PPDM approaches based on five dimensions: (1) data distribution – whether the data is centralized or distributed; (2) data modification – where the modification technique is used to transform the data values; (3) data mining algorithm – the data mining task to which the approach is applied; (4) data or rule hiding – refers to whether raw data or aggregated data should be hidden; (5) privacy preservation – the type of selective modification performed on the data as a part of the PPDM technique: heuristic-based, cryptography-based or reconstructing-based.

The approach presented in this paper displays all these five features: DMPD works on a centralized dataset (dimension 1); DMPD transforms the original input feature set to subsets (dimension 2); DMPD considers classification algorithms (dimension 3); DMPD hides the raw data (dimension 4); since we are using the *k*-anonymity framework for data mining privacy, DMPD performs as a heuristic-based technique (dimension 5).

There are other data mining privacy issues that may arise in relation to various PPDM approaches. For instance, Oliveira and Zaïane [49] argue that it is meaningful to define the type of violation to be avoided when applying the PPDM algorithm. Most state-of-art PPDM approaches consider linking an attack on the dataset or data mining results (e.g. classification rules). For example, in cases where an algorithm provides a *k*-anonymous dataset, Friedman et al. [20] discuss the possibility of building *k*-anonymous data mining models using *k*-anonymous datasets. These types of models protect individual privacy when using the model, e.g. classification tree. Another PPDM issue is the definition of privacy used in the algorithm. For example, for perturbation (reconstructing-based) techniques using adding noise [2], privacy is defined by a range of values that can contain the original value. A wider range assumes more privacy.

The most popular PPDM techniques use the *k*-anonymity concept for data mining privacy. *k*-anonymity, as proposed by Sweeny [62], requires that no individuals can be linked with fewer than *k* rows in a dataset. This is achieved by ensuring that in a dataset the owner releases there are at least *k* rows with the same combination of values in the attributes that potentially can be used for individual privacy violation. This statement guarantees that the probability of identifying an individual

based on the released data does not exceed $1/k$. The model provides clear privacy definition and primarily considers linking attacks on the dataset. Despite simplicity and understandability, the k-anonymity model rests on two main assumptions: the data owner knows the set of attributes which can be used to identify the individual (quasi-identifier constraint) and that the k-level is high enough to protect the data for all possible misuse of the dataset.

The most common approach for achieving compliance with k-anonymity is to replace certain values with less specific but semantically consistent values (generalization). Alternatively there is the possibility of not releasing some of the values at all (suppression). The problem of finding optimal κ-anonymous datasets using generalization or suppression has been proved to be NP-hard [46,57]. Therefore, heuristic algorithms are needed. One group of generalization and suppression approaches is guided by various heuristic measures based on minimizing *data loss* [60,6,37,38]. In such cases, data quality depends on how far away each attribute value is from the original one after applying the anonymization process. Optimizing an aggregated value over all features and records leads to minimum data loss. Li and Li [39] present a general taxonomy and a comparative study of different generalization schemes.

In the context of KDD, the k-anonymization process is designed to retain useful information for data mining. In other words, the goal is to comply with k-anonymity while providing the best data mining results (for example, classification accuracy).

Iyengar [28] uses a genetic framework to search for the best set of generalizations to satisfy k-anonymity constraints. For this purpose, each generalization is presented as a chromosome and genetic operators produce a new type of valid generalization.

Wang et al. [65] presented a practical and effective bottom-up generalization for preserving classification information while ensuring privacy. In this work the authors defined the metric for incorporating the privacy level and the information gain for choosing the appropriate generalization. The bottom-up generalization technique, however, can only generalize categorical attributes.

Fung et al. [22] presented another practical generalization method for classification using k-anonymity: the top-down specialization (TDS) algorithm. This algorithm handles both categorical and continuous attributes and has some practical advantages over the earlier bottom-up approach. TDS starts from the most general state of the dataset and tries to specialize it by assigning specific values to attributes until a violation of the anonymity occurs. More recently Fung et al. [21] presented an improved version of TDS which is called top-down refinement (TDR). In addition to the capabilities of TDS, TDR is capable of suppressing categorical attribute values to treat attributes without any taxonomy tree.

One common disadvantage of current generalization algorithms is the need to perform manual pre-processing, i.e., generating domain generalization taxonomy to define the hierarchy of the categorical attributes values, a process that involves prior knowledge about the domain. In addition to the necessity of separately preparing a domain hierarchy for every attribute, domain experts are also faced with differences among themselves about the correct structure of the taxonomy tree. This may lead to differences in results [28,22]. In an effort to automatically build the domain taxonomy, Nergiz and Clifton [48] proposed clustering techniques for generating domain hierarchies. The generated hierarchies are not restricted to trees. To the best of our knowledge, the TDR algorithm is the only algorithm that efficiently relaxes the need for a taxonomy tree for categorical attributes by using a single-dimension suppression operator for attributes for which a domain taxonomy does not exist.

While generalization using domain knowledge can provide more visible patterns [4], it also can result in overly shallow knowledge which may be uninteresting (or even useless) in a specific domain. For example, in predicting the likelihood of disease, we can discover the pattern "USA $\diamondsuit$ Disease". Here the "USA" value was obtained by generalizing a patient's geography attribute. Such a pattern in an international study is not worthwhile for acquiring new knowledge, but a model built on a generalized dataset still able provide high classification accuracy. Some reference to the generalization problem can be found in [28]. The author presents two generalization techniques for satisfying k-anonymity constraints driven by two different metrics. The first technique tries to minimize the data loss metric (LM). The metric must be used to meet the possibility of multiple data usage by various users and for different purposes that cannot be known at the time the data is disseminated. Obviously, the approach can preserve more interesting patterns, that is, a higher quality of data mining results. The second metric tries to maximize the classification metric (CM) that incorporates class distribution. Generalizations based on this metric are more oriented to preserving information for classification. The results presented by Iyengar indicate that the classification metric (CM) provides more accurate results than the data loss metric (LM). On the basis of Iyengar's results, we maintain that to achieve high classification accuracy, it does not always to preserve data, but preserve useful information for classification.

Another approach is to embed the anonymization process into the learning algorithm. Friedman et al. [19] proposed a method for embedding anonymization procedures into decision tree construction from an original dataset. The proposed algorithm is basically an improvement of the classical decision tree-building algorithm, combining mining and anonymization in a single process. The algorithm produces models that protect individual privacy when using the model, e.g. the classification tree.

Still another important issue in k-anonymity approaches is the treatment of numeric attributes. Discretization is the most direct way of dealing with numeric attributes for PPDM. This technique was mentioned in [2,21] as one of several possible ways of modifying numeric value in data mining privacy. During the procedure, each numeric value of an attribute in the data is mapped to an appropriate interval. The bound of the interval can be found by learning the relation between the attribute and target feature (supervised discretization) or by bounding based on some information about the attribute values

(unsupervised discretization). After the discretization, the numeric attribute is treated as a categorical attribute with the domain values as the intervals created during the discretization. DMPD treats numeric features by supervised discretization performed at initialization step of the algorithm. Supervised discretization uses the class information of the training instances to select discretization cut points. Since DMPD performs *k*-anonymization for classification tasks, the choice of supervised discretization for exploiting additional possibilities for achieving better classification performance is obvious [14]. In general, discretization is performed for attributes defined as a part of *k*-anonymity constraint for quasi-identifiers.

2.2. Feature set partitioning for data mining

The DMPD algorithm presented in this work considers anonymization for classification through feature set partitioning. The concept of feature set partitioning, initially presented by Rokach and Maimon [55], was proposed for improving supervised learning tasks. 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 unlabelled instance is classified by combining the classifications of all classifiers.

Maimon and Rokach [42] present the following example of feature set partitioning 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.

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 [44] 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.

Rokach and Maimon's [55] 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 [1].

The framework for feature set partitioning that Rokach [52] provided includes a range of techniques for applying the approach to classification tasks. The framework's components are described in more detail in relation to DMPD properties in Section 4.

To summarize, the feature set partitioning approach offers improved classification performance due to its ability to [41,54,53]:

- avoid the "curse of dimensionality" for large input feature spaces
- act as a feature selection procedure
- act as a ensemble approach

2.3. 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 [24].

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.

GAs have found to be useful in many data mining tasks in general and in feature selection in particular [18,23,68]. Empirical comparisons between GAs and other kinds of feature selection methods can be found in [59] as well as in [36]. 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 [53] 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.

Mitchell [47] indicates the circumstances in which GAs are particularly useful: "the space to be searched is large; is known not to be perfectly smooth and unimodal; or it is not well understood or if the fitness function is noisy." The search space in feature set partitioning is known to be large [53] and in our case, the goal function is not well understood due to the tradeoff between *k*-anonymity constraints and classification performance.

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. Fortunately, the DMPD algorithm presented in this paper shows relatively quick convergence to accurate solutions (see Section 5.7). Additionally, a relatively long anonymization time might be acceptable, if the anonymization can be performed offline. Finally even if GA cannot be always used for online scenarios, it provides a tight lower bound for the performance of the optimal partition. Thus, from the research point of view, GA can be faithfully served to evaluate the suitability of feature set partitioning in solving PPDM tasks.

Generalized accuracy can possibly operate as an appropriate fitness function for GAs. To evaluate the generalized accuracy, we can use the wrapper approach, initially presented by Kohavi and John [34]. The main advantage of this approach is that it generates reliable evaluations and can be utilized for any induction algorithm. A major drawback, however, is that this procedure repeatedly executes the inducer. For this reason, wrappers may not scale well to large datasets containing many features or instances.

Other state-of-art *k*-anonymity approaches consider optimizing data mining performance given the *k*-anonymity level constraint as an input. Based on a clear understanding that classification performance and *k*-anonymity level are conflicting objectives, real-world decisions are based on a tradeoff between these two performance measures rather than determining that one is better or more preferable than the other.

2.4. Genetic algorithms for multiobjective optimization

The DMPD algorithm presented in this work was extended in a natural way to perform a multiobjective optimization to assist data owners in deciding about an appropriate anonymity level in released datasets.

The multiobjective genetic algorithm (MOGA) was designed to solve multiobjective problems where the objectives are generally conflicting thus preventing simultaneous optimization of each objective. The final choice of the solution depends on the user characterizing a subjective approach. User participation in this process is important for obtaining useful results [15].

Optimizing competing objective functions is different from single function optimization in that it seldom accepts one perfect solution, especially for real-world applications [35]. The most successful approach for multiobjective optimization is to determine an entire Pareto optimal solution set or its representative subset [24,17]. A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) *vs.* a certain amount of gain in the other (s). In [29], the authors reported that 90% of the approaches to multiobjective optimization aimed to approximate the true Pareto front for the underlying problem.

The ultimate goal of a multiobjective optimization algorithm is to identify solutions in the Pareto optimal set. However, identifying the entire Pareto optimal set is practically impossible for many multiobjective problems due to the set's size. In addition, for many problems, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Therefore, a practical approach to multiobjective optimization is to investigate a set of solutions (the best-known Pareto set) that represent the Pareto optimal set as best as possible.

With these concerns in mind, a multiobjective optimization approach should achieve the following three, often conflicting, goals [35]:

1. The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto optimal set.
2. Solutions in the best-known Pareto set should be uniformly distributed and diverse over the Pareto front in order to provide the decision-maker a true picture of the tradeoffs.
3. The best-known Pareto front should capture the whole spectrum of the Pareto front through investigating solutions at the extreme ends of the objective function space.

Being a population-based approach, genetic algorithms are well suited to solve multiobjective optimization problems. 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.

One state-of-art multiobjective genetic algorithm is the strength Pareto evolutionary algorithm (SPEA). An earlier version of SPEA was among the first techniques that were extensively compared to several existing evolution-based methods [73,71]. Improved versions of the algorithm (SPEA2) have been subsequently proposed [72]. One of the main concepts of the algorithm is that an archive set implements the elitism concept where some best solutions are introduced as is, without any modification, into the next generation. The algorithm uses Pareto ranking for fitness value calculation. In cases where there are many different solutions, the algorithm performs density estimation using a cluster-based technique. Density information has to be used in order to guide the search more effectively in achieving goals 2 and 3 described above. In the final phase, the archive set contains the solutions that form the Pareto frontier. SPEA2 provides an improved ability in searching for a better solution in the solution space and in achieving a uniform and complete Pareto front. This algorithm in relation to DMPD approach is described in more detail in Section 4.4.

To sum up this section, we propose a privacy-preserving data mining algorithm that does not require any application domain 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.

