ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Ezgi ZORARPACI

PRIVACY PRESERVING RULE-BASED CLASSIFIERS USING MODIFIED ARTIFICIAL BEE COLONY OPTIMIZATION ALGORITHM

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PRIVACY PRESERVING RULE-BASED CLASSIFIERS USING MODIFIED ARTIFICIAL BEE COLONY OPTIMIZATION ALGORITHM

Ezgi ZORARPACI

PhD THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Philosophy by the board of jury on 05/07/2019

……………………………...	…………………………………	…………………………
Prof. Dr. Selma Ayşe ÖZEL	Assoc. Prof. Dr. Umut ORHAN	Assoc.Prof.Dr. Serdar YILDIRIM
SUPERVISOR	MEMBER	MEMBER

……………………………………	……………………………….
Prof.Dr. Derviş KARABOĞA	Assist. Prof. Dr. Buse Melis ÖZYILDRIM
MEMBER	MEMBER

This PhD Thesis is written at the Computer Engineering Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Note: The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

PhD THESIS

PRIVACY PRESERVING RULE-BASED CLASSIFIERS USING MODIFIED ARTIFICIAL BEE COLONY OPTIMIZATION ALGORITHM

Ezgi ZORARPACI

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Prof. Dr. Selma Ayşe ÖZEL
Prof. Dr. Yücel SAYGIN
Year: 2019, Pages: 128
Jury : Prof. Dr. Selma Ayşe ÖZEL
: Assoc. Prof. Dr. Umut ORHAN
: Assoc. Prof. Dr. Serdar YILDIRIM
: Prof. Dr. Derviş KARABOĞA
: Assist. Prof. Dr. Buse Melis ÖZYILDIRIM

Privacy preserving data mining is a hot research field for data mining. The aim of privacy preserving data mining is to prevent the leakage of the sensitive information of individuals while performing data mining techniques. Classification task is one of the most studied fields in data mining hence in privacy preserving data mining as well. On the other hand, differential privacy is a powerful privacy guarantee that determines privacy leakage ratio by using ϵ parameter and enables researchers to mine data which includes sensitive information. Although the success of the rule-based classifiers using meta-heuristics such as Ant-Miner etc. in data mining has been demonstrated, any implementation of these classification algorithms with differential privacy has not been proposed in the literature to our best knowledge. Motivated by this, implementations of the rule-based classification algorithms by using meta-heuristics with differential privacy are performed in this thesis. According to the experimental results, the proposed rule-based classification algorithms outperform other classification techniques in the literature for low ϵ parameters (i.e., ϵ=1).

Key Words: Differentially Private Rule-Based Classifiers, Artificial Bee Colony Optimization, Privacy Preserving Classification.

ÖZ

DOKTORA TEZİ

DEĞİŞTİRİLMİŞ YAPAY ARI KOLONİSİ OPTİMİZASYON ALGORİTMASINI KULLANAN GİZLİLİK KORUYUCULU KURAL-TABANLI SINIFLANDIRICILAR

Ezgi ZORARPACI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Selma Ayşe ÖZEL
Prof. Dr. Yücel SAYGIN
Yıl: 2019, Sayfa: 128
Jüri : Prof. Dr. Selma Ayşe ÖZEL
: Assoc. Prof. Dr. Umut ORHAN
: Assoc. Prof. Dr. Serdar YILDIRIM
: Prof. Dr. Derviş KARABOĞA
: Assist. Prof. Dr. Buse Melis ÖZYILDIRIM

Veri madenciliğinde verilerin gizliliğin korunması yeni bir araştırma alanıdır. Gizlilik korumalı veri madenciliğinin amacı veri üzerinde veri madenciliği tekniklerini gerçekleştirirken aynı zamanda da kişilerin hassas bilgilerinin sızmasını engellemektir. Sınıflandırma veri madenciliğinin en çok çalışılan konularından biridir ve bu nedenle gizlilik koruyuculu veri madenciliği alanında da popüler olmuştur. Diferansiyel gizlilik, gizlilik sızıntısının oranını ϵ parametresi kullanarak belirleyen ve araştırmacılara hassas bilginin bulunduğu veriyi analiz etme imkânı sağlayan güçlü bir gizlilik garantisidir. Literatürde Ant-Miner gibi meta-sezgisel kullanan kural tabanlı sınıflandırıcılar oldukça başarılı olmasına rağmen, bu algoritmaların diferansiyel gizlilik ile ilgili herhangi bir uygulaması gerçekleştirilmemiştir. Bu nedenle, bu tezde kural tabanlı sınıflandırıcıların meta-sezgisel algoritmalar kullanılarak diferansiyel gizlilik ile uygulamaları gerçekleştirilmektedir. Önerilen kural tabanlı sınıflandırma algoritmaları küçük ϵ değerleri için (ϵ=1) literatürde bulunan diğer sınıflandırma yöntemlerinden daha iyi bir performans göstermiştir.

Anahtar Kelimeler: Diferansiyel olarak gizli kural tabanlı sınıflandırıcılar, Yapay Arı Kolonisi Optimizasyonu, Gizlilik koruyuculu sınıflandırma.

EXTENDED ABSTRACT

Data mining is the process of exploring the beneficial information from the data. Although data mining techniques extract the useful knowledge, it may cause the security threats since the sensitive or private information shows up from this extracted knowledge. Consequently privacy preserving data mining which is a sub-field of data mining has been emerged, and its goal is to protect the privacy of individuals while making possible to apply data mining techniques.

Many privacy preserving methods to analyse sensitive data have been studied for years, which covers perturbation of output or data (Adam and Worthmann, 1989), secure multiparty computation (Lindell and Pinkas, 2002; Goldreich, 2004), and some anonymization techniques such as *k*-anonymity, *l*-diversity, *t*-closeness (Samarati and Sweeny 1998; Samarati and Sweeny 1998; Samarati, 2001; Machanavajhala et al., 2007; Li et al., 2007, Mendes and Vilela, 2017).

Recently, differential privacy has been proposed to provide security to the databases by reducing the probability for the disclosure of the sensitive information of records. It ensures a formal and strong privacy guarantee and it asserts that the output of a function running on a database does not entirely depend on any record since yielding of the same output is highly probable even if an instance is or not in the database (Dwork et al., 2006; Dwork, 2008; Dwork and Roth, 2014). Therefore, differential privacy has been broadly used in the studies of privacy preserving data mining in the literature. For instance, some differentially private logistic regression, differentially private *k*-means clustering, differentially private classification algorithms based-on decision trees, random trees, random forests, Naïve Bayes, *k*-NN have been proposed in the literature (Chaudhuri and Monteleoni, 2008; Blum et al., 2005; Jagannathan et al., 2012; Jagannathan et al., 2013; Patil and Signh, 2014; Rana et al., 2015; Su et al., 2015; Fletcher and Islam, 2016; Fletcher and Islam, 2017). None of these classifiers are rule-based except

decision trees such as ID3 which is an indirect rule-based classifier. However, among all classification algorithms, rule-based classifiers, which produce relatively small number of accurate classification rules, are preferred since the classification rules are readily confirmed, highly expressive, easy to understand and interpret by humans, and the required time to classify new instances is quite short (Duch et al., 2000). Although decision trees are well-known rule-based classifiers, they have some drawbacks. Therefore, instead of applying decision tree learning algorithms, rule induction methods which mine data to discover the classification rules by exploiting meta-heuristic methods have been proposed in the literature. For example, Parpinelli et al. (2002) have developed Ant-Miner, which uses Ant Colony Optimization (ACO) to explore classification rules, and showed that Ant-Miner has competitive classification performance with respect to CN2 (Clark and Niblett, 1989) which is a well-known classification algorithm (Parpinelli et al. 2002). At the same time, some other rule discovery algorithms based-upon Artificial Bee Colony (ABC), Differential Evolution (DE), Genetic Algorithms (GA) etc. have been developed and the success of these rule-based classifiers have been demonstrated with the experimental results in the literature (Fidelis et al., 2000; Celik et al., 2011; Shukran et al. 2011; De Falco, 2013; Talebi and Abadi, 2014, Celik et al., 2016). Although these algorithms so successful, any implementation of them with differential privacy has not been proposed for performance evaluation in the literature to our best knowledge.

Motivated by this shortcoming in the literature, we propose to develop privacy preserving rule-based classifiers using ABC algorithm with the input perturbation technique of differential privacy for the first implementation in this thesis. Input perturbation technique is one of the three techniques to apply differential privacy while the others are objective and output perturbation techniques.

According to the literature, most of the differentially private classification techniques are based-on output perturbation. However, output perturbation is not suitable for the data mining algorithms which require too many appeals to the database to perform mining process (Ji et al., 2014). In similar cases, input perturbation which means perturbation of the data itself under differential privacy can be a solution to perform privacy preserving data mining (Mivule et al., 2012; Ji et al., 2014; Sarwate and Chaudhuri, 2013; Edlich, 2017). Therefore, in this thesis, the input perturbation technique in the studies of Mivule et al. (2012) is applied to provide differential privacy guarantee for the privacy preserving rule-based classifiers using ABC optimization algorithm. The performance of the proposed rule-based classifiers using ABC are compared with eleven well-known classification algorithms such as C4.5 (Quinlan, 1993), Naïve Bayes (NB) (Murphy, 2006), Bayesian Networks (BN) (Jensen, 1996), Multilayer Perceptron (MLP) (Rumelhart et al., 1986), IBk (Aha et al., 1991), Kstar (Cleary and Trigg, 1995), One Rule (1R) (Holte, 1993), PART (Frank and Witten, 1998), Random Tree (RT) (Breiman, 2001), Bagging (Brieman, 1996), and RIPPER (Cohen, 1995) over private data which is perturbed with input perturbation technique of differential privacy and non-private data separately. According to the experimental results, the proposed rule-based classifiers using ABC can be efficiently used for both of the private and non-private data.

In this thesis, the output perturbation technique of the differential privacy is adopted to build a differentially private 1R (Holte, 1993) classification algorithm as the second implementation of rule-based classifiers by using meta-heuristics. In the second implementation made in this thesis, differentially private 1R classifier is developed.To our knowledge this is the first implementation of differentially private 1R classifier. In this thesis 1R classification algorithm is preferred because it is simple, but efficient and accurate classifier, and it does not have any differentially private implementation in the literature. In this implementation, ABC-DE based feature selection method proposed in the studies of Zorarpacı and Özel (2016) is applied as a pre-processing step to reduce the required count queries sent to the differentially private database during the construction of 1R. It is

demonstrated that the accuracy values of the differentially private 1R is increased by narrowing the attribute space thanks to the proposed ABC-DE based feature selection for all values of ϵ parameter used. For the performance evaluation of the proposed differentially private 1R classifier, the privacy preserving model used for 1R is applied to build differentially private NB classifier as well since NB is utilized as a baseline for differentially private classification in the literature. The experimental results show that the proposed differentially private 1R classification algorithm is a simple but efficient privacy preserving classification algorithm.

GENİŞLETİLMİŞ ÖZET

Veri madenciliği büyük miktardaki veriden faydalı bilgilerin çıkarılması işlemidir. Ancak veri madenciliği yöntemleri ile bu faydalı bilgiler açığa çıkarılırken kişilerin hassas ya da özel bilgilerinin de ortaya çıkması güvenlik tehditlerine neden olabilmektedir. Bu nedenle gizlilik koruyuculu veri madenciliği veri madenciliğinin bir alt alanı olarak ortaya çıkmıştır ve amacı veri madenciliği tekniklerini uygulamayı mümkün kılarken aynı zamanda da kişilerin gizli ya da hassas bilgilerinin korunmasını sağlamaktır.