3. Problem formulation

In this section we formulate the problem and introduce several basic definitions that are used in the course of this paper.

Definition 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: $A = \{a_1, \ldots, a_i, \ldots, a_n\}$ and y represent the target feature (or class attribute). 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.

When the feature a_i is categorical, it is useful to denote its domain values by $dom(a_i)$. In a similar way, $dom(y) = \{c_1, \ldots, c_{|dom(y)|}\}$ represents the domain of the target feature. Numeric features have infinite cardinalities and their domain can be presented by a range of possible values. For discretized numeric features, any numeric value can be related to one of intervals and further treated as a categorical value. The instance space X presents the set of all possible examples and is defined as a Cartesian product of all the input feature domains: $X = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n)$. The *labeled instance space* U is defined as a Cartesian product of all input feature domains and the target feature domain, i.e., $U = X \times dom(y)$. The training dataset, consisting of a set of m instances/tuples, is denoted as $S = (\langle x_1, y_1 \rangle, \ldots, \langle x_m, y_m \rangle)$ where $x_q \in X$, $y_q \in dom(y)$ and x_{qi} denotes the value of feature i of instance q. Usually, it is assumed that the train dataset instances are distributed randomly and independently according to some fixed and unknown joint probability distribution D over U.

It is assumed that a given inducer I is used to build a classifier (also known as a classification model) by learning from S. The classifier can then be used for classifying unlabelled instances. The notation $I(S)$ represents a classifier which was induced by activating the induction method I onto dataset S. The classifier also provides estimates to the conditional probability of the target feature given the input features.

We next consider the common notation of bag algebra to present projection and selection of instances [25] where S denotes a relation. The following relation S (Fig. 1) represents the sample drawn from the adult dataset of the UC Irvine Machine Learning Repository [45]. This dataset contains census data and has become a commonly used benchmark for

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
27	Local-gov	54826	Assoc-voc	10	Widowed	Prof-specialty	Not-in-family	White	Female	0	0	20	US	<=50K.
47	Local-gov	181344	Some-college	10	Married	Prof-specialty	Husband	Black	Male	0	0	55	US	>50K.
41	Local-gov	523910	Bachelors	13	Married	Prof-specialty	Husband	Black	Male	0	0	40	US	<=50K.
42	Local-gov	254817	Some-college	10	Married	Prof-specialty	Not-in-family	White	Female	0	1340	40	US	<=50K.
27	Private	213921	HS-grad	9	Divorced	Other-service	Not-in-family	White	Male	0	0	40	US	<=50K.
46	Private	51618	HS-grad	9	Married	Other-service	Not-in-family	White	Female	0	0	40	US	<=50K.
59	Private	159937	HS-grad	9	Married	Sales	Husband	White	Male	0	0	48	US	>50K.
49	Private	343591	HS-grad	9	Divorced	Prof-specialty	Not-in-family	Black	Female	14344	0	40	US	>50K.
53	Private	346253	HS-grad	9	Married	Sales	Husband	White	Male	0	0	35	US	<=50K.
55	Private	198282	Bachelors	13	Married	Sales	Husband	White	Male	15024	0	60	Germany	>50K.
40	Private	118853	Bachelors	13	Widowed	Sales	Unmarried	White	Male	0	0	60	Germany	<=50K.
49	Private	77143	Bachelors	13	Married	Sales	Husband	White	Male	0	0	40	Germany	>50K.
47	Private	253814	HS-grad	9	Divorced	Prof-specialty	Unmarried	White	Female	0	0	25	US	<=50K.
21	Private	312956	HS-grad	9	Married	Prof-specialty	Not-in-family	Black	Male	0	0	40	US	>50K.
43	Private	114580	Some-college	10	Married	Prof-specialty	Not-in-family	White	Female	0	0	40	US	<=50K.
61	State-gov	267989	Bachelors	13	Married	Prof-specialty	Husband	White	Male	0	0	50	US	>50K.
46	State-gov	102628	Masters	9	Widowed	Prof-specialty	Unmarried	White	Male	0	0	40	US	<=50K.
28	State-gov	175325	HS-grad	9	Widowed	Prof-specialty	Unmarried	White	Male	0	0	40	US	<=50K.
28	State-gov	149624	Bachelors	13	Divorced	Prof-specialty	Unmarried	White	Male	0	0	70	US	>50K.
56	State-gov	149624	Bachelors	13	Divorced	Other-service	Unmarried	White	Male	0	0	25	US	>50K.

Fig. 1. Adult dataset sample.

k-anonymity-based algorithms. We also use the sample data from the adult dataset to provide further explanation of Definitions 2–9. The adult dataset has six numeric and eight categorical features. The target feature is income level with two possible values "$\leqslant$ 50 K" or ">50 K".

Definition 2 (*Projection*). The projection operator π with the form $\pi_B(S)$ is used to project relation S onto a subset of features $B \subseteq A$. For example, $\pi_{age,relationship,native\text{-}country}(S)$ results in the relation presented in Fig. 2 (note that duplicates are not removed):

Definition 3 (*Aggregation*). The result of an aggregation operator with the form ${}_BG_{\theta(C)}(S)$ is produced by grouping S according to all grouping features ($B \subseteq A$). At the next step, within each group we compute $\theta(C)$, where θ is an aggregation operator and C is set of aggregated features ($C \subset A$ and $B \cap C = \phi$). For example, ${}_{sex}G_{COUNT(*)}(S)$ results in the relation presented in Fig. 3.

Definition 4 (*Quasi-identifier*). The quasi-identifier definition presented here is adopted from [62].

Given a population of entities E, an entity-specific dataset $S \in U$ with input feature set A, $f_1{:}E \to S$ and $f_2{:}S \to E'$ where $E \subseteq E'$. Q is quasi-identifier of S, if $Q \subseteq A$ and $\exists e \in E$; such that $f_2(\pi_Q f_1(e)) = e$.

The formulation defines quasi-identifiers as a set of features whose associated values may be useful for linking in order re-identify the entity that is the subject of the data. For example, in the "Adult" dataset sample (Fig. 1), $Q = \{$*marital-status, occupation*$\}$ is a quasi-identifier since the values of these features can be linked to identify an individual who is the subject of the 6th tuple in the dataset with values {*Married, Other-service*}.

Definition 5 (*Feature categories in k-anonymity model*). We now provide several basic definitions related to the *k*-anonymity model in regard to privacy-preserving data mining methods. These definitions are based on previous research of the model [62,30,5,48,20].

Each of the input features can be considered as an attribute related to one of three categories. The first category $QI = \{Q_1, Q_2, \ldots, Q_m\}$ is a subset of *public features*. Each of these attributes is to be found in a public data source or can be discovered

age	relationship	native-country
27	Not-in-family	US
47	Husband	US
41	Husband	US
42	Not-in-family	US
27	Not-in-family	US
46	Not-in-family	US
59	Husband	US
49	Not-in-family	US
53	Husband	US
55	Husband	Germany
40	Unmarried	Germany
49	Husband	Germany
47	Unmarried	US
21	Not-in-family	US
43	Not-in-family	US
61	Husband	US
46	Unmarried	US
28	Unmarried	US
28	Unmarried	US
56	Unmarried	US

Fig. 2. Adult sample after projection.

sex	count
Female	6
Male	14

Fig. 3. Aggregated Adult sample dataset.

from other sources in order to perform individual identification. This list of attributes is created by the data owner. If the assumption about how input features can be used or the relation between them is too strict, we can use a scenario where all features act as potential quasi-identifiers. However, this forceful way is often impractical in real-world applications.

The second category $P = \{P_1, P_2, \ldots, P_k\}$ presents a set of unknown features that cannot be used for individual identification. Features in the category are treated as non-quasi-identifiers features.

The third category $S = \{S_1, S_2, \ldots, S_l\}$ presents a set of sensitive features that must be protected from individual disclosure. Typically, in classification problems there is a target feature.

Definition 6 (*Quasi-identifier constraint*). A quasi-identifier constraint QI of S is a subset of an input feature set:

$$QI = \{Q_1, Q_2, \ldots, Q_m\}, \quad QI \subseteq A$$

Set $QIS = \{QI_1, QI_2, \ldots, QI_r\}$ includes all the quasi-identifier constraints considered in the problem. As in previous studies (such as [32]), we assume that the user provides the quasi-identifier constraint and that there is only one feature set in the quasi-identifier constraint.

Definition 7 (*k-Anonymity*). The dataset's anonymity level with respect to QI, denoted as $DAL_{QI}(S)$, is equal to the minimum number of tuples that have the same combination of values on QI in $\pi_{QI}(S)$, or:

$$DAL_{QI}(S) = {}_{QI}G_{\min(count)}({}_{QI}G_{COUNT(*)}(\pi_{QI}(S)))$$

where *count* denotes the result column of the first aggregation operation.

A dataset is said to satisfy k-anonymity with respect to QI if and only if $DAL(QI) \geqslant k$.

In other words, a dataset S complies with k-anonymity constraints if no individual can be distinguished on QI from at least $k-1$ other individuals whose data also appears in the dataset.

For example, for the data presented in Fig. 1, if $QI = \{$*workclass, native-country*$\}$, the dataset's anonymity level is 3 because the sequence of values {*Private, Germany*} is related to three tuples, the minimum number. Therefore, in regard to the question of whether the dataset adheres to three-anonymity, the answer is positive but negative for higher values of k. If we define $QI = \{$*workclass, native-country, marital-status*$\}$, the k-anonymity level of the dataset is 1. In other words, this feature set is a quasi-identifier for this dataset and individual privacy is violated.

Specifying higher values for k (k-anonymity level constraint) and more features in quasi-identifier QI (quasi-identifier constraint) results in stricter privacy constraints.

Definition 8 (*k-Anonymous feature set partitioning*). A disjoint partitioning $Z = \{G_1, \ldots, G_r, \ldots, G_\omega\}$, has a *partial anonymity level* defined as:

$$PAL_{QI}(Z) = \min_r DAL_{QI}(\pi_{G_r}(S))$$

A partitioning satisfies *partial k-anonymity* (*or partially valid*) with respect to QI if and only if $PAL_{QI}(Z) \geqslant k$. Partial k-anonymity guarantees that each projection complies with k-anonymity and that individual privacy is protected according to the classic k-anonymity model (Definition 7).

Set k-anonymity guarantees k-anonymity protection for a pair of projections without selection conditions and having one join path [67]. A pair of projections V_1, V_2 complies with k-anonymity if:

$$DAL_{QI \cup y}(V_1| \times |V_2) \geqslant k$$

A set of projections corresponding to given partitioning Z has a *set anonymity level* defined as:

$$SAL_{QI}(Z) = \min_{\forall G_p, G_r \in Z} DAL_{QI \cup y}(PDS_p(S)| \times |PDS_r(S))$$

where $PDS_r(S) = \pi_{G_r \cup y}(S)$ and $|Z| \geqslant 2$. Due to the fact that the natural join is an associative operation, we can choose any order of projections for validating the partitioning. A partitioning satisfies *set k-anonymity* with respect to QI if and only if $SAL_{QI}(Z) \geqslant k$.

The *partitioning anonymity level* is defined in general way as:

$$PartAL_{QI}(Z) = \min\{PAL_{QI}(Z), SAL_{QI}(Z)\}$$

If $|Z| = 1$, then $PartAL_{QI}(Z) = PAL_{QI}(Z)$.

A partitioning satisfies k-anonymity with respect to QI if and only if:

$$PartAL_{QI}(Z) \geqslant k$$

In this case, the partitioning Z is defined as a *valid partitioning*. Since such a partitioning complies with *partial* and *set anonymity*, then, under reasonable assumptions, no operation on projections allows the attacker to violate privacy provided by a k-anonymity model. Clearly, an empty partitioning complies with k-anonymity. We use these definitions as building blocks for proof of DMPD correctness in Section 4.3.

To demonstrate the idea how feature set partitioning effectively meet k-anonymity requirements, we follow the example presented in Definition 7. For $QI = \{workclass, native\text{-}country, marital\text{-}status\}$, the k-anonymity level of the entire dataset is $k = 1$. However, by partitioning the input feature set and distributing the QI features among two partitions, the k-anonymity constraint is partially relaxed. For example the partitioning that contains the two feature subsets:

$$G_1 = \{age, workclass, fnlwgt, education, education\text{-}num, occupation, native\text{-}country\} \quad \text{and}$$
$$G_2 = \{marital\text{-}status, relationship, race, sex, capital\text{-}gain, capital\text{-}loss, hours\text{-}per\text{-}week\}$$

The *partial anonymity level* equal to three, since this is the minimum number of rows that share the same values combination ({*Private, Germany*} for first feature subset). In addition we check the *set anonymity level* and realize that the value combination ({*Private, Divorced, Germany*, $\leqslant 50K$.} appears twice in the join result. Consequently, we conclude that *partitioning anonymity level* is $k = 2$. Namely, by releasing the two partitions G_1 and G_2 instead of releasing the entire dataset, we succeeded to increase the anonymity level from $k = 1$ to $k = 2$.

The main goal of this study is to introduce a new k-anonymity-based algorithm which is capable of finding a feature set partitioning that appeals to k-anonymity constraints and achieves a classifier that will perform as similarly as possible to a classifier trained on the original dataset. Consequently, the problem can be formally phrased as follows:

Definition 9 (*Problem formulation: preserving k-anonymity via feature set partitioning*). Given an inducer I, a combination method C, and training set S with input feature set $A = \{a_1, \ldots, a_i, \ldots, a_n\}$ and target class y from a distribution D over the labeled instance space. The goal is to find an optimal and valid partitioning $Z_{optvalid} = \{G_1, \ldots, G_r, \ldots, G_w\}$ of the input feature set A into w mutually exclusive subsets when $G_r \subseteq A$. Optimality is defined in terms of minimization of the generalization error of the induced classifiers $I(\pi_{G_r \cup y} S)$ combined using method C over the distribution D.

4. Methods

In this section we describe our new method for k-anonymity based on a feature set partitioning framework using the algorithm framework that Rokach [52] developed for feature set partitioning. In this framework, each method that captures feature set partitioning must define its various basic properties. We now provide an overview of the properties and DMPD settings for these properties:

1. Structure acquisition method – the way the partitioning structures is obtained. In our case, the structure is acquired with a GA search procedure with an appropriate fitness value assignment for the partitioning, crossover and mutation operators.
2. Mutually exclusive property – determines if the algorithm uses mutually exclusive or partially overlapping partitioning. Our method is implemented with mutually exclusive partitioning of subsets.
3. Inducer usage – indicates the relation between the decomposer and the inducer used. In this context our method uses an "inducer-independent" type of partitioning method. The algorithm works with any given inducer and the same inducer is used in all subsets. More specifically, the method uses a fitness value based estimating the generalization classification accuracy using a wrapper approach for any given inducer.
4. Exhaustiveness – indicates whether all features should be used in the partitioning structure. Our method, using non-exhaustive feature set partitioning, takes into account the possibility that some features will not participate in the partitioning structure.
5. Combiner usage – specifies the relation between the decomposer and the combiner. Out method is combiner-dependent. More specifically, our method uses a naïve Bayes combination for classifying unlabeled instances.
6. Sequentially or concurrently – indicates whether the various sub-classifiers are built sequentially or concurrently. We use a concurrent framework where the classifiers are built independently and their results combined.

We now define in more detail the framework's three main procedures – the search for optimal partitioning, partitioning evaluation and combining partial classifiers. Previous work into feature set partitioning includes different procedures for searching for the optimal partitioning. Among these methods are serial search with incremental oblivious trees, multi-search [54], and genetic algorithm-based search [53]. In [53], genetic-based algorithms demonstrate a clear superiority over all other search methods. Partitioning evaluation methods include the VC-dimension [42] and the wrapper approach [54]. Combining methods include voting and naïve Bayes [52].

We refer to our new method as data mining privacy by decomposition (DMPD). In Section 4.1 we describe the method. In Section 4.2 we demonstrate the method with an example. In Section 4.3 we analyze the correctness and complexity of the proposed method.

4.1. Privacy-preserving classification by feature set partitioning

As noted previously, DMPD consists of three main procedures. In the first procedure, a search for an optimal and valid partitioning based on genetic search is carried out. The second involves evaluating a given partitioning while the third

procedure combines multiple classifiers to obtain unlabeled instance classifications. Procedure dependency is described as follows: the genetic algorithm uses a wrapper approach to assign a partitioning fitness value while the wrapper uses combined multiple classifiers to obtain test instance prediction. In the final phase, given the optimal partitioning, we anonymize the original dataset.

4.1.1. Procedure 1: Genetic algorithm-based search

A genetic algorithm for feature set partitioning was presented in [53]. We use this work's partition presentation, crossover and mutation operator.

Given the partitioning $Z = \{G_1, \ldots, G_k, \ldots, G_w\}$, elements of the matrix B with dimension $n \times n$ are assigned as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } i = j \quad \text{and} \quad \exists k : a_i \in G_k \\ 1, & \text{if } i \neq j \quad \text{and} \quad \exists k : a_i \in G_k \quad \text{and} \quad a_j \in G_k \\ -1, & \text{if } \forall k : a_i \notin G_k \quad \text{and} \quad a_j \notin G_k \\ 0, & \text{otherwise} \end{cases}$$

For example, matrix presentation can be used to present a dataset containing four non-target features in the following partitioning $\{\{a_1\}\{a_2,a_3\}\}$ (see Fig. 4).

After the initial population has been produced, the genetic algorithm provides a procedure for choosing the individuals in the population in order to create the offspring that will form the next generation. More particularly, in this stage two parents are chosen and two output offsprings become a combination of these parents using crossover and mutation operator.

In consideration of a special encoding scheme for partitioning, there is a crossover operator. The operator, "group-wise crossover" (GWC), works with probability $P_{crossover}$ on two selected partitions. The operator, together with the proposed encoding, does not slow the convergence of the GA [53]. From two of these parents, two ancestor sets are chosen. An ancestor set can be a filtered out, representing a set of features that have not participated in parent partitioning. Fig. 5 presents an example of two offspring from two possible parent partitions from the presented feature set.

As Fig. 5b indicates, the bright gray in the columns and rows denotes ancestor sets from the first parent. The dark grey presents cells copied from another parent. A similar operation is performed on the second parent.

The mutation operator is defined as follows: with probability P_{mut}, each feature can pass from one subset (source subset) to another (target subset). The feature and subset to which it passes are chosen randomly. If there is only one subset in a partitioning, the source feature will create its own subset. Fig. 6 demonstrates the idea. In this case, the second feature was randomly chosen and passed to a subset containing a third feature.

As an inner procedure, the GA uses a wrapper approach to evaluate a partitioning fitness value.

4.1.2. Procedure 2: Wrapper-based fitness evaluation

We use a wrapper procedure for evaluating the partitioning fitness value. The fitness value was the average accuracy over n runs, where each time $n - 1$ folds were used for training classifiers and one-fold for estimating the generalized accuracy of

1	0	0	0
0	1	1	0
0	1	1	0
0	0	0	- 1

Fig. 4. Encoding of example partitioning.

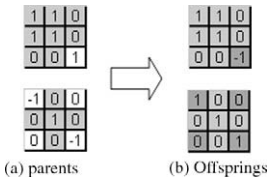

Fig. 5. Example of GWC operator.

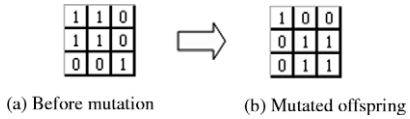

Fig. 6. Example of mutation operator.

a validation dataset after combining predictions from different classifiers and obtaining final instance predictions. To obtain appropriate classifications, the wrapper uses Procedure 3 for instance classification.

4.1.3. Procedure 3: Instance classification

In the naive Byes combination, a classification of a new instance is based on the product of the conditional probability of the target feature given the values of the input features in each subset. Mathematically it can be formulated as follows:

$$v_{MAP}(x_q) = \underset{c_j \in dom(y)}{\arg\max} \frac{\prod_{k=1}^{w} \widehat{P}_{I(\pi_{G_k \cup y} S)}(y = c_j | a_i = \pi_{G_k} x_q)}{\widehat{P}_{I(S)}(y = c_j)^{w-1}} \tag{1}$$

where $\widehat{P}_{I(\pi_{G_k \cup y} S)}(y = c_j | a_i = \pi_{G_k} x_q)$ is estimated by using the appropriate frequencies in the relevant decision tree leaves and then adjusting them by performing a Laplace correction.

Our algorithm is described in Algorithm 1.

Algorithm 1. [DMPD Algorithm]

DMPD (S,I,QI,k,nGen,nPop,Pcrossover,Pmutation,nFolds)

```
Input:
S (Original dataset); I (Inducer); QI (The quasi-identifier); k (anonymity level); nGen (Maximum
  number of generations); nPop (Number of candidates in a population); Pcrossover (Probability of
  crossover); Pmutation (Probability of mutation); nFolds (Number of folds for wrapper procedure)
Output:
S'    Set of anonymized projections
1:  InitializeAlgorithm ()
2:  Repeat
3:    Generation ← PerformSelection (Population,nPop)
4:    Generation ← CreateNewGeneration (Population,Pcrossover,Pmutation)
5:    Do ∀Z ∈ Generation
6:      Evaluation (Z) = EvaluatePartitioning (Z)
      End Do
7:    Population ← UpdatePopulation (Population∪Generation)
8:  Until Performed nGen generations
9:  BestValidPartitioning ← SelectBestValidPartitioning (Population)
10: S' = PrepareAnonymisedData (BestValidPartitioning)
11: Return S'
```

EvaluatePartitioning (Z)

```
Input: Z   Partitioning
Output: fv   Partitioning fitness value
12:  If PAL_QI(Z) < k Then fv=0
13: Else fv = PerformWrapperCrossValidation(Z,S,I,nFolds)
     End If
14: Return fv
```

SelectBestValidPartitioning (Population)

```
Input: Population Population in descending order of fitness values
Output: Z   Best partitioning satisfying set k-anonymity
15:  Repeat
16:    Z ← Get next best partition in the population
17:    If SAL_QI (Z) ⩾ k Return Z
18:  Until no partitions are left
19: Return Z = ∅.
```

PrepareAnonymisedData (Z)

```
Input: Z   Optimal partitioning
Output: S' Set of projections
20:  For each G_r ∈ Z Do
21:        PDS_r ← π_{G_r ∪ y}(S)
22:        Randomize (PDS_r)
23:  End For
25: Return PDS_1,...,PDS_|Z|
```

4.2. Illustration

In this section we use the dataset of Fig. 1 to illustrate the proposed algorithm. We assume *QI* = {*capital-gain, age, marital-status, relationship*} and that *k*-anonymity level constraint is 2 (*k* = 2). Assume that the genetic algorithm selects (Line 3) two parent partitioning structures on the basis of fitness values (Line 6), calculated previously to produce offsprings for the next generation (Line 4): G_i and G_j, as follows: G_i = {{*relationship*}, {*education-num*}}, G_j = {{*hours-per-week, age*}}.

After applying a GWC operator (with probability Pcrossover on two random ancestor subsets), we get two offsprings as follows: $G_i' = \{\{relationship\}, \{hours\text{-}per\text{-}week, age\}\}, G_j' = \{\{relationship\}, \{education\text{-}num\}, \{hours\text{-}per\text{-}week, age\}\}$. After applying the mutation operator on G_j' (with probability Pmutation), a feature "*education-num*" joins the set {*relationship*} to produce $G_j'' = \{\{relationship, education\text{-}num\}, \{hours\text{-}per\text{-}week, age\}\}$.

Continuing with G_j'' , we check that the partitioning appeals to partial *k*-anonymity: the anonymity of the first subset is 6 while that of the second is 7; the anonymity level of the partitioning is 6 (Definition 7). This level is more than *k* so G_j'' satisfies the first condition for valid partitioning. If otherwise, the fitness would be set to 0. Note that "*age*" is a feature in the quasi-identifier constraint that has been discretized.

The next step is to build partial classifiers on the adult dataset according to a given partitioning structure and then combining them to get a final instance classification from the validation dataset. In partitioning G_j'' for first subset {*relationship, education-num*} the following classification tree is constructed:

```
relationship = Not-in-family: ⩽50 K (7.0/2.0)
relationship = Husband: >50 K (7.0/2.0)
relationship = Unmarried
|   education-num ⩽10: ⩽50 K (3.0)
|   education-num > 10: >50 K (3.0/1.0)
```

For the second subset {*hours-per-week, age*} the following classification tree is built:

```
age = '(-inf-48]'
|   hours-per-week ⩽40: ⩽50 K (10.0/1.0)
|   tt hours-per-week >40: >50 K (3.0/1.0)
age = '(48-inf)': >50 K (7.0/1.0)
```

The a priori distribution of the target feature values is 0.55 ("⩽50 K")/0.45 (">50 K"). Assume the combiner receives the following instance from the validation dataset with the values of the relevant features as follows: {*education-num, relationship, hours-per-week, age*} = {13, "*Unmarried*", 30, 25}. The first partial classifier assigns the instance's target value to " >50 K". The second partial classifier classifies the instance as "⩽ 50 K". After calculating $v_{MAP}(x_q)$ using the distribution on leafs, we conclude that the target feature value of the instance must be assigned to "⩽50 K".