Hassas veriyi analiz etmek için çıkış ya da giriş sarsımı, güvenli çok parçalı hesaplama ve k-anonim, l-farklılık, t-yakınlık gibi bazı anonimleştirme tekniklerini kapsayan birçok gizlilik koruyuculu veri madenciliği yöntemi yıllardır çalışılmaktadır (Samarati and Sweeny 1998; Samarati and Sweeny 1998; Samarati, 2001; Machanavajhala et al., 2007; Li et al., 2007, Mendes and Vilela, 2017).

Son zamanlarda kayıtların hassas bilgilerinin ifşa olasılığını azaltarak veritabanlarının güvenliğini sağlamak amacıyla diferansiyel gizlilik kavramı önerilmiştir. Diferansiyel gizlilik güçlü bir gizlilik garantisi olmakla birlikte herhangi bir kaydın veritabanında bulunup bulunmaması bu veritabanında yürütülen fonksiyonun çıktısından tamamen bağımsız olduğunu iddia eder. Diferansiyel gizlilik kavramına göre bir kayıt veritabanında bulunmasa bile veritabanı üzerinde işletilen bu fonksiyonun aynı çıktıyı üretmesi son derece olasıdır (Dwork et al., 2006; Dwork, 2008; Dwork and Roth, 2014). Bu nedenle diferansiyel gizlilik literatürdeki gizlilik koruyuculu veri madenciliği çalışmalarında geniş ölçüde kullanılmıştır. Örneğin, literatürde bazı diferansiyel gizli lojistik regresyon, k-ortalama kümeleme algoritmaları ile karar ağaçları, rastgele ağaçlar, rastgele ormanlar, Naïve Bayes, k-en yakın komşuluk vb. tabanlı diferansiyel gizli sınıflandırma algoritmaları önerilmiştir (Chaudhuri and Monteleoni, 2008; Blum et al., 2005; Jagannathan et al., 2012; Jagannathan et al., 2013; Patil and Signh, 2014; Rana et al., 2015; Su et al., 2015; Fletcher and Islam,

2016; Fletcher and Islam, 2017). Ancak literatürdeki diferansiyel gizli sınıflandırma algoritmalarından ID3 karar ağacı algoritması (dolaylı kural-tabanlı sınıflandırıcı) dışındaki yöntemler kural-tabanlı değildir. Bununla birlikte tüm sınıflandırma algoritmaları arasından kural-tabanlı sınıflandırıcılar görece az sayıda kural sayısına sahip olduğu durumlarda, kuralların kolay bir şekilde doğrulanabildiği, kullanıcılar tarafından anlaşılması ve yorumlanması kolay olduğu ve yeni örnekleri kısa zamanda sınıflandırabildiğinden diğer sınıflandırıcılara tercih edilmektedir (Duch et al., 2000). Karar ağaçları iyi bilinen kural-tabanlı sınıflandırıcılar olmasına rağmen bazı dezavantajları mevcuttur. Bu nedenle karar ağaçları öğrenme algoritmalarını uygulamak yerine literatürde meta-sezgisel algoritmaları kullanarak sınıflandırma kurallarını veriden direkt olarak çıkaran kural indükleme yöntemleri önerilmiştir. Örneğin, Parpinelli ve arkadaşları (2002) tarafından sınıflandırma kurallarını çıkarmak için Karınca Kolonisi Optimizasyon algoritmasını kullanan Ant-Miner geliştirilmiş ve bu algoritmanın iyi bilinen bir sınıflandırma algoritması olan CN2 (Clark and Niblett, 1989) algoritması ile rekabet edebilen bir sınıflandırma performansı sergilediği gösterilmiştir. Yine bununla birlikte literatürde Yapay Arı Kolonisi, Diferansiyel Gelişim, Genetik Algoritmalar vb. yöntemleri kullanan bazı diğer kural çıkarım algoritmaları geliştirilmiş ve bu kural-tabanlı sınıflandırıcıların başarısı deneysel sonuçlarla ispatlanmıştır (Fidelis et al., 2000; Celik et al., 2011; Shukran et al. 2011; De Falco, 2013; Talebi and Abadi, 2014, Celik et al., 2016). Ancak bu algoritmalar oldukça başarılı olmasına rağmen, literatüre bakıldığında bu algoritmaların diferansiyel gizlilik ile herhangi bir uygulaması mevcut değildir.

Bu nedenle literatürdeki bu eksiklikten yola çıkarak, bu tezin ilk uygulamasında Yapay Arı Kolonisi optimizasyon algoritmasını ve diferansiyel gizliliğin giriş sarsımı yöntemini kullanarak gizlilik koruyuculu kural-tabanlı sınıflandırıcılar geliştirilir. Giriş sarsımı tekniği diferansiyel gizliliği uygulamak için kullanılan üç teknikten birisidir ve diğerleri ise objektif sarsımı ve çıkış sarsımı teknikleridir.

Literatüre bakıldığında diferansiyel olarak gizli sınıflandırma yöntemlerinin çoğu çıkış sarsımına dayanmaktadır. Ancak çıkış sarsımı tekniği veri madenciliği işlemini gerçekleştirmek için veritabanına çok fazla başvuruda bulunan algoritmalar için uygun değildir (Ji et al., 2014). Bu gibi benzer durumlarda gizlilik koruyuculu veri madenciliğini gerçekleştirmek için diferansiyel gizlilik garantisi altında verinin kendisinin sarsımı anlamına gelen giriş sarsımı yöntemi bir çözüm olabilmektedir (Mivule et al., 2012; Ji et al., 2014; Sarwate and Chaudhuri, 2013; Edlich, 2017). Bu nedenle bu tezde, Yapay Arı Kolonisi optimizasyon algoritmasını kullanan gizlilik koruyuculu kural-tabanlı sınıflandırıcılar için Mivule ve arkadaşlarının (2012) çalışmalarında yer alan diferansiyel gizliliği sağlamak için kullandıkları giriş sarsımı tekniği uygulanmıştır.

Önerilen Yapay Arı Kolonisi optimizasyon algoritmasını kullanan kural-tabanlı sınıflandırıcıların performansı on bir popüler sınıflandırma algoritması ile kıyaslanmıştır. Performans karşılaştırması için WEKA veri madenciliği aracından C4.5 (Quinlan, 1993), Naïve Bayes (NB) (Murphy, 2006), Bayesian Ağları (Jensen, 1996), Çok Katmanlı Algılayıcı (Rumelhart et al., 1986), k-en yakın komşuluk (Aha et al., 1991), K* (Cleary and Trigg, 1995), Tek Kural (1R) (Holte, 1993), PART (Frank and Witten, 1998), Rastgele Ağaç (Breiman, 2001), Bagging (Brieman, 1996) ve RIPPER (Cohen, 1995) sınıflandırma yöntemleri kullanılmıştır.

Algoritmaların performans karşılaştırması diferansiyel gizliliğin giriş sarsımı yöntemi ile diferansiyel olarak gizli hale getirilmiş veri ve gizli olmayan veri üzerinde ayrı ayrı gerçekleştirilmiştir. Önerilen Yapay Arı Kolonisi optimizasyon algoritmasını kullanan kural-tabanlı sınıflandırma algoritmaları küçük ϵ değerleri için (ϵ=1) literatürde bulunan diğer sınıflandırma yöntemlerinden daha iyi bir performans göstermiştir.

Literatürdeki diferansiyel olarak gizli sınıflandırma teknikleri için genel olarak diferansiyel gizliliğin çıkış sarsımı yöntemi kullanılmıştır. Bu tez çalışmasının ikinci uygulamasında, gizlilik koruyuculu kural-tabanlı sınıflandırıcı olarak diferansiyel olarak gizli 1R (Holte, 1993) algoritması gerçekleştirilmiştir. Bu uygulamada literatürdeki diğer diferansiyel olarak gizli sınıflandırma algoritmalarının uygulamış oldukları diferansiyel gizliliğin çıkış sarsımı tekniği kullanılmıştır.

Literatür incelendiğinde karar ağaçları, rastgele ağaçlar, rastgele ormanlar, Naïve Bayes, k-en yakın komşuluk gibi bazı iyi bilinen sınıflandırma algoritmalarının diferansiyel gizlilik ile uygulaması mevcut iken iyi bilinen algoritmalardan birisi olan 1R (Holte, 1993) sınıflandırma algoritmasının diferansiyel gizlilik ile herhangi bir uygulaması gerçekleştirilmemiştir.

Tek tek belirleyicilerin değerleri için sıklık tablolarını kullanarak kural çıkarımı yapan 1R sınıflandırma algoritması basit, verimli ve doğru bir sınıflandırma yöntemidir ve oldukça az sayıda kural ile sınıflandırma yapabilmektedir. Tezin bu uygulamasında, 1R sınıflandırıcısının sıklık tablolarını oluşturmak için gerekli olan diferansiyel olarak gizli veritabanlarına gönderilen sayma sorgularını azaltmak amacıyla Zorarpacı ve Özel (2016) tarafından geliştirilen Yapay Arı Kolonisi optimizasyon algoritması ve Diferansiyel Gelişim algoritmasına dayalı olan özellik seçimi yöntemi ön-işleme adımı olarak uygulanmaktadır. Deneylerde kullanılmış olan tüm ϵ parametreleri için Yapay Arı Kolonisi ve Diferansiyel Gelişim algoritmalarını kullanan özellik seçimi yönteminin özellik uzayını büyük ölçüde daraltması ile sınıflandırma doğruluk değerlerinin oldukça yükseldiği gözlemlenmiştir.

Diferansiyel olarak gizli 1R algoritmasının performansını karşılaştırmak amacıyla diferansiyel olarak gizli Naïve Bayes algoritması kullanılmıştır. Naïve Bayes diferansiyel olarak gizli sınıflandırma açısından temel ve karşılaştırmalarda sıklıkla kullanılan bir yöntemdir. Deney sonuçları incelendiğinde diferansiyel olarak gizli 1R algoritmasının diferansiyel olarak gizli Naïve Bayes algoritmasına benzer bir performans sergilediği gözlenmiştir.

ACKNOWLEDGEMENTS

I would like to state my hearty appreciation to my advisors, Prof. Dr. Selma Ayşe ÖZEL and Prof. Dr. Yücel SAYGIN for their supervisons, patience, motivations, beneficial advices and their precious times for this thesis.

I would like to thank my PhD thesis juries, Prof. Dr. Derviş KARABOĞA, Assoc. Prof. Dr. Umut ORHAN, Assoc. Prof. Dr. Serdar YILDIRIM, and Assist. Prof. Dr. Buse Melis ÖZYILDIRIM for their comments and suggestions.

I would like to thank Assist. Prof. Dr. Ali İNAN for his good suggestions, and his precious time for this work.

Lastly, I would like to state my hearty gratitude to my dear family who have promoted me with their encouragements and love in the course of the whole process of this thesis.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES PAGE

LIST OF ALGORITHMS PAGE

LIST OF ABBREVIATIONS

ABC	: Artificial Bee Colony
ACO	: Ant Colony Optimization
BN	: Bayesian Networks
DE	: Differential Evolution
GA	: Genetic Algorithms
k-NN	: k-Nearest Neighbours
MLP	: Multilayer Perceptron
NB	: Naïve Bayes
SOM	: Self Organizing Maps
SVM	: Support Vector Machines
1R	: One Rule

1. INTRODUCTION

Data mining is a process to discover beneficial information from big quantity of data. The extracted information can be patterns, rules, clusters or a classification model. Throughout data mining process, sensitive information of individuals, subjects to several parties such as collectors, owners, users and miners are needed. Consequently, privacy preserving data mining has emerged as a significant sub-field of the data mining. Privacy preserving data mining is interested in maintaining data mining techniques without disclosing the privacy of individual data or sensitive information (Vaghashia and Ganatra, 2015).