Given all the predictions, DMPD calculates classification accuracy and the average accuracy, evaluated on *n* train folds, is returned to the GA search procedure as a fitness value of the partitioning (Line 14). The algorithm's GA search procedure stops after the maximum allowed number of generations is achieved (Line 8).

Before constructing a set of anonymized projections, we check that the second condition is met (Lines 15–19). In this example, we first check that the best partition (i.e. with the highest fitness value): $G_j'' = \{\{relationship, education\text{-}num\}, \{hours\text{-}per\text{-}week, age\}$ complies with the second condition. Since $SAL(G_j'') = 2, G_j''$ is returned as the selected solution. Otherwise, we should continue by checking the second best partition.

After we select the best valid partition, we construct the projections (Lines 20–24). First, we add the target feature to each of the subsets (Lines 20–21). Then each projection is randomized to get a final anonymization. In our example, the set of anonymized projections appears as in Fig. 7.

4.3. Correctness analysis

Proposition 1. *The DMPD algorithm is correct.*

Proof. First we prove in Theorem 1 that the DMPD algorithm terminates then we prove in Theorem 2 that the algorithm's result complies with *k*-anonymity. □

Theorem 1. *DMPD algorithm terminates.*

Proof of Theorem 1. In the first step, the DMPD algorithm performs an initialization that contains a finite number of the algorithm's parameters. This creates an initial population bounded by number of features from the original dataset (Line 1).

education-num	relationship	income1	age	hours-per-week	income2
10	Not-in-family	<=50K.	(-inf-48]	20	<=50K.
10	Husband	>50K.	(-inf-48]	55	>50K.
13	Husband	<=50K.	(-inf-48]	40	<=50K.
10	Not-in-family	<=50K.	(-inf-48]	40	<=50K.
9	Not-in-family	<=50K.	(-inf-48]	40	<=50K.
9	Not-in-family	<=50K.	(-inf-48]	40	<=50K.
9	Husband	>50K.	(48-inf)	48	>50K.
9	Not-in-family	>50K.	(48-inf)	40	>50K.
9	Husband	<=50K.	(48-inf)	35	<=50K.
13	Husband	>50K.	(48-inf)	60	>50K.
13	Unmarried	<=50K.	(-inf-48]	60	<=50K.
13	Husband	>50K.	(48-inf)	40	>50K.
9	Unmarried	<=50K.	(-inf-48]	25	<=50K.
9	Not-in-family	>50K.	(-inf-48]	40	>50K.
10	Not-in-family	<=50K.	(-inf-48]	40	<=50K.
13	Husband	>50K.	(48-inf)	50	>50K.
9	Unmarried	<=50K.	(-inf-48]	40	<=50K.
9	Unmarried	<=50K.	(-inf-48]	40	<=50K.
13	Unmarried	>50K.	(-inf-48]	70	>50K.
13	Unmarried	>50K.	(48-inf)	25	>50K.

Fig. 7. Sample set of anonymized projections.

In the next step DMPD performs a constant number (*nGen*) of iterations in the main function (Lines 2–8). In each iteration of the main function, the algorithm chooses *nPop*/2 candidates from a current population of candidates (Line 3) that is performed in a finite number of steps. To create a new generation (Line 4), we perform a constant number of operations on a matrix of $n \times n$ size that also can be accomplished in a finite number of steps.

Given a finite set of partitionings, we evaluate them (Line 6). Each partitioning can impose a limited number of projections ($w \leqslant n$). As the first step in the partitioning evaluation (Line 12), we check that the given partitioning appeals to partial *k*-anonymity. This procedure counts the occurrences of each combination of values over *QI* features in the specific projection ($\pi_{G_r}(S)$); searches for the minimum number over all combinations and projections; and compares this value to *k*. The procedure can be done in a finite number of steps. For each projection corresponding to a partitioning, we build *nFolds* classifiers and perform *nFolds* evaluations on the validation dataset (Line 13). Assuming that the training of a classifier terminates, we just need to prove that the evaluation of partitioning accuracy terminates. During partitioning evaluation, we simply review a finite number of instances and obtain from each classifier the classification of the instance. Assuming that the classification of each instance always terminates, then partitioning accuracy evaluation also terminates.

Searching for the best valid partitioning (Line 9) includes reviewing an ordered list of partitionings whose fitness value appeals to partial *k*-anonymity and searching for partitionings (Lines 15–19) that also appeal to set *k*-anonymity. Assuming that an examination of whether a partitioning appeals to set *k*-anonymity can be accomplished in a finite number of steps and that the number of partitionings is bounded by $nGen^{*}nPop$, we conclude that this step terminates.

Preparing a set of anonymized projections consist of two steps (Lines 20–24). As a first step, DMPD selects a finite number of instances in each projection. In the second step, the target feature column is appended to a finite number of earlier generated projections ($w \leqslant n$) and then randomized. After passing over all instances in each projection, this step also terminates.

Since a simple loop in the main program and out-of-loop procedures have terminated, we conclude that DMPD algorithm also terminates.

Theorem 2. *DMPD outputs a set of projections that complies with k-anonymity.*

Proof of Theorem 2. We consider three cases:

1. The partition is empty. In this case, in releasing an empty dataset we comply with *k*-anonymity because no private data was released.
2. There is only one subset in the selected partition ($|Z_{optvalid}| = 1$). In this case the released dataset complies with *k*-anonymity based on Lemma 1.
3. There are at least two subsets ($|Z_{optvalid}| \geqslant 2$). In this case we assume that each subset separately complies with *k*-anonymity (Case 2). Thus we need to show that any set of two or more projections cannot be used to violate the *k*-anonymity (see Lemma 2).

Lemma 1. *DMPD outputs a set of projections that appeals to partial k-anonymity.*

Proof of Lemma 1. We must show that partitioning used for constructing projections (Line 9) appeals to partial k-anonymity (see Definition 8), i.e., each projection $\pi_{G_r}(S)$ imposed by best partitioning appeals to k-anonymity to protect individual identification.

The initial population includes an empty partitioning (all features are filtered out) that enters the initial population during initialization step (Line 1). Its fitness value is greater than 0 since a classification is performed according to the majority class. Releasing an empty partitioning corresponds to a decision to release $S' = \varnothing$. In Fig. 8 we present a sample of original data. When there is no valid partitioning for any non-empty, quasi-identifier constraint and $k \geqslant 2$, DMPD explicitly releases an empty set of anonymized projections.

At each *nGen* generation of a genetic algorithm, if a better partitioning is found it must comply with partial k-anonymity. A better partitioning is one that has the highest fitness value found so far. Partitionings that do not appeal to partial k-anonymity (Definition 8) are set as fitness values that are equal to 0 (Line 12) and therefore cannot provide better partitioning for the population (Line 7).

After *nGen* generations of GA, the best valid partitioning is selected from an ordered list of all partitionings that appeal to partial k-anonymity as ordered by their fitness values in descending order. In the worst case, we release an empty dataset that also appeals to partial k-anonymity (Lines 15–19) by Definition 8.

Lemma 2. *DMPD outputs a set of projections that complies with k-anonymity.*

Proof of Lemma 2. The following lemma is formulated on the assumption that the functional dependencies, other than the quasi-identifier, are not known to the attacker. For a projection-only sub-case, we use a verification procedure presented in [67] where a set of projections is k-anonymous. We must consider that each subset of projections corresponds to partitioning since each such subset provides a join path on class attributes and can be used for violating privacy.

According to the post-processing (Line 9) procedure, the best valid partitioning appeals to set k-anonymity (Lines 15–19). From Definition 8, each subset of two projections is k-anonymous. We show that according to this assumption, partitioning complies with set k-anonymity according to a verification procedure presented by Yao et al. [67]. Or mathematically, $\forall z \in 2^{|Z_{optvalid}|}$ the following requirement is fulfilled:

$$DAL_{QI\cup y}(|PDS_1(S)| \times |PDS_2(S)| \times |\cdots| \times |PDS_{|z|}(S)) \geqslant k \tag{2}$$

Since the natural join operation is associative, the order of projections and feature subsets is not important.

We prove by induction that a number of projections corresponding to optimal and valid partitioning also comply with set k-anonymity (see Definition 8 and Line 17 of Algorithm 1).

Base step. $|Z_{optvalid}| = 2$. Each pair of subsets in partitioning complies with set k-anonymity and thus complies with k-anonymity.

Assumption step. We assume that it is true that $|Z_{optvalid}| = n$.

Induction step. We prove for $|Z''_{optvalid}| = n + 1$. Or we must prove that the requirement of Eq. (2) is fulfilled when each join result J on subsets in $2^{|Z_{optvalid}|}(|Z_{optvalid}| = n)$ is joined with one additional subset $PDS_{|z|+1}$ with feature subset $G_{|z|+1}(Z'_{optvalid} = Z_{optvalid} \cup G_{|z|+1})$.

The result of natural join $J| \times |PDS_{|z|+1}$ described by relational bag consists of each pair $\langle t, Count_{QI\cap G_{Z_{optvalid}}\cup y}\rangle$, where $G_{Z_{optvalid}} = \bigcup_{w=1}^{|Z_{optvalid}|} G_w$. If t agrees on $QI \cap G_{Z_{optvalid}} \cup y$ in J and on $QI \cap G_{|z|+1} \cup y$ in $PDS_{|z|+1}$ then:

$$Count_{QI\cap G_{Z'_{optvalid}}\cup y}(t) = Count_{QI\cap G_{Z_{optvalid}}\cup y}(t) * Count_{QI\cap G_{|z|+1}\cup y}(t).$$

Dayal et al. [11] proved the validity this expression for any number of conjunctive expressions where each an expression can be a result of natural join operator.

In other words, in the final join result, each combination of values on $QI \cap Z'_{optvalid} \cup y$ will appear a number of times as a multiplication of number of appearances in J and $PDS_{|r|+1}$. Since the induction assumption of J contains at least k such rows, it is true also for $J| \times |PDS_{|r|+1}$. Therefore, we can argue that a natural join of an arbitrary number of projections complies with k-anonymity.

Proposition 2. *The computational complexity of the DMPD algorithm above is*

$$O(nGen * nPop(nPop + n * nFolds * F(m,n) + n^3m^2))$$

where m is number of instances, n is a number of features, nFolds is a number of folds is used for wrapper procedure, nGen is the number of generations and nPop is the population size.

a_1	a_2	a_3	y
a	b	c	y_1
d	e	f	y_2
g	h	i	y_3
j	k	l	y_4

Fig. 8. Sample of original data causes empty data to be released.

Proof. As a first step DMPD performs a supervised discretization. This step is bounded by $m \log m$ operations. At each iteration of the GA's outer loop, we perform *nGen* generations where each implements the steps that follow. The selection of best candidates (Line 3) is bounded by $nPop^2$). During new generation creation, each candidate follows a crossover and mutation procedure (Line 4). Since crossover and mutation operators perform a constant number of operations on a matrix of $n \times n$ size, their complexity is linear with respect to n. Given the candidate partitioning (Line 6) and train dataset, we calculate a partial anonymity level and use it in fitness value assignment (Line 12). This step can be implemented by using a hash table. Given the constant complexity for searching and retrieving an object, the complexity of this step in the worst case is O (nm), when the anonymity level calculated for n feature subsets and each instance forms a unique combination of values in the quasi-identifier. To describe the worst case, we assume that each partitioning complies with partial k-anonymity. For each partially valid partitioning, we perform *nFolds* cross-validation for partitioning evaluation (Line 13). At each iteration of the cross-validation procedure, we run at most n inducers on the *nFolds*-1 data and evaluate the partitioning on the remainder of the data. The overall complexity of the partial induction classification models is bounded by $n^*F(m,n)$, where $F(\bullet)$ denotes the complexity of given inducer I. For all feature subsets, the overall complexity for obtaining the conditional probability vector of a certain instance as well as its prediction for partitioning evaluation is bounded by the complexity of the induction classification models. Therefore, the complexity of the wrapper-based partitioning evaluation is dominated by $n^*F(m,n)$. □

Searching for the best valid partitioning (Line 9) includes searching for a partitioning that appeals to set k-anonymity (Lines 15–18). In the worst case we should review the whole history of partially valid partitions that were created during the GA search and there are $nGen^*nPop$ such partitionings. For each partitioning we should calculate its set anonymity level as stated in Definition 8. Each partitioning contains n subsets. Therefore, the number of natural joins to be evaluated is bounded by n^2. First, each natural join is performed in $O(m \log m)$ operations by using the sort-merge join algorithm. Then, given the result of the natural join, we calculate the anonymity level which takes $O(nm^2)$ operations. Thus, in total, calculating the set anonymity level has a complexity of $O(nGen^*nPop^*n^3m^2)$.

Projection size during final anonymization is bounded by n. Each anonymized projection prepared by performing O(m) operations on the original dataset. Therefore, this step has a complexity of O(mn) and computationally dominated by previous steps.

Summarizing all the steps we obtain the overall complexity as formulated in Proposition 2.

Practically, the computational complexity is affected by k-anonymity constraints: the size of the quasi-identifier and that of the required k-anonymity level. For large k and quasi-identifiers, more candidate partitioning structures are not partially valid and the step of cross-validation is not performed. A more precise conclusion about the influence of k-anonymity constraints on complexity proposes making several assumptions on train data distribution (how the constraints are strict in the presence of knowledge about the underlying data).

4.4. DMPD for multiobjective optimization

The DMPD extension for multiobjective optimization uses a basic genetic algorithm extension to provide Pareto frontiers with two objectives: maximizing classification performance and maximizing the individual privacy level. In DMPD the first objective presented by generalization accuracy and the second – by anonymity level of partitioning.

Similar to SPEA2 [72], which is a state-of-art multiobjective GA (MOGA), DMPD uses a regular population and an archive (external set). Starting with an initial population and an empty archive, the following steps are performed per iteration. First, all non-dominated population members are copied to the archive; any dominated individuals or duplicates (with the same objective values) are removed from the archive during this update operation. If the size of the updated archive exceeds a predefined limit, further archive members are deleted by a clustering technique which preserves the characteristics of the non-dominated front. Existing archives provide an elitist concept brought from a classic genetic algorithm. A non-dominated member is one that has some objective value better than all others partitioning structures in a population. If the first partitioning's two objectives are better than those of another, we conclude that the first partitioning dominates.

To illustrate the process, consider a population of three partitioning structures with accuracy and anonymity levels: {{80.5, 200}, {90, 500}, {88, 700}}. The first partitioning is dominated by the second and third partitioning while the second partitioning has better accuracy but lower k-anonymity compared to third partitioning. Consequently, these two partitioning structures are on Pareto frontier and present us with an interesting tradeoff.

As discussed earlier, the main difference of MOGA from regular genetic algorithms lies in its fitness value assignment. Fitness values are assigned to both archives and new population members. First, each individual i is assigned a strength value $S(i)$ which represents the number of solutions it dominates. These Svalues are calculated for all individuals in the population and archive sets. In the next step, for each individual j, the raw fitness value $R(j)$ is calculated as a sum of the strengths of individuals that dominate j. As a result, better solutions are assigned the lower raw fitness values.

The second component of fitness value is the density metric $D(i)$ defined as the distance between two objectives. $D(i)$ is calculated between i and its k-nearest neighbor. The density metric has values between 0 and 1 since its purpose is to discriminate between two non-dominated candidates to provide a uniform distribution of non-dominated candidates along the Pareto frontier. The sum of the two metrics provides a fitness value for the MOGA.

As a mating method, we use rank-based weighting instead of binary tournament selection proposed as a part of SPEA2. After a stop condition (in DMPD it is the maximum number of generations) for the GA is achieved, the archive set contains a set of non-dominated solutions that form the Pareto frontier.

5. Experimental evaluation

The proposed method was evaluated in the presence of *k*-anonymity constraints for classifications tasks. The comparative experiment was conducted on 10 benchmark datasets containing individual information about various objects. Specifically, the experimental study had the following goals:

1. to examine whether the proposed algorithm succeeded in satisfying the broad range of *k*-anonymity constraints without sacrificing data mining performance (i.e. the original classification accuracy vs. increasing *k*-anonymity constraints)
2. to investigate the sensitivity of the proposed method to different classification methods
3. to compare the proposed method to existing *k*-anonymity-based methods in terms of classification accuracy
4. to examine the sensitivity of runtime cost to different *k*-anonymity constraints and the scalability of the proposed method.
5. to investigate the sensitivity of the proposed method to different GA settings
6. to investigate the multiobjective DMPD method in providing Pareto frontiers of classification accuracy vs. partitioning anonymity level.

The following subsections describe the experimental set-up and the results obtained.

5.1. Experimental process

Fig. 9 graphically represents the experimental process that was conducted. The main aim of this process was to estimate the generalized accuracy (i.e. the probability that an instance was classified correctly). First, the dataset (box 1) was divided into a train dataset (box 3) and test dataset (Step 4) using five iterations of a two-fold cross validation (Step 2 – known as the 5×2 CV procedure). The 5×2 CV is known to be better than the commonly used 10-fold cross-validation because of the acceptable Type-1 error [3]. At each iteration, the dataset is randomly partitioned into two equal-sized sets, S1 and S2, such that the algorithm is evaluated twice. During the first evaluation S1 is the train dataset and S2 the test dataset, and vice versa during the second evaluation. We apply (Step 5) the *k*-anonymity method on the train dataset and obtain a new anonymous train dataset (Step 6). Additionally, we obtain a set of anonymity rules (Step 7) that is used to transform the test dataset into a new anonymous test dataset (Step 8). In the case of DMPD, the rule of partitioning must be applied on an original feature set. For generalization and suppression-based techniques, generalization or suppression rules are applied to different original values in the dataset.

An inducer is trained (Step 9) over the anonymous train dataset to generate a classifier (Step 10). Finally the classifier is used to estimate the performance of the algorithm over the anonymous test dataset (Step 11).

The same cross-validation folds were implemented for all the algorithms in the pairwise comparison that we conducted. In each such comparison we used the combined 5×2 CV *F*-test to accept or reject the hypothesis that the two methods (DMPD vs. TDS, DMPD vs. TDR) have the same error rate with a 0.95 confidence level.

It should be noted that the above experimental process is different from the process used by Fung et al. [21]. According to his experimental design, the generalization of the original dataset takes place in the first step. Afterwards, the train and test datasets are split. Thus, the estimated variance of the cross-validation solely measures the inducer's variance and not the variance due to applying *k*-anonymity constraints on the underlying dataset.

5.2. Datasets

Privacy-preserving classification algorithms are usually evaluated only on the Adult dataset which has become a commonly used benchmark for *k*-anonymity [65,22,20]. Recently Fung et al. [21] used a German credit dataset to evaluate

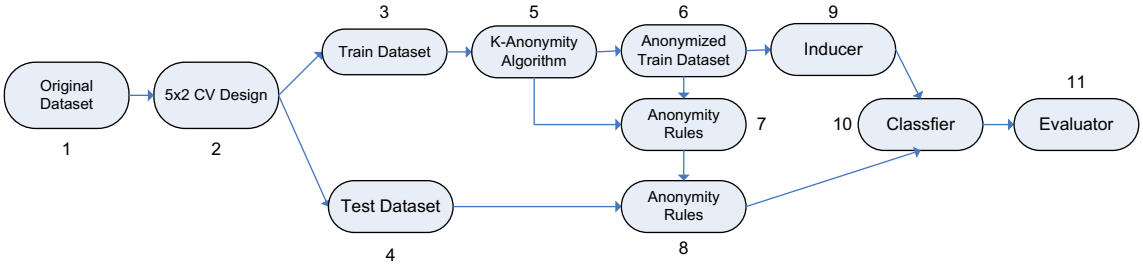

Fig. 9. The experimental process.

the TDR algorithm. In our experimental study we used an additional seven datasets that were also selected from the UCI Machine Learning Repository [45] and which are widely used by the machine-learning community for evaluating learning algorithms. An additional dataset, drawn from a real-world case study performed on commercial banks, is described below. The datasets vary across such dimensions as the number of target feature classes, instances, input features and their type (nominal, numeric).

5.3. Real-world case study dataset

The set of datasets that we used also includes a real-world dataset named "Portfolio" that contains various data from the investment department of a bank. The dataset was designed to predict the prospect of customers buying a new type of investment product. Initial operational data included the history of proposals that were made to customers and their responses measured in terms of product purchases within half a year from the date of the proposal. The initial dataset contains various groups of input features:

- Customer demographic and geographic data – age, sex, zip code, number of children, level of income, etc.
- Customer activity in different banking fields – number of loans and deposits, liquid account balance, etc.
- Customer's current investment position – ownership of other investment products at the bank, etc. These features were extracted within a time frame of one year prior to this study.
- Behavioral measures – volume of financial transitions, response to marketing campaigns, etc.

The initial dataset included 35 features. After data preparation and exploration, we took the 28 most suitable features: 11 continuous and 17 nominal. The dataset contains about 5000 instances with approximately equal distribution of target feature values.

5.4. Algorithms used

For the anonymity phase (Step 5 in Fig. 9), we compared the DMPD algorithm to TDS [22] and TDR [21] in terms of classification accuracy. In addition, we considered performing a comparison with a GA-based algorithm [28]. However, since we could not access the algorithm's implementation, we were unable to meaningfully compare Iyengar's algorithm with ours. It is important to note that the TDS algorithm we compared to the DMPD algorithm provided no worse results than the GA algorithm, at least from the partial comparison performed earlier [22].

It should also be noted that the TDS and TDR experiments were based on using the original software obtained from the algorithms' inventors. Because TDS requires the user to provide generalization taxonomy, we could use it only on datasets for which generalization taxonomy was previously provided by the algorithm's inventor. Specifically, we compared DMPD to TDS on the Adult dataset using the earlier generalization taxonomy that had been proposed. Because our algorithm works with no such taxonomy, for other datasets we used the two-level taxonomy where the lower level contains feature domain values and the higher level contains "Any" value. In other words, if the TDS algorithm considers such a feature generalization, it only results in suppressing all feature values. DMPD is compared to TDR by choosing the suppress option of the TDR algorithm in the absence of a generalization taxonomy.

K-anonymity quasi-identifiers constraints were formed as proposed by the TDS inventors [22]. In this work, to ensure that the anonymization algorithm is operating on features that impact classification, the quasi-identifiers subsets contain the top N features ranked by the C4.5 classifier. Accordingly, the structure of the decision tree's most important feature is the feature at the top of the tree. The next important feature can be evaluated by the C4.5 classifier in the absence of a previous top feature. Following this procedure recursively, the top nine features were defined. Three quasi-identifier constraints were defined by selecting the top 5, 7 and 9 features. A 10-CV procedure was used to evaluate the tree. All remaining features are included in the training set but are treated as non-quasi identifiers.

For the induction phase (Step 9 in Fig. 9), we examined the following base induction algorithms: naïve Byes and C4.5 [51]. The C4.5 algorithm was selected because it is considered a state-of-the-art decision tree algorithm that is widely used in many other comparative studies. Naïve Byes was selected due to its simplicity and the fact that it uses a quite different learning scheme.

For DMPD, TDS and TDR, 12 experiments were performed for each dataset and induction algorithm; each of three quasi-identifier constraints was tried with four different *k*-anonymity levels.

DMPD implementation and all experiments were performed in WEKA, a Java-based environment [66]. The experiments with C4.5 and naïve Byes took place using WEKA implementation of the algorithms along with their default settings.

5.5. Effect of k-anonymity constraints on classification accuracy

In this section we analyze the effect of the value of k (anonymity level) on classification accuracy. Table 1 shows the accuracy results obtained by the proposed algorithm for four different values of k for various datasets using different inducers. In this section we take the top nine features as quasi-identifiers (top eight for *nursery* and *pima* datasets). Note that the column with $k = 1$ represents the DMPD algorithm result (i.e. when no *k*-anonymity constraints were implied) enabling us to exam-

Table 1
Accuracy vs. anonymity for DMPD algorithm.