Many privacy preserving techniques to mine sensitive data have been studied for years, which includes perturbation of output or data (Adam and Worthmann, 1989), secure multiparty computation (; Lindell and Pinkas, 2002; Goldreich, 2004), and some anonymization techniques such as *k*-anonymity, *l*-diversity, *t*-closeness (Samarati and Sweeny 1998; Samarati and Sweeny 1998; Samarati, 2001; Machanavajhala et al., 2007; Li et al., 2007, Mendes and Vilela, 2017).

Recently, differential privacy, which is a strong privacy guarantee, has been proposed to perform data mining algorithms over databases which contain sensitive information (Dwork et al., 2006). Differential privacy determines privacy leakage ratio by ϵ parameter, and enables individuals' data to be taken safely in a database (Dwork et al., 2006; Dwork, 2008; Dwork and Roth, 2014). Differential privacy asserts that the presence or absence of an individual in a database cannot change the statistics given out substantially. This formal privacy concept eliminates the side information suppositions by considering privacy in the worst case in which an attacker has all information about the records except one record in a database. Thus, independent of the auxiliary information known by attacker, it is indistinguishable by the participation of an individual in the database.

In principle, differential privacy aims at maximizing the utility of data by minimizing the leakage of sensitive data. A differential privacy mechanism such as Laplace mechanism adds random noise drawn from Laplace distribution to the outputs of the functions running over sensitive data (Mivule et al., 2012 ; Sarwate and Chaudhuri, 2013; Ji et al., 2014; Sanchez et al., 2015; Chaudhuri, 2011; Chaudhuri and Monteleoni, 2008; Rubinstein et al., 2009; Zhang et al., 2012; Ji et al., 2014; Fukuchi et al., 2017; Friedman and Schuster, 2010; Bojarski et al., 2015; Fletcher and Islam, 2015; Fletcher and Islam, 2016; Gursoy et al., 2017). If only a small amount of noise is added to the output, the resulting output is close to the actual output. On the other hand, when the noise is of large amount, the privacy guarantee is rigorous, but the data utility gets worse. Hence, the privacy-utility trade-off is a basis for the private algorithms. For instance, when a differentially private classification algorithm is considered to illustrate privacy-utility trade-off, the case is that the lower the values of ϵ parameter are, the lower classification accuracy results are but the more privacy; while the higher the values of ϵ parameter are, the higher classification accuracy results are but the less privacy.

The underlying idea of differential privacy is that the output of a query which is directed to the database is unsusceptible to whether a person is in the database. Namely, the result of a particular query Q sent to database D is indistinctive from the same query Q being dispatched to the database D' which differs from database *D* with a single record.

Differential privacy perturbs the data by adding noise (such as Laplace) to the query results. The amount of noise to be added is determined by the sensitivity of the query result statistics. The sensitivity of a function (i.e., query) grants an upper bound on how much we must perturb its value to protect privacy. For instance, we consider a simple count query, whether any person is or not in the database changes all of the query result set with only one unit (i.e., $\|D' - D\|$). Hence, the sensitivity of a count query is equal to 1 (Dwork et al., 2006; Dwork and Roth, 2014). On the other hand, the sensitivity is vital for data utility (such as good accuracy for classification) since the low sensitivity values are expected to result in better accuracies compared to the high sensitivity values on especially for small datasets including a small number of instances.

Accordingly, high level data mining algorithms such as differentially private decision trees, random trees, random forests, NB etc. have been proposed by using the queries which have low sensitivity values (such as count queries) in the literature (Friedman and Schuster, 2010; Vaidya et al., 2013; Jagannathan et al., 2012; Jagannathan et al., 2013; Patil and Signh, 2014; Rana et al., 2015; Bojarski et al. 2015; Fletcher and Islam, 2015; Fletcher and Islam, 2016; Fletcher and Islam, 2017).

Although there exist some differentially private classification algorithms, none of them are rule-based except decision trees such as ID3 which is an indirect rule-based classifier. However, among all classification algorithms, rule-based classifiers, which produce relatively small number of accurate classification rules, are preferred since the classification rules are readily confirmed, highly expressive, easy to understand and interpret by humans, and the required time to classify new instances is quite short (Duch et al., 2000). Although decision trees are well-known rule-based classifiers, they have some drawbacks such as unrelated attributes and noise in the data may cause decision trees to be unstable. For instance, a little shift in one split near to the root node will change the subtrees. Also, a tiny change in the training dataset can cause the algorithm to select a wrong attribute as the root node which affects the construction of the whole tree (Quinlan, 2014). Therefore, instead of applying decision tree learning algorithms, rule induction methods which mine data to discover the classification rules by exploiting meta-heuristic methods have been proposed in the literature. For example, Parpinelli et al. (2002) have developed Ant-Miner, which is based on Ant Colony Optimization (ACO) to extract classification rules, and showed that Ant-Miner has competitive classification performance with respect to CN2 (Clark and Niblett, 1989) which is another well-known data mining algorithm for the classification task (Parpinelli et al. 2002). At the same time, some other rule discovery algorithms based-upon Artificial Bee Colony (ABC), Differential Evolution (DE), Genetic Algorithms (GA) etc. have been developed and the success of these rule-based classifiers have been demonstrated with the experimental results in the literature (Fidelis et al., 2000; Celik et al., 2011; Shukran et al. 2011; De Falco, 2013; Talebi and Abadi, 2014, Celik et al., 2016). Although these algorithms are so successful, any privacy preserving implementation of them has not been proposed for performance evaluation in the literature to our best knowledge.

ABC (Karaboğa, 2005) is a nature inspired algorithm which imitates foraging behavior of bees. It has been proven to be a powerful algorithm to solve global optimization problems of continuous space. Also, it has some advantages such as simplicity, flexibility, and having just a few parameters. On the other hand, some approaches using ABC to discover classification rules have been proposed in the literature and the success of ABC algorithm for the discovery of classification rules has been demonstrated over some datasets that are from University of California Irvine Repository (UCI) (Celik et al, 2011; Shukran et al., 2011; Talebi and Abadi, 2014; Celik et al., 2016).

Motivated by this shortcoming in the literature, we propose to develop rule-based classifiers using ABC algorithm with the input perturbation technique of differential privacy for the first implementation in this thesis. Input perturbation technique is one of the three techniques to apply differential privacy, while the others are output perturbation and objective perturbation.

The flexibility of privacy mechanism such as Laplace facilitates a few alternating techniques to construct differentially private implementations as summarized below:

1. *Input perturbation technique*: In this technique, the data is perturbed by adding noise to the values of its numerical attributes for a certain privacy level (i.e., ϵ). Functions running over this perturbed dataset, private or non-private, will

provide the differential privacy guarantee. It is the most straightforward technique to apply differential privacy and its most significant advantage is that it enables to release the noisy dataset while maintaining the privacy. In other words, it is independent of any data mining algorithm unlike objective perturbation and output perturbation techniques, and several researchers can utilize this private data to run their own functions on it (Mivule et al., 2012; Sarwate and Chaudhuri, 2013; Ji et al., 2014; Sanchez et al., 2015; Antonova, 2015; Edlich 2017).

2. *Output perturbation technique*: In this technique, the output of an algorithm or function is perturbed for preserving privacy. For example, to create a differentially private machine learning algorithm to perform logistic regression, logistic regression is trained as non-private; then a noise vector with the same dimension is added to the original estimate, which is a differentially private logistic regression proposed by Chaudhuri et al. (2011) (Chaudhuri et al., 2011; Antonova, 2015). For another example, a count query can be used as a function, such that let the actual query result of a count query be δ, then the differentially private result (i.e., noisy result) is $\delta + b$, where b is noise which is drawn from Laplace distribution with mean 0 and standard deviation $\frac{\Delta f}{\epsilon}$, where Δf is the sensitivity of count query and equal to 1, and ϵ is the privacy level.

3. *Objective perturbation technique*: Objective perturbation, which was proposed by Chaudhuri et al. (2011) for the empirical risk minimization, adds noise to an objective function prior to optimization (Antonova, 2015; Edlich, 2017).

According to the literature, most of the differentially private classification techniques are based-on output perturbation. However, output perturbation is not suitable for the data mining algorithms which require too many appeals to the database to perform mining process (Ji et al., 2014). In these cases, input perturbation can be a solution to perform privacy preserving data mining (Mivule et al., 2012; Ji et al., 2014; Sarwate and Chaudhuri, 2013; Edlich, 2017). For instance, Mivule et al. (2012) have proposed to perturb the input data under differential privacy guarantee for classification, and find an optimal noise amount (i.e., ϵ) to achieve satisfactory classification results iteratively. It has been demonstrated that higher values of ϵ parameter provide better classification accuracies, but lower privacy for ensemble classifier; while the lower ones lead to unsatisfactory classification results with high level privacy as in other differentially private classification algorithms based-upon output perturbation technique in the literature (Mivule et al., 2012). Therefore, the input perturbation technique adopted in the studies of Mivule et al. (2012) to provide differential privacy guarantee is used in this thesis to build rule-based classifiers using ABC optimization algorithm for the privacy preserving classification. The performance of the proposed rule-based classifier using ABC is compared with eleven well-known classification algorithms such as C4.5 (Quinlan, 1993), Naïve Bayes (NB) (Murphy, 2006), Bayesian Networks (BN) (Jensen, 1996), Multilayer Perceptron (MLP) (Rumelhart et al., 1986), IBk (Aha et al., 1991), Kstar (Cleary and Trigg, 1995), One Rule (1R) (Holte, 1993), PART (Frank and Witten, 1998), Random Tree (Breiman, 2001), Bagging (Brieman, 1996), and RIPPER (Cohen, 1995) over differentially private data which is perturbed with input perturbation technique of differential privacy. Additionally, the experiments have been performed over non-private data as well to show the performance of the classifier.

In the literature, the differentially private implementations of well-known classification algorithms such as decision trees, random trees, random forests, NB, and *k*-NN have been proposed. However, any implementation of differentially private 1R algorithm which is simple and short, but efficient and accurate classifier has not been studied so far to our best knowledge. To cover this gap, we propose to develop a differentially private 1R algorithm (Holte, 1993) for the second implementation of rule-based classifiers by using meta-heuristics in this thesis. For this implementation, the output perturbation technique to ensure differential privacy is adopted to build a differentially private 1R algorithm.

In the differentially private 1R implementation, firstly ABC-DE based feature selection method proposed in the study (Zorarpacı and Özel, 2016) is applied as a pre-processing step in the data owner side before the construction of the differentially private 1R classification algorithm. After the feature selection, this pre-processed data is located in differentially private database which responds to count queries that are necessary for the classification algorithm (i.e., 1R), by adding Laplace noise to the actual results of the count queries. The usage of ABC-DE based feature selection method in the data owner side arises from the necessity that the most significant issue for differential privacy is to appeal a database as few as possible for any classification algorithm. In our proposed differentially private 1R classification algorithm, we need to access to the database for only count queries. The number of these queries is equal to $class_{number} \times \sum_{j=1}^{n} \sum_{i} 1$, where *n* represents the number of attributes in the data, *j* is the j^{th} attribute (i.e., predictor) of the data, and *i* is the i^{th} value of the attribute *j*. Accordingly, it is clear that the reduction of *n* (i.e., the number of attributes) decreases the number of count queries sent to the differentially private database. Hence, in this implementation of the thesis, we propose to apply ABC-DE based feature selection method to the data as a pre-processing step, which reduces the number of attributes on a large scale, in the data owner side, and then this pre-processed data is located in a database which uses Laplace mechanism to guarantee differential privacy to respond the count queries, and called as differentially private database. Then, 1R classifier is built with the differentially private count query results. For the performance evaluation of the proposed differentially private 1R classifier, the privacy preserving model used for 1R is applied to build differentially private NB classifier as well. Because NB is utilized as a baseline for differentially private classification in the literature and its construction process is very similar to 1R algorithm in terms of requirement of count queries for each value of each attribute (i.e., predictor) in the data to build it. Therefore, the proposed differentially private 1R algorithm is compared to the differentially private NB classifier for performance evaluation.

1.1. The Aims and Objectives of this Thesis

The superiority of rule-based classifiers which use meta-heuristic algorithms such as ACO, ABC and DE to decision trees have been demonstrated in the literature recently (Parpinelli et al., 2002; Celik et al, 2011; Shukran et al., 2011; De Falco, 2013; Talebi and Abadi, 2014; Celik et al., 2016). Although these data mining algorithms are quite successful, any implementation of these rule induction methods under differential privacy guarantee has not been studied so far. Therefore, the first aim of this thesis is to develop ABC based classification rule induction algorithms with differential privacy as ABC is a powerful optimization technique. Accordingly, we propose to perform rule induction algorithms using ABC for the differentially private and non-private data in the first implementation in this thesis.

Some classification algorithms such as ID3 (Friedman, Schuster, 2010), NB (Vaidya et al., 2015), *k*-NN (Gursoy et al., 2017) etc. have been implemented with differential privacy. However any implementation of 1R algorithm, which is short, simple, and accurate rule-based classifier, has not been proposed so far although the success of this algorithm for classification is well-known in the literature (Holte, 1993). Therefore, in this thesis the second aim is to propose a differentially private 1R algorithm and show its performance.

During the construction of the differentially private 1R algorithm, we propose to take advantage of ABC-DE based feature selection proposed in the study (Zorarpacı and Özel, 2016) to reduce the count queries sent to the differentially private database, which increases the classification accuracies significantly. On the other hand, differentially private NB (Vaidya et al., 2015) is implemented by combining with ABC-DE based feature selection as in 1R algorithm to make fair performance comparison with our proposed classifier since the very similar count queries to those of 1R algorithm are required to build NB classifier.

1.2. Contributions of this Thesis

Studies covered in this thesis aim to investigate the classification performance of some rule-based classifiers by using meta-heuristics under differential privacy guarantee. For the first implementation, rule-induction algorithms based on ABC algorithm by using input perturbation technique of differential privacy are proposed. To our best knowledge, it is the first implementation of differential privacy with the rule induction algorithms using meta-heuristics in the literature.

For the second implementation, differentially private 1R algorithm which is a short, effective, and well-known rule-based classifier is proposed. To our best knowledge it is the first implementation of 1R algorithm with differential privacy. To increase the classification accuracy of the proposed differentially private 1R algorithm, ABC-DE based feature selection proposed in the study (Zorarpacı and Özel, 2016) is applied since this pre-processing step decreases the number of count queries sent to the differentially private database on a large extend. At the same time, differentially private NB classifier (Vaidya et al., 2013) is developed by applying the ABC-DE based feature selection as pre-processing also, and it has been observed that the classification accuracies of these differentially private algorithms increase significantly thanks to the feature selection method which decreases the number of required count queries sent to the private database.

1.3. Outline of the Thesis

This thesis is organized as follows:

In Section 2, the related studies in the literature are reviewed and general information is given about them.

Section 3 covers materials for this thesis that are ABC and DE algorithms, the concept of differential privacy, 1R classification algorithm, and NB classification algorithm. In the last part of this section, the datasets used for evaluating the proposed methods are given.

In Section 4, the proposed rule-based classifiers are explained in detail.

In Section 5, the experimental results of the algorithms are presented, and they are compared with the existing methods in the literature.

Finally, the advantages and disadvantages of the proposed rule-based classifiers under differential privacy are discussed, and the perspectives about future works are presented in the last part of this thesis.

2. PREVIOUS WORKS

2.1. Classification with ABC Algorithm

Classification is the task of assigning predefined class labels to previously unseen data objects according to values of their features. To perform classification task, a set of data objects with their associated class labels are used to train a classifier, then this classifier is applied to new objects to assign class labels.

Classification has many application domains such as patients' records classification to diagnose a specific illness, image classification to identify certain objects, text classification to determine its author, sentiment classification to analyze reputation of a company, etc.

In the literature, classifiers such as SVM, MLP, etc. take advantages of ABC algorithm to optimize their parameters, or to apply preprocessing by selecting a subset of features (Gao et al., 2017; Shah et al., 2014; Palanisamy and Kanmani, 2012; Rangasamy and Duraisamy, 2018). On the other hand, just a few approaches use ABC as a classifier in the literature.

Celik et al. (2011) have developed ABCminer which uses ABC algorithm to extract classification rules from data. They have applied ABC algorithm for the discovery of classification rules over the datasets such as Breast Tissue, Breast Wisconsin, and Zoo (Celik et al., 2011). The proposed method run 5 times with 10 folds cross validation and the experimental results are compared to C4.5 decision tree classifier. It is observed that 74.92%, 93.31%, and 90.49% of average accuracy values are reached by ABCminer while 75.09%, 95.42%, and 93.07% average accuracies are achieved with C4.5 decision tree for the datasets Breast Tissue, Breast Wisconsin, and Zoo respectively.

In 2011, Shukran et al. have adopted an ABC algorithm based on enhanced local strategy to discover classification rules from data reducing enormous amount of time for the convergence of ABC. They have compared the performance of these discovered classification rules with standard data mining algorithms that are SOM, PART, NB, *k*-NN over 6 UCI datasets such as Breast Tissue, Iris, Zoo, Can, Monk, and Soybean. In this study, a simple modification is proposed to change the local strategy of ABC algorithm and they compare the improvement of the new modified ABC with the classical ABC in terms of classification accuracy. The proposed ABC classification algorithm reaches 96.3%, 94.8%, 92.5%, 96.4%, 99.9%, and 95.5% average accuracy values at the end of 10 runs of 10 folds cross-validation (Shukran et al., 2011).

Talebi and Abadi (2014) have constructed BeeMiner which is a novel ABC algorithm for rule discovery. BeeMiner uses an information-theoretic heuristic function (IHF) different from classical ABC algorithm for searching most up-and-coming areas across the search space. For rule representation, four conditions are used for each attribute to determine whether the attribute is removed from the condition list of the rule or not, and the continuous condition of the attribute are specified by considering the other three cases. They have compared the performance of BeeMiner with those of J48, JRip, and PART on nine benchmark datasets that are Breast Tissue, Vertebral Column, Ecoli, Glass, Ionosphere, Liver disorders, Parkinsons, Sonar, and Wine from the UCI Machine Learning Repository. 70.94%, 84.97%, 83.39%, 68.69%, 89.86%, 67.71%, 89.03%, 72.31%, and 95.39% average accuracy values are obtained by BeeMiner.

CoABCMiner (Celik et al., 2016) has been proposed for cooperative rule classification system which extracts all classification rules at once. In other words, ABC algorithm is run to discover all classification rules simultaneously. This method takes the data and learns the rule list. New updating strategy and token competition are employed, and new scout bee mechanism is used to discover different rules for different classes simultaneously. It has been demonstrated that CoABCMiner is employed for the discovery of classification rules from the data sets utilized in the experiments effectively.

2.2. Differentially Private Classification

Data mining is the process of discovering the useful information from the data. Privacy preserving data mining is an important research area in data mining. The goal of the privacy preserving data mining is to ensure the privacy of individuals while enabling to perform data mining techniques. Many privacy preserving techniques such as privacy preserving association rule mining, privacy preserving clustering (Vijayarani andPrabha, 2011; Preethi et al., 2018; Inan et al., 2007; Hyma et al., 2019), privacy preserving classification relying on a number of data mining algorithms such as SVM, decision trees, NB, *k*-NN etc. (Kantarcioglu and Clifton, 2004; Jalla and Girija, 2019; Hyma et al., 2018; Liu et al., 2008; Tsang et al., 2011; Liu et al., 2009) have been studied. On the other hand, heuristic methods such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), ABC, Differential Evolution (DE) etc. have been utilized in privacy preserving manner (Ravi et al., 2012; Vijarayani and Prabha, 2011; Ravi and Chitra, 2014; Mohana and Sahaaya Arul Mary, 2017; Vijayarani and Janakiram, 2016).

Vijarayani and Prabha (2011) have performed an association rule hiding method using ABC algorithm (Vijarayani and Prabha, 2011).

Ravi and Chitra (2015) have adopted a privacy preserving approach to investigate the effects of *k*-anonymization. In the study, ABC and DE algorithms are used for feature generalization and suppression to remove features without decreasing classification accuracy (Ravi and Chitra, 2014; Ravi and Chitra, 2015).

On the other hand, differential privacy has recently been proposed method to guarantee strong privacy and it has been used for privacy preserving classification. Therefore, differential privacy has been implemented with some data mining algorithms in the literature. A differentially private logistic regression algorithm has been proposed (Chaudhuri, and C. Monteleoni, 2008). Su et. al (2015) has developed a differentially private k-means clustering algorithm. Vaidya et. al performed differential privacy on Naïve Bayes classification algorithm (Su et al., 2015). Gursoy et al. has conducted a differentially private nearest neighbor classification method by using k-NN (Gursoy et al., 2017). Moreover, some indirect rule-based differentially private algorithms have been proposed in the literature (Blum et al., 2005; Jagannathan et al., 2012; Jagannathan et al., 2013;

Patil and Signh, 2014; Rana et al., 2015; Fletcher andIslam, 2016; Fletcher and Islam, 2017).

In 2010, a differentially private version of ID3 tree (Quinlan, 1993) where the information gain is estimated with the utilization of noisy counts obtained by adding noise drawn from Laplace distribution has been proposed (Friedman and Schuster, 2010). After that, Jagannthan et. al has demonstrated that construction of such a differentially private ID3 tree with the usage of low-level queries cannot ensure both of good privacy and accuracy meanwhile (Jagannathan et al., 2012). Hence, they have presented a private ensemble method attributed to random decision trees. They observe that this algorithm performs better than the differentially private ID3 tree in terms of accuracy values even for small datasets. In 2013, they have proposed a variant of the differentially private random tree ensemble in (Jagannathan et al., 2013). In this study, a semi-supervised method which modifies the random decision tree approach to use with the unlabeled data has been performed. This hybrid technique increases the accuracy values of the previous study (Jagannathan et al., 2012) without decreasing the privacy (Jagannathan et al., 2013).

Patil and Singh integrated the differential privacy idea with the random forest algorithm. Their experimental results show that the accuracy values of the non-private and differentially private random forest are approximately equal for the datasets (Patil and Signh, 2014).

Fletcher and Islam (2015) have developed a differentially private decision forest approach which employs Gini index to construct a decision tree. The proposed approach has been compared to differentially private ID3 of (Friedman and Schuster, 2010) and non-private random forest algorithms. It has been demonstrated that the proposed method has very close accuracy values to those of classical random forest algorithm (Fletcher and Islam, 2015). At the same time, Bojarski et al. (2015) have presented three variants of differentially private random decision trees with majority voting, threshold averaging, and probabilistic averaging mechanism to classify instances.

The above mentioned differentially private classifiers are based-on output perturbation techniques, but Mivule et al. (2012) have proposed to perturb the input data to find an optimal noise amount which provides differential privacy guarantee to achieve satisfactory classification results. They have demonstrated that higher values of ϵ parameter provide better classification accuracies, but lower privacy for ensemble classifier; while the lower ones lead to unsatisfactory classification results with high level privacy (Mivule et al., 2012).

According to the literature, all differentially private classification techniques are based-on output perturbation except the study in (Mivule et al., 2012) in which perturbing the input data under differential privacy guarantee for classification has been proposed. In the first implementation of this thesis, the input perturbation technique adopted in the studies of Mivule et al. (2012) to provide differential privacy guarantee (Sarwate and Chaudhuri, 2013) is used to build rule-based classifiers using ABC optimization algorithm for the privacy preserving classification. The proposed rule-based classifiers using ABC are run over non-private and differentially private data and this study is the first to discover classification rules from a differentially private data.