Dataset	Inducer	*k*-Anonymity level					
Adult	*k*	Baseline	1	50	100	200	500
	C4.5	85.58 ± 0.51	86.59 ± 0.22	83.03 ± 0.10*	83.00 ± 0.10*	82.63 ± 1.15*	82.99 ± 0.08*
	Naïve Bayes	82.68 ± 0.27	84.87 ± 0.20	81.81 ± 0.13*	81.41 ± 0.79*	81.47 ± 0.77*	81.19 ± 1.11*
	k	Baseline	1	10	20	30	50
Credit	C4.5	85.53 ± 2.45	85.82 ± 1.54	86.71 ± 1.15	85.08 ± 1.75	85.29 ± 1.99	85.66 ± 2.19
	Naïve Bayes	76.82 ± 1.87	85.66 ± 1.89	85.94 ± 1.42	86.46 ± 1.46	85.66 ± 2.45	85.45 ± 2.98
Vote	C4.5	96.23 ± 2.41	96.35 ± 1.95	96.26 ± 2.09	96.35 ± 2.38	95.04 ± 4.06	96.70 ± 1.91
	Naïve Bayes	90.78 ± 1.98	96.26 ± 2.05	96.43 ± 1.85	96.52 ± 2.09	96.61 ± 1.85	96.52 ± 1.97
Wisconsine	C4.5	93.28 ± 1.83	97.00 ± 0.90	95.97 ± 1.40*	94.88 ± 1.00*	95.09 ± 1.71	95.23 ± 1.77
	Naïve Bayes	93.28 ± 1.78	96.11 ± 1.01	95.27 ± 1.57	94.77 ± 2.09	93.89 ± 2.03	93.04 ± 1.09*
German	C4.5	69.88 ± 2.12	73.23 ± 7.44	71.86 ± 2.35	70.78 ± 2.22	71.12 ± 2.51	63.01 ± 16.49
	Naïve Bayes	73.81 ± 1.13	75.03 ± 2.03	73.79 ± 2.37	73.71 ± 2.30	71.26 ± 2.59	69.84 ± 1.87*
Heart	C4.5	74.89 ± 2.97	84.03 ± 6.71	79.18 ± 4.14	78.96 ± 3.53	77.31 ± 3.76	74.78 ± 3.83
	Naïve Bayes	84.89 ± 3.44	82.61 ± 4.37	79.93 ± 4.43	77.69 ± 4.84	78.06 ± 4.75	74.93 ± 3.38*
Portfolio	C4.5	74.82 ± 0.98	76.41 ± 0.56	72.04 ± 1.34*	71.61 ± 2.37*	72.07 ± 1.13*	71.76 ± 1.57*
	Naïve Bayes	58.96 ± 1.81	75.46 ± 0.71	68.74 ± 2.67	69.11 ± 1.84*	68.58 ± 2.72*	68.46 ± 2.47*
Cmc	C4.5	50.90 ± 1.61	56.59 ± 2.73	52.53 ± 4.64	51.51 ± 4.53	51.02 ± 4.19	48.30 ± 2.30
	Naïve Bayes	48.78 ± 1.37	54.04 ± 2.57	54.04 ± 3.33	52.73 ± 4.69	52.12 ± 3.49	48.07 ± 2.00*
Pima diabetes	C4.5	72.09 ± 2.20	78.20 ± 2.05	77.75 ± 1.53	77.62 ± 2.66	76.79 ± 3.01	75.85 ± 1.87
	Naïve Bayes	75.20 ± 2.67	77.75 ± 1.93	77.75 ± 2.27	78.02 ± 2.34	78.41 ± 2.45	78.30 ± 2.26
	k	Baseline	1	50	100	150	200
Nursery	C4.5	96.16 ± 0.31	97.13 ± 0.52	92.64 ± 1.15*	91.15 ± 0.81*	90.86 ± 1.27*	88.91 ± 4.79
	Naïve Bayes	90.08 ± 0.40	90.25 ± 0.51	90.24 ± 0.57	90.27 ± 0.67	90.00 ± 0.77	89.46 ± 0.94

ine the effect of anonymity on the accuracy of the results. The superscript "*" indicates that the degree of accuracy of the original dataset was significantly higher than the corresponding result with a confidence level of 95%.

As expected, the results indicate that there is a tradeoff between accuracy performance and the anonymity level for most datasets. Usually, increasing the anonymity level decreases accuracy. For some datasets, the feature set partitioning approach improves baseline accuracy, even despite applying *k*-anonymity constraints such as for *vote*, *cmc*, *pima*, *wisconsine* or *nursery*. Supervised discretization, as a part of DMPD, also contributes to classification accuracy, for example, in the *heart* dataset. These above results are marked out for both inducer types.

5.6. Scalability analysis

The aim of this section is to examine the DMPD's ability to handle expanding datasets in an elegant manner. We tested for scalability using the procedure that Fung et al. [21] proposed to measure the runtime costs of algorithms on large datasets. We based our scalability test on a German dataset. For this purpose, the original dataset containing 1000 records was expanded as follows: for every original instance q, we added $\sigma - 1$ variations where σ is a scale factor. Together with all original instances, the enlarged dataset has $\sigma \times 1000$ instances. Each instance variation was generated by randomly drawing appropriate feature values (x_{qi}, y_q) from the feature domain $(dom(a_i), dom(y))$.

We conducted all experiments on a hardware configuration that included a desktop computer implementing a Windows XP operating system with Intel Pentium 4–2.8 GHz, and 1 GB of physical memory.

Table 2 presents the average time (in minutes) measured on 10 runs for various values of σ, a quasi-identifier that includes top 5 features and a *k*-anonymity level = 10. The time, including model generation time, reflects the runtime cost of a J4.8 inducer in a WEKA package and is beyond our control.

Fig. 10 shows that the execution time is almost linear in the number of records. This confirms the DMPD's ability to handle a growing search space since an increase in train dataset size leads to more partitions that the algorithm must consider.

5.7. GA settings for DMPD

A major problem with genetic algorithms is their sensitivity to the selection of various parameters such as population size, maximum number of generations, crossover and mutation probabilities. This sensitivity is due to a degree of randomness in searching for an optimal solution. We set crossover and mutation values to probabilities that are similar to those found in the GA literature [24,27]. In this section we try to estimate if better results can be obtained from a larger population

Table 2
Scalability analysis for DMPD.

Scalability factor	Algorithm runtime (min)	Dataset size
1	1.41	1000
5	6.55	5000
10	13.90	10,000
15	24.48	15,000
20	32.76	20,000

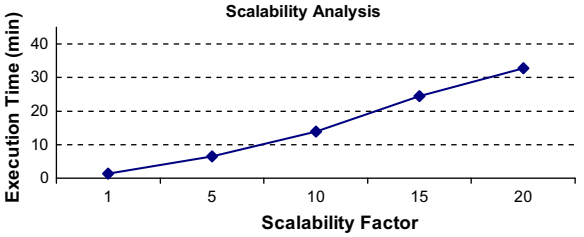

Fig. 10. Scalability trend in the extended German dataset.

size or a larger number of generations. Obviously, we have here a probable tradeoff between, better partitioning structure and computational costs of the algorithm due to our increasing the GA settings.

Our default settings for DMPD were 100 generations and 50 individuals in a population. Here we present some experiments from the *german* and *cmc* datasets. We present two of 12 experiments that were performed on the dataset with a C4.5 inducer with a minimum of *k*-anonymity constraints (top5 and $k = 10$). The number of generations considered is 50, 100, 150 and 200. Test runs were carried out on two populations numbering 50 and 100 individuals, respectively.

As Fig. 11 shows, increasing GA settings beyond 100 generations and 50 partitionings does not significantly improve classification accuracy. Similar behavior is true for other datasets used in the experimental study. The evidence points to the DMPD's ability to achieve good feature set partitioning with a relatively low number of generations and population size.

5.8. Multiobjective DMPD evaluation

In this section we introduce experimental results of the DMPD for multiobjective decision-making in privacy-preserving data mining. We follow a method presented by Grunert da Fonseca et al. [26] for experimentally evaluating multiobjective GA. Such an evaluation is performed on the basis of multiple, independent optimization runs and by building attainment functions. The concept of empirical attainment function is used in Pareto frontier estimation by building a specific type of *summary attainment surface* [33] on objective space. In our case, there are two objectives: partitioning anonymity level and classification accuracy. The w*orst attainment surface* describes the region that was attained by all optimization runs and provides a pessimistic estimation for the true Pareto frontier. The region, attained by only one optimization run, presents us with the *best attainment surface* and this is an optimistic estimation for a true Pareto frontier. *Median attainment surface*, attained by 50% of optimization runs, provides a reasonable estimation for the true Pareto frontier [17,33].

In Fig. 12 we present three types of empirical surfaces for true Pareto frontier estimation created by an algorithm for an Adult dataset with top5 and top9 quasi-identifier constraints and a C4.5 inducer.

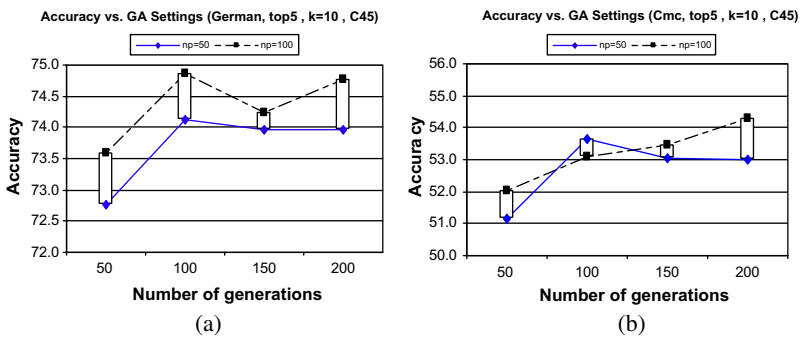

Fig. 11. GA settings vs. accuracy for *german* and *cmc* datasets.

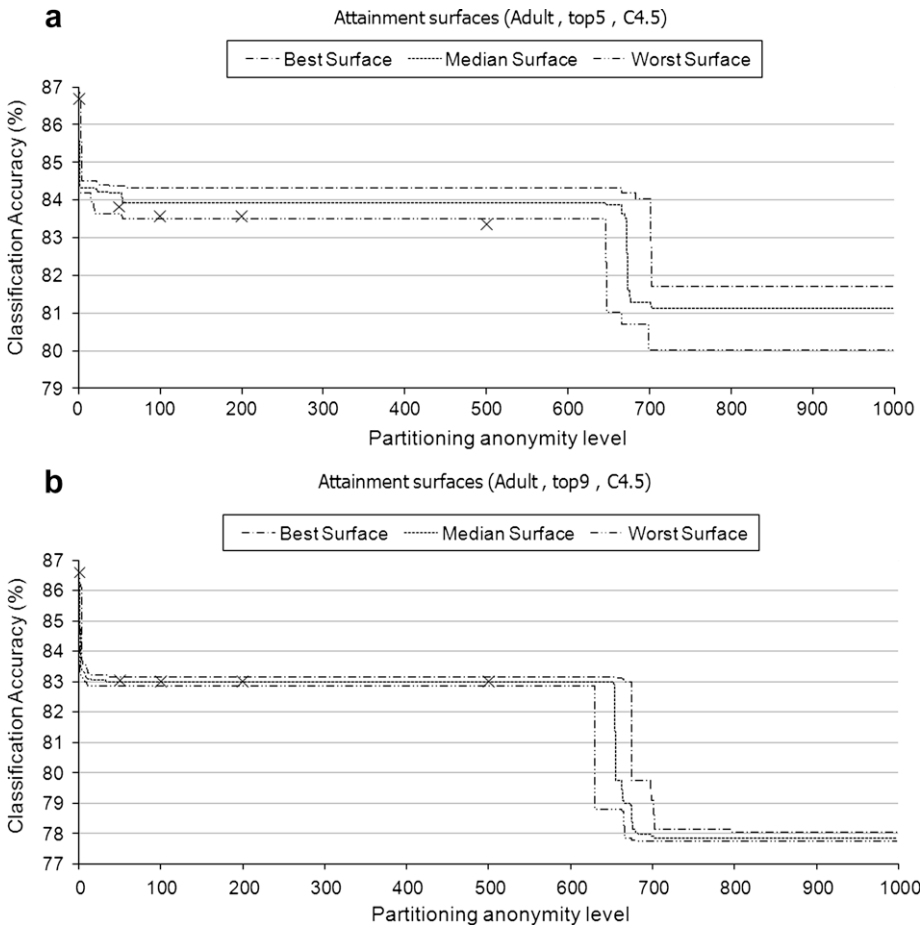

Fig. 12. Accuracy vs. *k*-anonymity level tradeoff.