Additionally, the proposed ABC rule-based classification algorithms (i.e., wLapMS ABC, 1-rule ABC, and sequential covering wLap ABC) differ from ABC-based rule miners in the literature in terms of the usage of ABC operators in binary form, the structure of solutions (i.e., food source), and the dataset pruning process as we do not need to remove the instances covered by the rules from the training data thanks to the proposed rule similarity measure which is utilized in the selection phase of ABC to generate different rules from the previously generated ones. In wLapMS ABC, we do not remove the training instances from the dataset which are classified by using the previously learned rules unlike ABC-based rule discovery algorithms in the literature (Celik et al., 2011; Shukran et al., 2011;

Talebi and Abadi, 2014). This allows us to keep our search space as large as possible, and to generate better classification rules as well as it enables to discover the rules with the desired number for each class (i.e., our proposed wLapMS ABC) unlike sequential covering rule induction methods (Celik et al., 2011; Shukran et al, 2011; Talebi and Abadi, 2014). On the other hand, we also propose 1-rule ABC in which each class is represented with a single rule by weighting the precision and coverage according to the training data to perform classification task since the goal of a rule-based classifier is to classify instances with the minimum number of rules.

In the second implementation of this thesis, output perturbation technique is used to build differentially private 1R and NB classification algorithms. The 1R implementation with differential privacy is the first study in the literature. However, Vaidya et al. (2013) have proposed a differentially private NB classification algorithm. In this study, total ϵ budget is used for each count query to build NB classifier and it does not provide differential privacy guarantee. However, in our proposed NB classification algorithm, total ϵ budget is partitioned into each count query and it provides differential privacy guarantee to evaluate the performance of the classifier.

3.MATERIALS

In this section, the materials such as ABC and DE algorithms, the concept of differential privacy, ABC-DE based feature selection method, and 1R and NB classification algorithms used to develop our proposed methods are presented in detail.

3.1. Basic ABC Algorithm

Artificial Bee Colony (ABC) is a meta-heuristic algorithm introduced by Karaboğa (2005) to optimize problems of continuous spaces. It imitates the foraging behavior of bees. The algorithm consists of some important components such as food sources, employed bees, and unemployed bees (Karaboğa, 2005).

The quality of a food source is related to adjacency to the nest, its nectar concentration and convenience of extracting this nectar. Each employed bee exploits a particular food source and share information such as nectar amount, distance and direction of own food source with other foragers. An unemployed bee always sights to exploit a food source. Unemployed bees consist of scouts who look for new food sources and onlookers which wait to find a food source thanks to information shared by employed bees (Karaboga, 2005).

The information sharing among bees is performed in waggle dancing area. Following an onlooker bee appreciates the information about rich sources, she makes a decision to exploit food source which is the most lucrative. Employed bees share information according to the quality of food sources (Karaboga, 2005).

Initially, each bee pretends to be an unemployed bee. This bee does not have any information about the food sources around the nest. In this situation, two choices can be possible (Karaboga, 2005):

i. It can be a scout bee and searches for a food source according to some interior motivation or exterior clue.

ii. It can move to discover a new food source employing waggle dances information.

After finding the food source, bee will act as employed bee and profit by food source. Then this bee returns to hive for draining nectar to a food store. Following draining the nectar, three choices are possible for the bee (Karaboğa, 2005):

i. It can be an independent follower after leaving the food source.
ii. It can dance and recruit nest mates before turning back to the same food source.
iii. It can continue to exploit the food source without collecting other bees.

In ABC algorithm, the half of the bees in the swarm are employed bees and other half of the bees are onlooker bees. Each food source stands for an employed bee which exploit own food source and returns to hive to put across information about his own food source with other bees. Each onlooker bee follows the dances of the bees and chooses a food source.

According to the algorithm, a food source symbolizes a potential solution (i.e., food source position) of the problem and the nectar amount of a food source exemplifies the fitness of the solution. The steps for ABC algorithm can be expressed as follows:

Step 1. *Initialization*: *SN* food source positions are generated randomly. *SN* denotes the number of employed bees or food source positions. Each food source, $X_i,\ i \in \{1,2,\dots,SN\}$, is a vector with dimension *D* that stands for the number of parameters to solve the optimization problem. Generally, the beginning food source positions are randomly produced by Equation (3.1).

$$X_i^j = X_{min}^j + rand(0,1).\left(X_{max}^j - X_{min}^j\right) \qquad (3.1)$$

where, *j=1,2,…,D*; X_{max}^j and X_{min}^j are the top and bottom points for the j th parameter of the problem; $rand(0,1)$ is a real value drawn from uniform distribution, and between 0 and 1.

Step 2. *Nectar amount (i.e., fitness value) evaluation of the food sources:* In this step, the fitness (i.e., nectar amount) of each food source is computed.

Step 3. *Employed bee process:* In this process, each employed bee is sent to a food source and seeks a new food source enjoying further nectar amount (i.e., fitness) of his own food source among his neighborhood. For each employed bee X_i, neighbor food source position is V_i computed by Equation (3.2).

$$V_i^{jrand} = X_i^{jrand} + rand[-1,1].\left(X_i^{jrand} - X_k^{jrand}\right) \qquad (3.2)$$

where X_k is a randomly selected food source, $k \in \{1,2,\dots,SN\}$ is randomly specified and has to be different from i, *jrand* $\in \{1,2,\dots,D\}$ is a random integer number, and $rand[-1,1]$ is a random value between -1 and 1.

Step 4. *Quality (i.e., fitness value) evaluation and selection*: Following exploring the new food source, the fitness of this new food source is found. If the fitness of the new food source is greater than that of the current food source, the bee stores this new food source position (i.e., solution) and abandones the old.

Step 5. *Onlooker bee process:* Following that all employed bees complete their search processes; employed bees put across the nectar amount of the food sources with the onlooker bees. When an onlooker bee perceives a food source, it evaluates the quality information obtained by all employed bees and detects a food source X_i with the probability value p_i related to its quality. For each X_i, the probability value p_i is calculated by Equation (3.3), where $fitness_i$ is the nectar amount of the food source *i* appreciated by its employed bee. To perform this, a random value which is between 0 and 1 is generated and compared with the probability value p_i. If the probability value of a food source, which is computed by using Equation (3.3), is greater than this random value, this food source is appointed to the onlooker bee which explores new food source.

$$p_i = \frac{fitness_i}{\sum_{n=1}^{SN} fitness_n} \tag{3.3}$$

Step 6. *Memorizing the best food source:* In this step, the best food source which has the highest fitness is memorized.

Step 7. *Scout bee process:* In this process of ABC algorithm, a new food source is specified by a scout bee and it is exchanged with the abandoned food source. For this process, a counter, which is called exceed limit, is used for each bee in the swarm. If there exist a bee that its counter value exceeds maximum limit, it leaves the food source and seeks a new food source. To seek a new food source, a scout bee employs Equation (3.1).

The steps through 3 to 7 are repeated until a predetermined termination criterion is met. The best solution is the (sub)optimum solution for the problem.

In Figure 3.1, flowchart of the ABC algorithm is presented. Important properties of the ABC optimization technique are summarized as follows:

i. If the fitness of a solution increases, the probability of producing a new solution from this solution increases as well.

ii. The global search process for a solution whose counter has exceeded the limit value is terminated.

iii. A random search process to discover a new solution is applied.

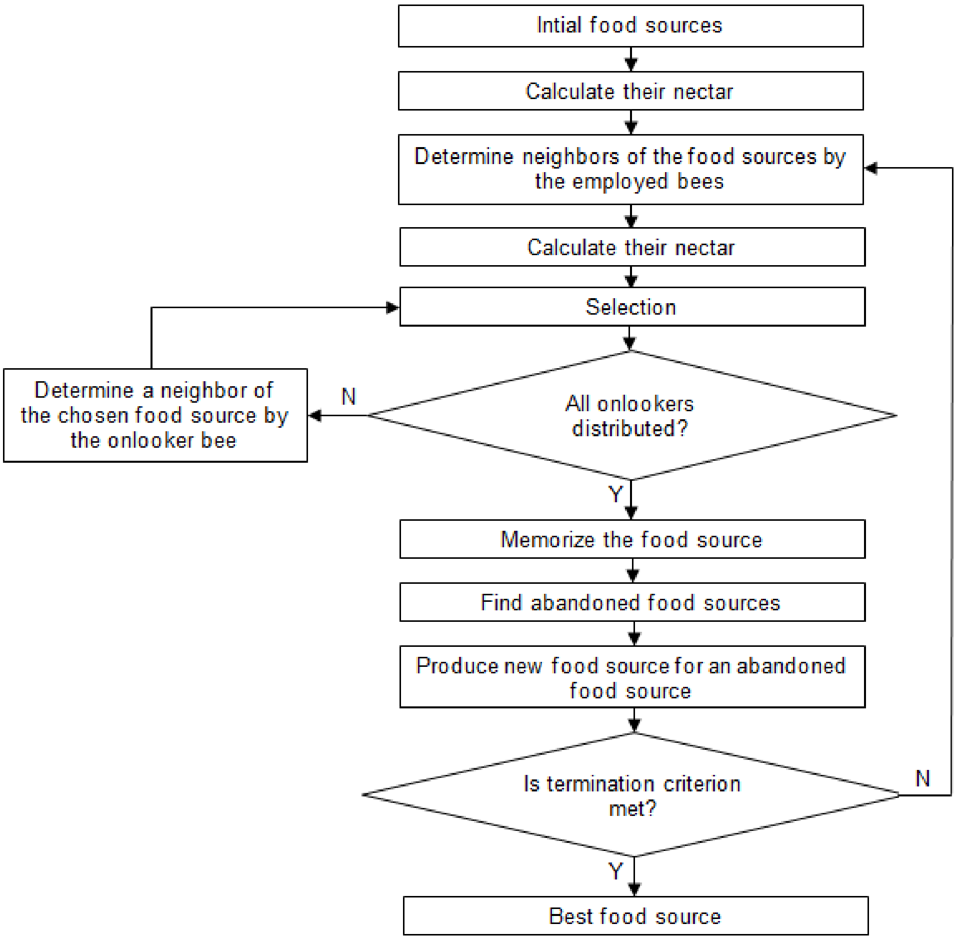

Intial food sources
Calculate their nectar
Determine neighbors of the food sources by the employed bees
Calculate their nectar
Selection
All onlookers distributed?
N
Determine a neighbor of the chosen food source by the onlooker bee
Y
Memorize the food source
Find abandoned food sources
Produce new food source for an abandoned food source
Is termination criterion met?
N
Y
Best food source

Figure 3.1. The flowchart of ABC algorithm (Zhang and Wu, 2011).

3.2. Basic DE Algorithm

DE is a straightforward real-valued evolutionary algorithm which consists of initialization of the population, difference-vector based mutation, recombination, fitness evaluation and selection steps as summarized below:

Step 1. *Initialial population:* The initial population is created by using a randomly generated real number, and a pre-described the upper and lower bounds for the parameter of problem solution, according to Equation (3.4).

$$x_i^j = x_{min}^j + rand(0,1).(x_{max}^j - x_{min}^j) \tag{3.4}$$

where x_i^j is the j^{th} parameter value of the i^{th} individual (i.e., solution) in the population, x_{min}^j and x_{max}^j are the lower and upper boundaries for the j^{th} parameter of the solution of the optimization problem respectively, $rand$ $(0,1)$ is a uniformly distributed random real value which is between 0 and 1, and x_i is the target or current individual in the population.