From the graphs we conclude that there is a clear tradeoff between the two objectives (Fig. 12a). The graphs provide an important tool for a data owner. If he can agree to a *k*-level between 1 and 3, the classification accuracy that can be achieved is nearly 86%. A further key decision point arises near a *k*-level of 660. Classification performance from $k = 3$ to $k = 655$ is essentially the same excluding some negligible decreases in accuracy in the range between $k = 4$ and $k = 35$. If these *k*-anonymity levels are insufficient for the data owner, he can advance along the frontier to an additional k level. Increasing *k* up to 1000, results in a decrease of 5% in the classification accuracy.

As shown in Fig. 12b, the DMPD forms a Pareto frontier with almost a similar degree of accuracy for the given *k*-anonymity levels (denoted by cross) presented in Section 5.5.

5.9. DMPD compared to other k-anonymity algorithms

In this section we compare the proposed algorithm to other existing *k*-anonymity algorithms. Firstly, we present the results from comparing DMPD to two other algorithms based on datasets used in a previous work [22,21].

We compare the performance for three different constraints of quasi identifiers (as described in Section 5.4) and with four different levels of *k*. Table 3 presents the results obtained on the Adult dataset when comparing DMPD with TDS. Table 4 compares DMPD and TDR. The superscript "*" indicates that the accuracy of the corresponding algorithm was significantly higher (with the same *k*, the same base inducer and the same dataset) with a confidence level of 95%. The results of the experimental study are encouraging. They indicate that there are only two cases where either TDS or TDR is more accurate than DMPD. In seven experiments DMPD is significantly better than TDS. There is no case where TDR is significantly more accurate than DMPD, but DMPD is better in most experiments. As expected, on the dataset, DMPD outperforms TDR or TDS, mainly, for relatively high levels for quasi-identifiers or *k*-anonymity. This can be explained by the fact that DMPD is available "to relax" *k*-anonymity restrictions by feature set partitioning.

Tables 5 and 6 present the results obtained on the German dataset when comparing DMPD with TDS and TDR, respectively. The results indicate that TDS and DMPD perform similarly. DMPD was slightly better than TDR, and 2 of 24 experiments show significant DMPD superiority.

Table 3
Adult dataset – comparing DMPD to TDS.

Dataset – Adult	*k*-Level	C45				NB			
		50	100	200	500	50	100	200	500
DMPD	top5	83.82 ± 0.54	83.56 ± 0.33	83.57 ± 0.50*	83.35 ± 0.68	81.83 ± 0.23	81.74 ± 0.19	81.27 ± 0.86	81.54 ± 0.62
TDS		84.79 ± 0.12	83.88 ± 1.19	82.35 ± 0.39	82.57 ± 0.51	82.99 ± 0.30*	82.25 ± 1.12	80.85 ± 0.16	81.03 ± 0.22
DMPD	top7	83.77 ± 0.30	83.30 ± 0.93	83.15 ± 0.98	83.19 ± 0.76	81.76 ± 0.25	80.88 ± 1.05	81.73 ± 0.13*	81.69 ± 0.21*
TDS		84.72 ± 0.13*	83.92 ± 1.02	81.89 ± 1.50	82.48 ± 0.38	82.88 ± 0.27*	81.88 ± 1.15	80.45 ± 0.30	80.56 ± 0.38
DMPD	top9	83.03 ± 0.10*	83.00 ± 0.10	83.00 ± 0.13*	83.01 ± 0.08	81.81 ± 0.13*	81.41 ± 0.79	81.47 ± 0.77	81.19 ± 1.11
TDS		82.03 ± 0.15	81.77 ± 0.75	80.60 ± 0.66	81.56 ± 1.15	79.07 ± 0.32	79.32 ± 0.73	79.65 ± 0.85	79.01 ± 1.05

Table 4
Adult dataset – comparing DMPD To TDR.

Dataset – Adult	*k*-Level	C45				NB			
		50	100	200	500	50	100	200	500
DMPD	top5	83.82 ± 0.54	83.56 ± 0.33	83.57 ± 0.50*	83.35 ± 0.68	81.83 ± 0.23	81.74 ± 0.19*	81.27 ± 0.86	81.54 ± 0.62
TDR		82.70 ± 0.31	81.26 ± 1.01	80.56 ± 1.01	80.95 ± 1.53	79.85 ± 0.84	79.25 ± 0.70	79.41 ± 1.10	79.27 ± 1.21
DMPD	top7	83.77 ± 0.30*	83.30 ± 0.93*	83.15 ± 0.98	83.19 ± 0.76*	81.76 ± 0.25*	80.88 ± 1.05*	81.73 ± 0.13*	81.69 ± 0.21*
TDR		82.16 ± 0.27	80.06 ± 1.31	79.09 ± 0.47	78.81 ± 0.67	78.81 ± 1.10	78.14 ± 0.86	77.08 ± 0.49	77.17 ± 0.56
DMPD	top9	83.03 ± 0.10*	83.00 ± 0.10*	83.00 ± 0.13*	83.01 ± 0.08*	81.81 ± 0.13*	81.41 ± 0.79*	81.47 ± 0.77*	81.19 ± 1.11*
TDR		79.48 ± 0.89	79.26 ± 0.51	79.03 ± 0.20	79.06 ± 0.31	76.66 ± 1.04	76.23 ± 0.22	76.70 ± 0.42	76.01 ± 0.18

Table 5
German credit dataset – comparing DMPD to TDS.

Dataset – German	*k*-Level	C45				NB			
		10	20	30	50	10	20	30	50
DMPD	top5	74.13 ± 2.45	71.30 ± 2.87	67.56 ± 13.12	70.50 ± 1.85	74.55 ± 1.63	74.15 ± 1.96	74.41 ± 0.87	70.24 ± 2.31
TDS		70.88 ± 1.64	70.50 ± 1.15	69.54 ± 2.51	66.71 ± 2.71	73.07 ± 1.04	72.91 ± 1.22	73.03 ± 1.43	72.83 ± 1.55
DMPD	top7	70.64 ± 3.37	70.06 ± 1.66	71.72 ± 2.63	70.28 ± 1.82	73.95 ± 2.90	73.71 ± 2.00	71.22 ± 2.59	70.02 ± 2.37
TDS		70.74 ± 1.88	71.30 ± 1.95	71.14 ± 1.95	70.42 ± 1.91	72.61 ± 1.56	72.67 ± 1.33	72.55 ± 1.62	71.92 ± 1.67
DMPD	top9	71.86 ± 2.35	70.78 ± 2.22	71.12 ± 2.51	63.01 ± 16.49	73.79 ± 2.37	73.71 ± 2.30	71.26 ± 2.59	69.84 ± 1.87
TDS		71.02 ± 2.30	71.44 ± 2.00	71.40 ± 2.13	71.42 ± 1.78	72.38 ± 1.35	72.46 ± 1.22	72.40 ± 1.47	72.34 ± 1.37

Table 6
German credit dataset – comparing DMPD to TDR.

Dataset – German	*k*-Level	C45				NB			
		10	20	30	50	10	20	30	50
DMPD	top5	74.13 ± 2.45	71.30 ± 2.87	67.56 ± 13.12	70.50 ± 1.85	74.55 ± 1.63	74.15 ± 1.96	74.41 ± 0.87*	70.24 ± 2.31
TDR		68.20 ± 1.94	67.68 ± 2.68	67.37 ± 2.08	67.01 ± 1.89	70.78 ± 2.00	70.16 ± 1.54	70.16 ± 1.62	69.82 ± 1.35
DMPD	top7	70.64 ± 3.37	70.06 ± 1.66	71.72 ± 2.63	70.28 ± 1.82	73.95 ± 2.90	73.71 ± 2.00	71.22 ± 2.59	70.02 ± 2.37
TDR		68.14 ± 2.15	67.41 ± 2.74	67.68 ± 2.02	66.97 ± 2.39	70.84 ± 1.73	69.90 ± 1.65	69.78 ± 0.91	69.38 ± 1.04
DMPD	top9	71.86 ± 2.35	70.78 ± 2.22	71.12 ± 2.51	63.01 ± 16.49	73.79 ± 2.37	73.71 ± 2.30*	71.26 ± 2.59	69.84 ± 1.87
TDR		69.46 ± 1.69	68.52 ± 2.73	68.96 ± 1.83	68.22 ± 2.81	70.38 ± 1.93	69.58 ± 1.87	69.20 ± 1.97	69.12 ± 1.52

In order to conclude which algorithm performs best over multiple cases, we followed the procedure proposed in [12]. In the case of multiple classifiers, we first used the adjusted Friedman test in order to reject the null hypothesis and then the Bonferroni–Dunn test to examine whether the new algorithm performs significantly better than existing algorithms. The null-hypothesis, that all algorithms perform the same and the observed differences are merely random, was rejected with a confidence level of 95% using the adjusted Friedman test (with a statistic value of $F = 56.79$ distributed as F-distribution with 2 and 478 degrees of freedom). We proceeded with the Bonferroni–Dunn test and found that DMPD statistically outperforms TDS and TDR with a 95% confidence level (statistic values of -6.53 and -15.84 correspondingly which are normally distributed).

5.10. Discussions

The advantages of the new DMPD algorithm, based on observations from the experimental study, can be summarized as follows:

relationship	native-country	workclass
Own-child	US	State-gov
Wife	US	Private
Husband	US	Private
Own-child	US	State-gov

(a)

relationship	native-country	workclass
Own-child	US	State-gov
?	US	Private
?	US	Private
Own-child	US	State-gov

(b)

Fig. 13. Increasing flexibility for DMPD.

1. 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.
2. The DMPD is capable of dealing with different *k*-anonymity constraints and inducers without any significant effect on classification accuracy.
3. When compared to the state-of-the-art *k*-anonymity methods, DMPD classifiers provide an equivalent or slightly higher degree of accuracy.
4. 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.
5. 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.
6. DMPD can provide useful information regarding the tradeoff between *k*-anonymity constraints and classification performance by means of a Pareto frontier between these two conflicting goals.

The DMPD algorithm also has several drawbacks:

1. Over-anonymity – there are two reasons leading to this drawback:
   1.1. When a certain partitioning does not comply with the *k*-anonymity restriction, DMPD excludes it entirely from producing offspring. This might be too aggressive and might result in over-anonymity of the anonymized dataset.
   1.2. Another over-anonymity issue is due to the global method of treating numeric features where the dataset is discretized at the initial phase.
2. Non-flexibility – DMPD depends entirely on the original distribution of data since it does not try to transform categorical values. DMPD tends to produce structures with a relatively small number of features in each subset. This can cause deficient data mining results. Here the solution may lay in applying generalization or suppression techniques. To demonstrate this idea we considered some subsets of the Adult dataset (Fig. 13a). If the required *k*-level is 2 and all three attributes are in the quasi-identifier set, the "relationship" feature will be excluded from any candidate solution produced during the GA search. After suppressing some values (Fig. 13b), the "relationship" feature can enter some subsets under the *k*-anonymity constraints. In this case we get a classifier with higher classification accuracy.
3. High computational cost – scalability analysis shows the algorithm's ability to work linearly on dataset size. Decreasing in runtime since *k*-anonymity constraints were grown was expected and was seen as well in regard to the TDR or TDS algorithms. But DMPD's absolute execution time is higher than with TDR and TDS. This is due to two main procedures that were proven earlier to be slow: genetic search and wrapper-based evaluation of partitionings.

These drawbacks are not so very difficult to deal with. In regard to the first drawback, a possible solution would be to use a more sophisticated fitness function, in which a partitioning anonymity level is combined in some way with classification accuracy (see Paragraph 1.1). Regarding treating numeric features (see Paragraph 1.2), a more accurate method could be applied, combining a supervised discretization principle and an anonymity level of partitioning. In regard to the second drawback we can try to incorporate existing approach for anonymizing datasets (for example, the simplest one – tuple suppression) to obtain more algorithm's flexibility. Third drawback can be alleviated by examining different search and evaluation procedures.

6. Summary and conclusions

In this paper we presented a new method for preserving privacy in classification tasks using a *k*-anonymity framework. The proposed method was designed to work with no prior knowledge and with any inducer. Compared to existing state-of-the-art methods, the new method also shows a higher predictive performance on a wide range of datasets from different domains.