Step 2. *Difference-vector based mutation:* The purpose of mutation step is to produce a *donor* or *mutant* individual for a target individual x_i, in the population. Therefore, three different individuals are drawn from the population randomly, and donor vector (i.e., individual) for the target individual is computed by using equation 3.5.

$$V_i^j = x_{r_1}^j + F.(x_{r_2}^j - x_{r_3}^j) \tag{3.5}$$

where F is the real valued scaling factor in range [0..2], $x_{r_1}^j$, $x_{r_2}^j$, and $x_{r_3}^j$are the j^{th} parameter of the individuals r_1, r_2, and r_3 in the population, V_i^j is the j^{th} parameter of donor vector for i^{th} individual, and $r_1 \neq r_2 \neq r_3 \neq i$. Hence, the minimum number of individuals in the population must be 4 for the DE algorithm.

Step 3. *Recombination (Crossover):* After production of donor vector through mutation, a crossover operation is performed between donor vector and target vector to produce a *trial* vector. Equation (3.6) is used to produce a trial vector for a target vector.

$$U_i^j = \begin{cases} V_i^j, & if\ (rand[0,1] \leq Cr\ or\ j = j_{rand}) \\ x_i^j, & otherwise \end{cases} \tag{3.6}$$

where U_i^j is the j^{th} parameter of the trial vector for the i^{th} individual (i.e., target vector), *Cr* is crossover rate which provides that a particular fraction of parameters is generated by the donor vector, j_{rand} is a parameter index which is drawn from $\{1,2, \dots D\}$, and *D* is the dimension or number of parameters of the problem.

Step 4. *Fitness evaluation and selection:* In this step, fitness functions of the trial vector and target vector are compared and the higher/lower fitness value varying according to the problem at hand is preferred. The individual having better fitness function value will be in the population, while the worse one will not. Equation (3.7) expresses the selection process of DE algorithm for a minimization problem.

$$x_i = \begin{cases} U_i, & if\ \left(f(U_i) < f(x_i) \right) \\ x_i, & otherwise \end{cases} \tag{3.7}$$

According to Equation (3.7), basic DE algorithm adopts greedy selection mechanism. Namely if trial vector U_i, has better fitness than the corresponding target vector (i.e., individual) x_i, the trial vector will substitute the target vector in the next population of individuals. Otherwise, the target individual will be transferred into the next population.

After creating of the initial population, Steps 2 to 4 are repeated for each target individual in the population until a termination criterion is satisfied. At the end of the algorithm, the individual which has the best fitness is taken as the solution of the problem. The flow chart of the DE algorithm is given in Figure 3.2.

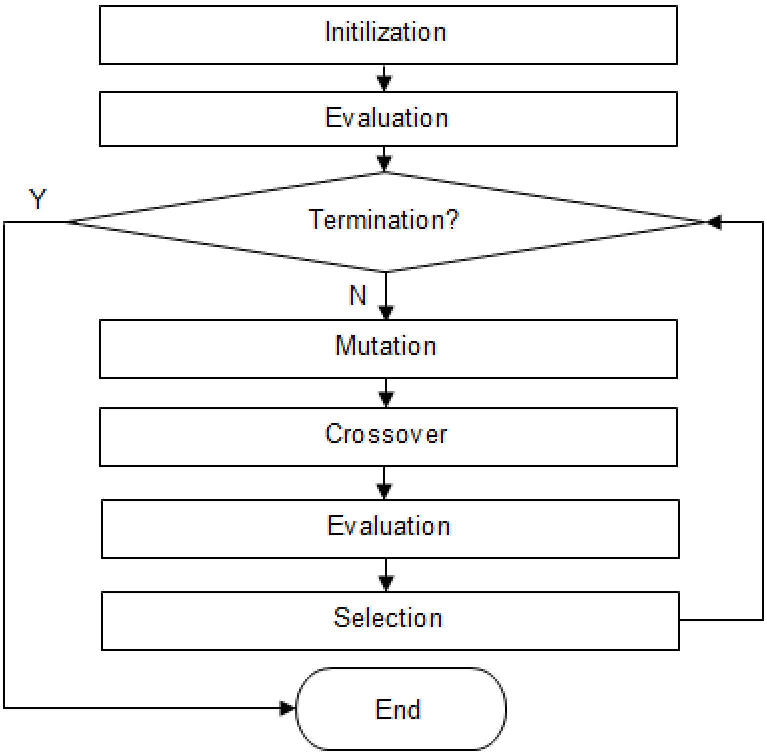


Figure 3.2. The flowchart of DE algorithm (Deng et al., 2013).

3.3. ABC-DE based Feature Selection Method

ABC-DE based feature selection method (Zorarpacı and Özel, 2016) is a wrapper method which is a hybrid approach that combines the superior properties of ABC and DE algorithms proposed in the studies of Zorarpacı and Özel (2016) to solve the feature selection problem in the classification tasks (Zorarpacı and Özel, 2016). The scheme of ABC-DE based feature selection process is presented in Figure 3.3.

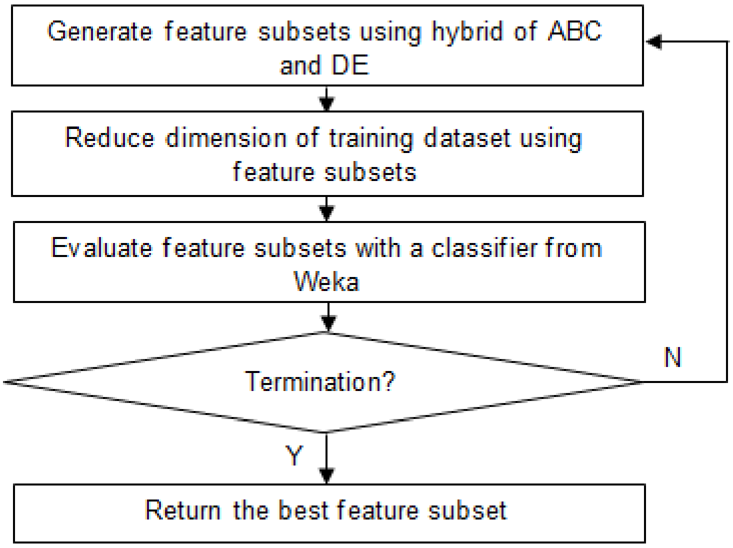

Generate feature subsets using hybrid of ABC and DE
Reduce dimension of training dataset using feature subsets
Evaluate feature subsets with a classifier from Weka
Termination?
N
Y
Return the best feature subset

Figure 3.3. The scheme of ABC-DE based feature selection process

The control parameters (i.e., scaling factor and crossover rate) affect the performance of DE algorithm. In particular, *CR* (i.e., crossover rate) is considerable to balance exploitation and exploration. Small values of *CR* promote exploitation while its large values enable exploration. Other significant control parameter *F* (i.e., mutation or scaling factor) takes real values between 0 and 2. Small values for *F* support the local search process while large values for it promote the global search process and diminish the probability to be trapped in local optimum (Mohamed et al., 2012). On the other hand, ABC algorithm shows quite well performance in terms of local search process. Therefore, ABC-DE based feature selection method combines the strong global search strategy of DE with a modified onlooker bee process of ABC algorithm.

In this method, modified mutation process of DE and modified employed bee and onlooker bee processes of ABC algorithm are used. The main steps of the ABC-DE based feature selection method are described as follows:

Step 1. *Determine the initial population of feature subsets*: To generate initial population, the feature subset vectors are created in the form of binary values. . A sample of feature subset vector for a dataset which has 8 features is depicted in Figure 3.4.

1	0	0	1	1	1	1	0

Figure 3.4. A sample of feature subset vector for ABC-DE based feature selection method

Considering the feature subset vector given in Figure 3.4, we have totally 8 features, and the chosen attribute indices are 1, 4, 5, 6, and 7, which means the classification task will be performed by using these 5 features.

Step 2. *Fitness evaluations*: Fitness evaluations for the feature subset vectors generated in the previous step are made. To compute the fitness value of a feature subset vector, the features which are indicated by 0 values in the feature subset vectors are extracted from the dataset, and the remaining ones are employed to perform classification task. This reduced dataset is classified by using 3-fold cross validation with Weka J48 (Quinlan, 1993) classifier and the classification result of J48 is equal to the fitness value for the feature subset vector. For each feature subset vectors in the population, fitness value is computed in the same way.

Step 3. *Calculating fitness probability values*: For each feature subset vector, fitness probability value is determined by using Equation (3.3).

Step 4. *DE/ABC neighborhood generation operators*: Two scenarios are considered for a source individual (i.e., feature subset vector) to perform exploitation or exploration. The scenarios are defined as follows :

i. If the probability of the source individual, X_i, is greater than a random value between 0 and 1, the mutation process of DE is applied to provide exploration. Therefore, three random individuals X_{r1} , X_{r2} and X_{r3} are chosen from the population for X_i. These individuals have to be different from each other and the X_i, which means that $r1 \neq r2 \neq r3 \neq i$. Following the determination of the individuals, the difference vector is found as in Equation (3.8). The mutant vector of X_i is specified after the computation of the difference vector. To find mutant vector, "OR" logic operator is applied over the components of X_{r3} and the difference vector, which is given in Equation (3.9). After the construction of the mutant vector, the crossover operation between X_i and the mutant vector is performed by using Equation (3.6) and the trial vector is found. In ABC-DE based feature selection method, *CR* is set to 1.

$$difference\ vector^j = \begin{cases} 0, & if(X_{r1}{}^j = X_{r2}{}^j) \\ X_{r1}{}^j, & otherwise \end{cases} \tag{3.8}$$

$$mutant^j = \begin{cases} 1, & if(difference\ vector^j = 1) \\ X_{r3}{}^j, & otherwise \end{cases} \tag{3.9}$$

ii. Otherwise, to support the exploitation, the employed bee process in ABC for X_i is applied. Therefore, two random individuals X_{r1} and X_{r2} are selected from the population $(r1 \neq r2 \neq i)$. Subsequently, a random component j_{rand} of X_i is specified and for j_{rand}, the difference between X_{r1} and X_{r2} is found, and it is called "difference component". Equation 3.10 describes how to compute the difference component. Then a random value $r1[0, 1]$ between 0 and 1, which imitates the coefficient to produce neighborhood solution of employed bee, is drawn from uniform distribution to decide whether “OR” logic operator is applied or not to the component of X_i and the difference component. Equation (3.11) describes how to build the trial vector. According to Equation (3.11), $r2[0,1]$ is a random value, j_{rand} is a random integer value in the range of [1, D] in which D is the number of attributes in the dataset.

$$difference\ component = \begin{cases} 0, & if(X_{r1}{}^{j_{rand}} = X_{r2}{}^{j_{rand}}) \\ X_{r1}{}^{j_{rand}}, & otherwise \end{cases} \quad (3.10)$$

$$trial = X_i$$

$$trial^{jrand} =$$

$$\begin{cases} 1, & if(difference\ component = 1\ and\ r1[0,1] > r2[0,1]) \\ X_i{}^{j_{rand}}, & otherwise \end{cases} \quad (3.11)$$

Step 5. *Fitness evaluation and selection process*: The fitness value for the trial vector is computed by using Weka J48 classifier with 3-fold cross validation. Following the calculation of the fitness value for the trial vector, it is compared with that of X_i. If fitness function value of the trial vector is higher than that of X_i or its fitness function value is equal to X_i, and the number of attributes in the trial vector is lower than that of X_i; the trial vector substitute X_i in the population.

Step 6. *Calculating probabilities of the individuals*: The fitness probability values of the individuals are computed by using Equation (3.3) for each X_i to perform modified onlooker bee process.