Additional issues to be further studied include:

- Examining DMPD with other classification algorithms
- Revising DMPD to overcome its existing drawbacks as noted above
- Examining the possibility of using DMPD along with known generalization/suppression-based methods that could result in improved data mining results in terms of classification accuracy and discovered patterns. Here the simple tuple suppression (TS) can be evaluated. The technique has not yet been applied in any state-of-art algorithm [8].
- Extending the proposed method for treating *k*-anonymity model leaks and proposed solutions for them. One solution is based on the *l*-diversity framework [40].
- Examining DMPD with overlapping partitioning structures. This evaluation can be valuable in increasing the algorithm's flexibility.
- Extending DMPD to handle multiobjective optimization with different quasi-identifier sets. The main idea is to eliminate the *k*-anonymity model assumption; the quasi-identifier set is determined prior to performing data anonymization. One of the solutions can include assigning a probability to each feature to be used for individual information discloser. Higher probabilities assume a higher degree of privacy. These probabilities can be used to produce an aggregated measure of overall privacy that must be maximized against data mining performance to form an appropriate Pareto frontier. Such a study may, for example, contribute to evaluating a tradeoff between a feature as a part of a quasi-identifier and classification accuracy.
- Extending the proposed method to other data mining tasks (such as clustering and association rules). The work could involve applying other types of decomposition, such as space or sample decomposition [52].

Acknowledgments

The authors gratefully thank Benjamin C.M. Fung, Ke Wang and Philip S. Yu for providing their proprietary software packages for evaluating TDS and TDR algorithms.

Also we would like to acknowledge E. Zitzler for providing information about SPEA-II selection algorithm for GA-based multiobjective optimization. Many thanks are owed to Arthur Kemelman. He has been a most helpful assistant in proofreading and improving the manuscript.

References

[1] C.C. Aggarwal, On *k*-anonymity and the curse of dimensionality, in: Proc. of the 31th VLDB Conference, VLDB Endowment, 2005, pp. 901–909.
[2] A. Agrawal, R. Srikant, Privacy preserving data mining, SIGMOD Record 29 (2) (2000) 439–450.
[3] E. Alpaydin, Combined 5×2 CV *F*-test for comparing supervised classification learning classifiers, Neural Computation 11 (1999) 1975–1982.
[4] S.S Anand, D.A. Bell, J.G. Hughes, The role of domain knowledge in data mining, in: Proc. of the Fourth International Conference on Information and Knowledge Management, ACM, New York, NY, pp. 37–43.
[5] M. Atzori, F. Bonchi, F. Giannotti, D. Pedreschi, *k*-Anonymous patterns, in: A. Jorge et al. (Eds.), Proc. of the Nineth European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD05, Lecture Notes in Computer Science, vol. 3721, Springer-Verlag, 2005, pp. 10–21.
[6] R.J. Bayardo, R. Agrawal, Data privacy through optimal *k*-anonymization, in: Proc. of the 21st IEEE International Conference on Data Engineering, ICDE05, IEEE Computer Society, Washington, DC, 2005, pp. 217–228.
[7] L. Cao, C. Zhang, Domain-driven, actionable knowledge discovery, Intelligent Systems 22 (4) (2007) 78–88.
[8] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, P. Samarati, *k*-Anonymity, in: T. Yu, S. Jajodia (Eds.), Secure Data Management in Decentralized Systems, Springer, Berlin Heidelberg, 2007, pp. 323–353.
[9] C. Clifton, M. Kantarcioglu, J. Vaidya, Defining privacy for data mining, in: H. Kargupta et al. (Eds.), Proc. of the National Science Foundation Workshop on Next Generation Data Mining, Baltimore, Maryland, 2002, pp. 126–133.
[10] S. Cohen, L. Rokach, O. Maimon, Decision-tree instance-space decomposition with grouped gain-ratio, Information Sciences 177 (17) (2007) 3592–3612.
[11] U. Dayal, N. Goodman, R.H. Katz, An extended relational algebra with control over duplicate elimination, in: J.D. Ullman, A.V. Aho (Eds.), Proc. of the First ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, ACM, New York, NY, 1982, pp. 117–123.
[12] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[13] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal of the European Communities I(281) (1995), 31–55.
[14] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: A. Prieditis, S. Russell (Eds.), Proc. 12th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1995, pp. 194–202.
[15] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, in: Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1996, pp. 1–31.
[16] M. Feingold, M. Jeffords, M. Leahy, Data Mining Moratorium Act of 2003, US Senate Bill (proposed), 2003.
[17] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: S. Forrest (Ed.), Proc. of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423.
[18] A. Freitas, Evolutionary algorithms for data mining, in: O. Maimon, L. Rokach (Eds.), The Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 435–467.
[19] A. Friedman, A. Schuster, R. Wolff, *k*-Anonymous decision tree induction, in: J. Furnkranz (Ed.), Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD06, Lecture Notes in Computer Science, vol. 4213, Springer-Verlag, 2006, pp. 151–162.
[20] A. Friedman, A. Schuster, R. Wolff, Providing *k*-anonymity in data mining, VLDB 17 (4) (2008) 789–804.
[21] B.C.M. Fung, K. Wang, P.S. Yu, Anonymizing classification data for privacy preservation, IEEE Transactions on Knowledge and Data Engineering 19 (5) (2007) 711–725.
[22] B.C.M. Fung, K. Wang, P.S. Yu, Top-down specialization for information and privacy preservation, in: Proc. of the 21st IEEE International Conference on Data Engineering, ICDE05, IEEE Computer Society, Washington, DC, 2005, pp. 205–216.
[23] M.S. Gibbs, G.C. Dandy, H.R. Maier, A genetic algorithm calibration method based on convergence due to genetic drift, Information Sciences 178 (14) (2008) 2857–2869.

[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Boston, Maryland, 1989.
[25] S. Grumbach S, T. Milo, Towards tractable algebras for bags, Journal of Computer and System Sciences 52 (3) (1996) 570–588.
[26] V. Grunert da Fonseca, C.M. Fonseca, A.O. Hall, Inferential performance assessment of stochastic optimizers and the attainment function, in: E. Zitzler et al. (Eds.), Proc. of Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), Lecture Notes in Computer Science, vol. 1993, Springer-Verlag, 2001, pp. 213–225.
[27] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley, 2004.
[28] V.S. Iyengar, Transforming data to satisfy privacy constraints, in: Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, 2002, pp. 279–288.
[29] D.F. Jones, S.K. Mirrazavi, M. Tamiz, Multiobjective meta-heuristics: an overview of the current state-of-the-art, European Journal of Operational Research 137 (1) (2002) 1–9.
[30] M. Kantarcioglu, J. Jin, C. Clifton, When do data mining results violate privacy?, in: Proc of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, 2004, pp. 599–604.
[31] S.W. Kim, S. Park, J.I. Won, A.W. Kim, Privacy preserving data mining of sequential patterns for network traffic data, Information Sciences 178 (3) (2008) 694–713.
[32] K. Kisilevich, L. Rokach, Y. Elovici, B. Shapira, Efficient multidimensional suppression for *k*-anonymity, IEEE Transactions on Knowledge and Data Engineering 22 (3) (2010) 334–347. Mar.
[33] J. Knowles, A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers, in: Proc. of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA V), IEEE Computer Society, Washington, DC, 2005, pp. 552–557.
[34] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.
[35] A. Konaka, D.W. Coitb, A.E. Smithc, Multi-objective optimization using genetic algorithms: a tutorial, Reliability Engineering and System Safety 91 (2006) 992–1007.
[36] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (2000) 25–41.
[37] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Incognito: efficient full domain *k*-anonymity, in: Proc. of the ACM SIGMOD Conference on Management of Data, ACM, New York, NY, 2005, pp. 49–60.
[38] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Mondrian multidimensional *k*-anonymity, in: Proc. of the International Conference on Data Engineering, ICDE'06, IEEE Computer Society, Washington, DC, 2006, p. 25.
[39] T. Li, N. Li, Optimal *k*-anonymity with Flexible Generalization Schemes through Bottom-up Searching, in: Proceedings of the Sixth IEEE International Conference on Data Mining – Workshops, IEEE Computer Society, Washington, DC, 2006, pp. 518–523.
[40] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, l-Diversity: privacy beyond *k*-anonymity, in: L. Liu et al. (Eds.), Proc. 22nd IEEE International Conference on Data Engineering, IEEE Computer Society, Washington, DC, 2006, p. 24.
[41] O. Maimon, L. Rokach, Data mining by attribute decomposition with semiconductors manufacturing case study, in: D. Braha (Ed.), Data Mining for Design and Manufacturing: Methods and Applications, Kluwer, London, 2001, pp. 311–336.
[42] O. Maimon, L. Rokach, Improving supervised learning by feature decomposition, in: T. Eiter, K. Schewe (Eds.), Proc. of the Second International Symposium on Foundations of Information and Knowledge Systems, Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 178–196.
[43] M. Meints, J. Moller, Privacy preserving data mining – a process centric view from a European perspective, Available online at http://www.fidis.net, 2004.
[44] E. Menahem, L. Rokach, Y. Elovici, Troika – an improved stacking schema for classification tasks, Information Sciences 179 (24) (2009) 4097–4122.
[45] C.J. Merz, P.M. Murphy, UCI Repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998.
[46] A. Meyerson, R. Williams, On the complexity of optimal *k*-anonymity, in: Proc. of the 23rd ACM SIGMOD-SIGCAT-SIGART Symposium, ACM, New York, NY, 2004, pp. 223–228.
[47] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[48] M.E. Nergiz, C. Clifton, Thoughts on *k*-anonymization, Data and Knowledge Engineering 63 (3) (2007) 622–645.
[49] S.R.M. Oliveira, O.R. Zaïane, Toward standardization in privacy-preserving data mining, in: Proc. of the Third Workshop on Data Mining Standards, ACM, New York, NY, 2004, pp. 7–17.
[50] Privacy online – OECD guidance on policy and practice, Available online at http://www.oecd.org, 2003.
[51] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman, San Francisco, CA, 1993.
[52] L. Rokach, Decomposition methodology for classification tasks – a meta decomposer framework, Pattern Analysis and Applications 9 (2) (2006) 257–271.
[53] L. Rokach, Genetic algorithm-based feature set partitioning for classification problems, Pattern Recognition 41 (5) (2008) 1693–1717.
[54] L. Rokach, O. Maimon, Feature set decomposition for decision trees, Journal of Intelligent Data Analysis 9 (2) (2005) 131–158.
[55] L. Rokach, O. Maimon, Theory and application of feature decomposition, in: Proc. of the First IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, 2001, pp. 473–480.
[56] P. Samarati, L. Sweeney, Generalizing data to provide anonymity when disclosing information, in: Proc. of the 17th ACM SIGMOD-SIGCAT-SIGART, PODS '98, ACM, New York, NY, 1998, p. 188.
[57] P. Samarati, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.
[58] D. Shah, S. Zhong, Two methods for privacy preserving data mining with malicious participants, Information Sciences 177 (23) (2007) 5468–5483.
[59] P.K. Sharpe, R.P. Glover, Efficient GA based techniques for classification, Applied Intelligence 11 (1999) 277–284.
[60] Standard for privacy of individually identifiable health information, Federal Register 67(157) (2002), 53181–53273. Available from: <www.ucdmc.ucdavis.edu/compliance/pdf/combinedregtext.pdf>.
[61] L. Sweeney, Achieving *k*-anonymity privacy protection using generalization and suppression, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10 (5) (2002) 571–588.
[62] L. Sweeney, *k*-anonymity: a model for projecting privacy, International Journal on Uncertainty, fuzziness and Knowledge-based Systems 10 (5) (2002) 557–570.
[63] H. Takeuchi, L.V. Subramaniam, T. Nasukawa, S. Roy, Getting insights from the voices of customers: conversation mining at a contact center, Information Sciences 179 (11) (2009) 1584–1591.
[64] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, Y. Theodoridis, State-of-the-art in privacy preserving data mining, ACM SIGMOD Record 3 (1) (2004) 50–57.
[65] K. Wang, P.S. Yu, S. Chakraborty, Bottom-up generalization: a data mining solution to privacy protection, in: Proc. of the Fourth IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, 2004, pp. 249–256.
[66] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed., Morgan Kaufman, San Francisco, CA, 2005.
[67] C. Yao, S. Wang, S. Jajodia, Checking for *k*-anonymity violation by views, In: Proc. of the 31st international Conference on Very Large Data Bases, VLDB Endowment, 2005, pp. 910–921.
[68] J. Zhang, J. Zhuang, H. Du, S. Wang, Self-organizing genetic algorithm based tuning of PID controllers, Information Sciences 179 (7) (2009) 1007–1018.
[69] S. Zhong, Privacy-preserving algorithms for distributed mining of frequent itemsets, Information Sciences 177 (2) (2007) 490–503.
[70] S. Zhong, Z. Yang, T. Chen, *k*-anonymous data collection, Information Sciences 179 (17) (2009) 2948–2963.
[71] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173–195.
[72] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.
[73] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.