Step 7. *Modified onlooker bee process*: In modified onlooker bee process, the aim is to ensure the diversity in the population and prevent to be trapped in local optimum. Therefore, the probability value p_i of an individual X_i is subtracted from the maximum probability value, and if this value is greater than rand[0,1], "NOT" logic operator is applied to ${X_i}^{jrand}$. "NOT" operator is given in Equation (3.12). Following the determination of trial vector, fitness evaluation and selection are employed as stated in Step 5. This step is repeated *n* times, and *n* is equal to the number of onlooker bees.

$$trial{=}X_i$$
$$trial^{jrand} = \begin{cases} 0, & if\ (trial^{jrand} = 1) \\ 1, & otherwise \end{cases} \tag{3.12}$$

Steps 3, 4, 5, 6, and 7 are reiterated until a termination criterion is provided. The best solution will be the (sub)optimum feature subset for the classification task.

3.4. 1R Classification Algorithm

1R (Holte, 1993) is a simple and efficient rule-based classifier. 1R finds the most informative attribute in the data and classify instances with the values of this single attribute. As it uses only one attribute for classification task, it is called "One Rule". On the other hand, as it uses only one attribute for classification rules, it may be less accurate with respect to the state-of-art classification algorithms, however it generates very few rules that are very easy to interpret for humans (Holte,1993). The pseudo-code of 1R is given in Algorithm 3.1.

Algorithm 3.1. 1R classification algorithm

```
Input: Database $D$
Output: The IF-THEN rules of 1R classifier
Begin
    for each attribute $A_j$ in $D$ do
        for each attribute value $v_i$ in $A_j$
            Count how often $v_i$ appears in each class, and set this value to $n_{ji}$
        end for
        Detect the most frequent class of $v_i$ by using $n_{ji}$ values
        Make an IF-THEN rule with consequent as the most frequent class label and the antecedent as $A_j = v_i$
        Calculate the total classification error of the rules of $A_j$
    end for
    Choose the best attribute $A_{best}$ of which IF-THEN rules that have the smallest total error among all $A_j$
    return The IF-THEN rules of $A_{best}$;
end
```

3.5. Differential Privacy

Differential privacy (Dwork et al., 2006), which is a strong privacy guarantee, has been proposed to perform data mining algorithms over databases which contain sensitive information. It determines privacy leakage ratio by ϵ parameter, and enables individuals' data to be taken safely in a database (Dwork et al., 2006; Dwork, 2008; Dwork and Roth, 2014).

Differential privacy asserts that the output of a function does not entirely depend on any instance in the database. It claims that yielding of the same output is highly probable even if an instance is or not in the database.

Definition 3.5.1. (*Neighbor databases*) D and D' are two neighbor databases which differ from each other with a single instance, $|D'\Delta D|$=1.

Definition 3.5.2. (ϵ-*differential privacy*) A randomized mechanism A (such as Laplace mechanism) is ϵ-differentially private if all subsets S of the outputs of the algorithm A for all neighbor databases D' and D is $S \subseteq Range(A)$. And

$$\Pr[A(D) \in S] \leq e^{\epsilon} \times \Pr[A(D') \in S] \tag{3.13}$$

where $\Pr[A(D) \in S]$ is the probability of $A(D)$ of being an element of S, $A(D)$ and $A(D')$ are the outputs of the randomness algorithm A for the databases D and D' respectively, and ϵ is used to check out how much a malicious client can recognize the difference between the databases D' and D, and $Range(A)$ represents the range of the outputs which can be generated by randomized mechanism A. The smaller values of ϵ mean much more privacy.

Definition 3.5.3. (*Sensitivity*) Let $f(D) : D \rightarrow \mathbb{R}$ be a function mapping a database D into real numbers. The sensitivity for $f(D)$ is determined by

$$\Delta f := max_{D,D'} \, || \, f(D) - f(D')|| \tag{3.14}$$

where Δf := 1, $||.||$ is the L_1 norm and the sensitivity is equal to 1 for all neighbor databases D and D'. The sensitivity of a function f represents the maximum magnitude in which the record of only one individual can alter the value of f for the worst case. In other words, the sensitivity for a function grants a maximum bound on how much its output must be perturbed to provide differential privacy (Dwork et al., 2006; Dwork and Roth, 2014).

Definition 3.5.4. (Laplace *mechanism*) Let $Lap(\gamma)$ be the Laplace distribution by mean 0 and standard deviation γ. For the function $f(D) : D \rightarrow \mathbb{R}$, the randomized algorithm A, represents Laplace mechanism and responds $f(D)$ as follows:

$$A\big(f(D)\big) = f(D) + V \tag{3.15}$$

where V is an independently and identically distributed random variable drawn from $Lap(\gamma)$, and provided that $\gamma \geq \Delta f / \epsilon$, algorithm A is ϵ-differentially private. Therefore, Laplace mechanism is ϵ-differentially private (Dwork et al., 2006; Dwork et al., 2008).

$$dPr(z) \propto \exp\,(\frac{\epsilon}{2\times\Delta f}\ ||\,z||\,) \tag{3.16}$$

where dPr represents the probability density function of Laplace distribution. According to Equation (3.16), the amount of the added noise is proportional to the sensitivity and inversely proportional to ϵ. Decreasing values of ϵ expands the range of noise distribution, which results in more noisy estimates. On the other hand, increasing the sensitivity results in more perturbation as well (Antonova, 2015).

Definition 3.5.5. (*Composition property*) The goal of composition property is to provide differential privacy guarantee. Stated a total budget ϵ, the budget should be divided cleverly considering the composition properties to maximize data utility by minimizing the added noise to the actual results of the function. ϵ is a positive real number which specifies privacy level and larger values of ϵ result in less guarantees than smaller values of ϵ (Dwork et al., 2006; Dwork and Roth, 2014). Composition property consists of sequential composition and parallel composition.

The sequential composition property is used when a database is appealed to multiple times by differentially private algorithms. The parallel composition property is utilized when the disjoint subsets of data are given as input to a differentially private algorithm.

Let A_1 and A_2 be ϵ_1-differentially private and ϵ_2-differentially private algorithms. Sequential composition provides the $(\epsilon_1 + \epsilon_2)$ –differential privacy for the outputs of the functions resulted by $A_1\big(f(D)\big)$ and $A_2\big(f(D)\big)$. Parallel composition provides $\max(\epsilon_1, \epsilon_2)$–differential privacy for the outcomes of the functions resulted by $A_1\big(f(D_1)\big)$ and $A_2\big(f(D_2)\big)$.

Differential privacy provides three alternatives to construct differentially private implementations:

1. *Input perturbation*: In this technique, the data is perturbed by adding noise to the values of its numerical attributes for a certain privacy level (i.e., ϵ).

Definition 3.5.6 Let *x* be a *d*-dimensional vector of a database *D*, a differentially private variant of *x* can be given as in Equation (3.17).

$$x_{priv} = x + z \tag{3.17}$$

where *z* is a *d*-dimensional vector with a Laplace density probability function given in Equation (3.16). With this noise addition to each individual data vector x_i in the database D, it can be guaranteed that the resulting database D_{priv}= $(x_{priv_1}, x_{priv_2}, x_{priv_3}, \dots, x_{priv_n})$ is an ϵ-differentially private approximation to *D* (Sarwate and Chaudhuri, 2013).

2. *Output perturbation*: In this technique, the output of an algorithm or function is perturbed to provide privacy. For example, to create a differentially private MLE for logistic regression, logistic regression is trained as non-private; then a noise vector with the same dimension is added to the original estimate, which is a differentially private logistic regression proposed by Chaudhuri et al. (2011) (Chaudhuri et al., 2011; Antonova, 2015). As another example a count query can be given. A count query is represented as a function, if the actual query result of a count query is δ, the differentially private result (i.e., noisy result) is $\delta + b$, where b is drawn from Laplace distribution with mean 0 and standard deviation $\frac{\Delta f}{\epsilon}$, such that Δf is the sensitivity of count query and equal to 1, and ϵ is the privacy level.

3. *Objective perturbation technique*: Objective perturbation, which was proposed by Chaudhuri et al. (2011) for the empirical risk minimization, adds noise to an objective function prior to optimization (Antonova, 2015; Edlich, 2017).

In the literature differentially private classification algorithms are based-on output perturbation except the studies of Mivule et al. (2012). However, input perturbation could be utilized in case that the data is used in more than one ways, or the data analyst wants to employ a non-private algorithm to make privacy preserving data analysis (Mivule et al., 2012; Sarwate and Chaudhuri, 2013; Ji et al., 2014; Antonova, 2015) as in our proposed ABC based rule induction algorithms in this thesis. On the other hand, output perturbation cannot be suitable for the data mining algorithms which require too many appeals to the database to perform mining process (Ji et al., 2014). In such cases, input perturbation under differential privacy can be a solution to perform privacy preserving data mining as well (Mivule et al., 2012; Ji et al., 2014; Sarwate and Chaudhuri, 2013; Edlich, 2017). Therefore, input perturbation technique is used for the ABC-based rule induction methods proposed in this thesis, while output perturbation technique is applied to build a differentially private 1R algorithm as the secondly proposed method in this thesis.

3.6. Naïve Bayes Classification Algorithm

NB is a quite simple and Bayesian theory based algorithm. Although its simplicity, it is often employed as a baseline classifier (Murphy, 2006). According to Bayesian method, a new unseen instance is assigned to the most probable class label C_k from class label set *C*, given the attribute values which represent the instance to perform classification. The pseudo-code to classify a new instance by using NB classifier is given in Algorithm 3.2.

Algorithm 3.2. Naïve Bayes classification algorithm

```
Input: Database $D$, Class label set $C$,
Output: Class label $C_{NB}$
Begin
   for each  class label $C_k \in C$ do
      Compute $n_k$ which is total number of instances with class label $C_k$
         for each  attribute $A_j$ in $D$ do
            for each attribute value $v_i$ in $A_j$
                 Count how often $v_i$ appears for $C_k$, and set this value to  $n_{ji}$
                 Use $n_{ji}$ to compute $P(A_{ji}|C_k)$
            end for
          end for
      Use $n_k$ to compute $P(C_k)$
   end for
   Find $C_{NB}$= $argmax_{C_k \in C}\big(P(C_k)\prod P(A_{ji}|C_k)\big)$
   return $C_{NB;}$
end
```

According to Algorithm 3.2, A_{ji} stands for i^{th} value of attribute j, C_k is the class label with indices k, $P(A_{ji}|C_k)$ denotes the conditional probability between A_{ji} and C_k, $P(A_{ji}|C_k)=\frac{n_{ji}}{n_k}$ where n_k is total number of instances in the training dataset with class label C_k, and $P(C_k)$ is the probability of C_k, $P(C_k)=\frac{n_k}{n}$ where n is the total number of instances in the training dataset.

3.7. Datasets

In the experiments, UCI datasets are used, and their properties are given in Table 3.1. According to Table 3.1, number of classes for the datasets changes from 2 to 8, and number of attributes ranges from 5 to 61. Attribute types of the datasets are various including real, integer and categorical values.

Table 3.1. Description of Datasets

Dataset	# of Attributes	# of Classes	Attribute Type	# of Instances
Breast Wisconsin	10	2	Integer	699
Congressional Votings	16	2	Categorical	435
Credit Approval	15	2	Categorical, Real, Integer	690
Ecoli	8	8	Real	336
Heart-statlog	14	2	Categorical, Real	270
Iris	5	3	Real	150
Mushroom	22	2	Categorical	8124
Nursery	8	5	Categorical	12960
Spect-heart	23	2	Categorical	267
Sonar	61	2	Real	208

In Table 3.2, the class distribution of datasets used in our experiments are given. In Table 3.3, the datasets used for differentially private algorithms in the literature are presented as well. In the experiments, some common datasets used in the literature that are Congressional voting, Credit approval, Iris, Mushroom, and Nursery as well as different datasets that are Breast Wisconsin, Ecoli, Heart-statlog, and Sonar are employed. Different datasets from literature are used in the experiments since the differentially private classification algorithms in the literature are based-on output perturbation technique and use the datasets which have discretized or categorical attributes in general. However, for the ABC based privacy preserving classification proposed in this thesis, the input perturbation technique of differential privacy is applied and it is required to study with the datasets which have numeric attributes. Therefore, we investigate the performance of the proposed rule-based classifiers over the datasets which have different number of numeric attributes varying from 4 to 60.

For the second privacy preserving classification method proposed in this thesis the output perturbation technique of differential privacy is applied and for this purpose the datasets used in the literature that are Congressional voting, Credit approval, Mushroom, Spect-heart, Nursery, Breast wisconsin, and Heart-statlog are used in the experiments. Among them, Congressional voting, Credit approval, Mushroom, and Nursery datasets are common in the literature. On the other hand, Breast wisconsin, Heart-statlog, and Spect-heart datasets are employed in our experiments as well since medical datasets include sensitive information about the patients in general.

Table 3.2. Class distribution of the datasets

Dataset	Class Names and # of Instances for each Class							
Breast	Benign 458	Malig. 241						
Con. vot.	Dem. 267	Rep. 168						
Credit	Posit. 307	Nega. 383						
Ecoli	CP 143	IM 77	PP 52	IMU 35	OM 20	OML 5	IML 2	IMS 2
Heart	Absent 150	Present 120						
Iris	Iris-set. 50	Iris-vers. 50	Iris-vir. 50					
Mush.	Edible 4208	Poiso. 3916						
Nursery	Spec Prior 4044	Priority 4266	Very Com 328	Rec. 2	Not Rec. 4320			
Spect	No 55	Yes 211						
Sonar	Mines 111	Rocks 97						

Table 3.3. Datasets used for differentially private classification algorithms in the literature

Freidman and Schuster (2010)	Jagannathan et al. (2012)	Vaidya et al. (2013)	Bojarski et al. (2015)	Fletcher and Islam (2015)	Gursoy et al. (2017)
Adult		Adult	Adult		
					Banana
				Car	
	Cong. Vot.				
	Mushroom	Mushroom		Mushroom	
					Phoneme
	Nursery	Nursery		Nursery	
					Thyroid
					Banknote

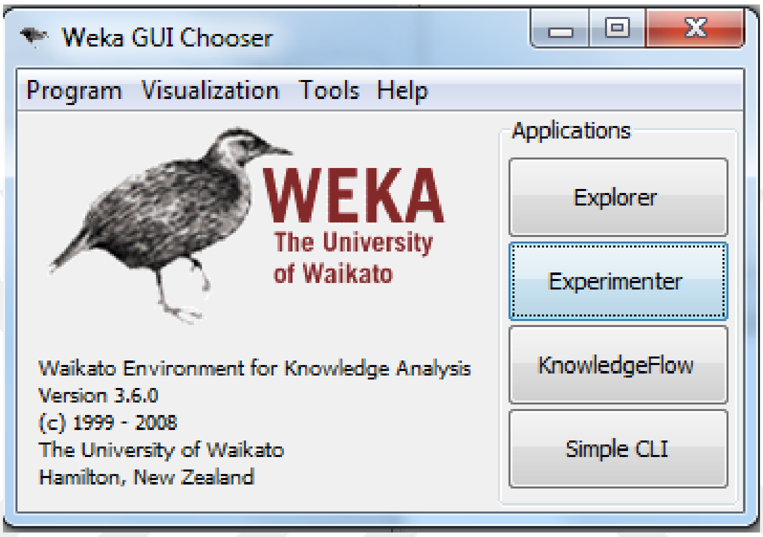
Table 3.4. Description of datasets used for differentially private classification algorithms in the literature

Dataset	**# of Attributes**	**# of Classes**	**# of Instances**	**Attribute Type**
Adult	14	2	48842	Categorical, Integer
Banana	2	2	5300	Real
Car	6	4	1728	Categorical
Con. vot.	16	2	435	Categorical
Credit	15	2	690	Categorical, Real, Integer
Iris	5	3	150	Real
Mushroom	22	2	8124	Categorical
Phoneme	5	2	5404	Real
Nursery	8	5	12960	Categorical
Thyroid	21	2	7200	Categorical, Real
Banknote	4	2	1372	Real

3.8. Weka Data Mining Tool

Weka (Waikato Environment for Knowledge Analysis) is a famous platform of machine learning software which was designed by using Java Programming Language and built in University of Waikato, New Zealand. It is free and open source software. The sample graphical user interface of Weka is shown in

Figure 3.5. Although Weka can be used through its GUI, the source codes of it can also be called in our Java code. Weka includes feature selection, data preprocessing, clustering, regression, filtering, classification, and visualization tools.

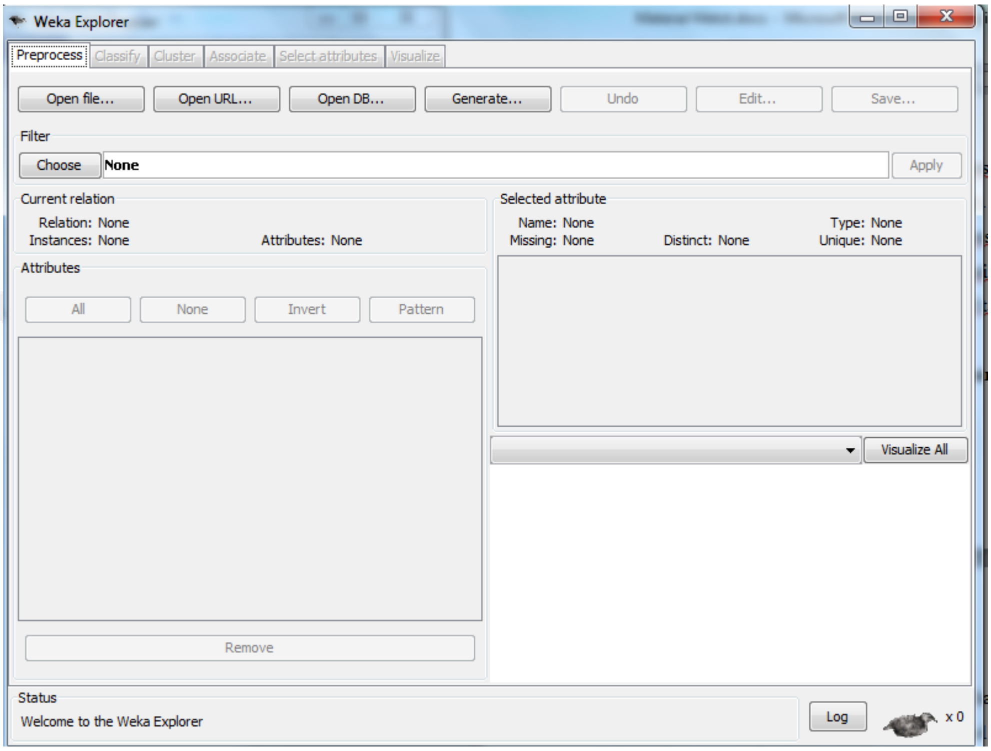


Waikato Environment for Knowledge Analysis
Version 3.6.0
(c) 1999 - 2008
The University of Waikato
Hamilton, New Zealand

Figure 3.5. Sample graphical user interface of Weka

Weka Data Mining Tool has 4 general applications that are Explorer, Experimenter, KnowledgeFlow, and SimpleCLI with several subtasks.

Explorer, Experimenter, and KnowledgeFlow have graphical user interface; while CLI has command line interface for performing data analysis. Explorer application consists of preprocessing, classification, clustering, association rule mining, attribute selection, and visualization main tasks. Preprocessing which is also called as “filters” can analyze and modify the data. Several classifiers (trees, rules, functions etc.) exist in the classification task. Clustering task includes different data clustering techniques such as SimpleKMeans etc. Association rule mining is performed by the associate task; whereas attribute selection algorithms are applied to data in the select attribute task. Finally, with visualization task, scatterplots for attribute values can be obtained. Weka Explorer Application GUI is shown in Figure 3.6.



Filter
None

Current relation
Relation: None
Instances: None
Attributes: None

Selected attribute
Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Attributes

Status
Welcome to the Weka Explorer

Figure 3.6. Weka explorer application GUI

Experimenter component provides users to apply the same techniques in the Explorer part with different parameters or apply different analysis techniques to a data. Knowledge Flow task presents to users the data sources, data sinks, filters, classifiers, clusterers, associations, evaluations, and visualization processes. CLI is used if Weka is to run in command line interface.

In this study, Weka is used for the implementation of well-known classification techniques that are J48 (Quinlan, 1993), NB (Murphy, 2006), BN (Jensen, 1996), MLP (Rumelhart et al., 1986), IBk (Aha et al., 1991), Kstar (Cleary and Trigg, 1995), 1R (Holte, 1993), PART (Frank and Witten, 1998), Random Tree (Breiman, 2001), Bagging (Brieman, 1996), RIPPER (Cohen, 1995) for the performance comparison with our proposed rule-based classifiers. To use these algorithms, the source codes of the algorithms are invoked in our Java implementations. On the other hand, to perform differentially private 1R and NB classification algorithms, the count query results used during the construction of the algorithms in the source codes of Weka Data Mining Tool are perturbed with necessary Laplace noise function.

4. METHODS

In this section, two proposed implementations of differential privacy to build rule-based classifiers by using meta-heuristics that are ABC and DE algorithms are explained in detail.

4.1. Proposed ABC-based Classifiers

To perform privacy preserving classification, three rule-based classifiers that are based on ABC algorithm are proposed in this thesis to extract IF-THEN classification rules from the input training dataset. Therefore, the proposed ABC-based classifiers in this section can be used for classification of both differentially private and non-private datasets.

4.1.1. Encoding of a Rule as Food Source Position for ABC algorithm

For the implementation, rules are formed as $IF < conditions > THEN < consequent >$ format. Here, the < *conditions* > part consists of one or more comparison-based *condition*(s) that are connected with *AND* Boolean operator. A *condition* is a predictive feature and a value pair combined with an equality operator if the predictive feature type is categorical; otherwise a *condition* consists of a predictive feature, a lower bound, and an upper bound triple connected with comparison operators which means that the value of the feature is between the lower and upper bounds if the attribute is a real or integer valued. As an example $IF\ cnd_1\ AND\ cnd_2\ AND\ cnd_3\ THEN\ cl_1$ is a rule where there are three conditions cnd_1, cnd_2, cnd_3 and a consequent cl_1 which is the class label.

We propose a two-tier structure to encode a rule (i.e., food source position) as shown in Figure 4.1. The first layer is a vector with binary values of size *n* where *n* is the number of features in the dataset, to show which features are used in the < *conditions* > part of the rule. If the value in the first layer is 1 for any feature *i*, this means the feature is used in the *conditions*, otherwise if the value is 0, the feature is not used in the *conditions* part. The second layer is a real-valued vector that represents the values of the features in the first layer. If the values of any feature (i.e., attribute) in the dataset are continuous, two-units area are allocated for this feature in the second layer to show the lower and upper bounds for the range covered by the rule. Otherwise, one unit space (i.e., 4 bytes) for each attribute is allocated to store its value.

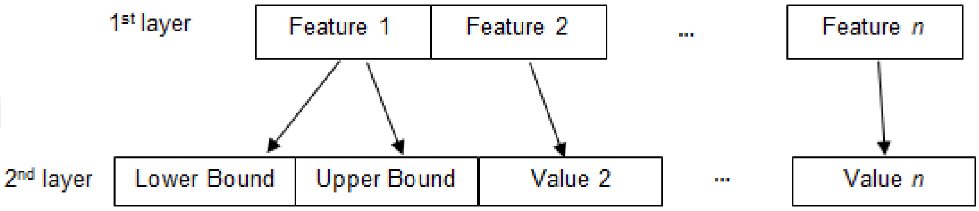


Figure 4.1. The encoding of a rule for ABC algorithm.

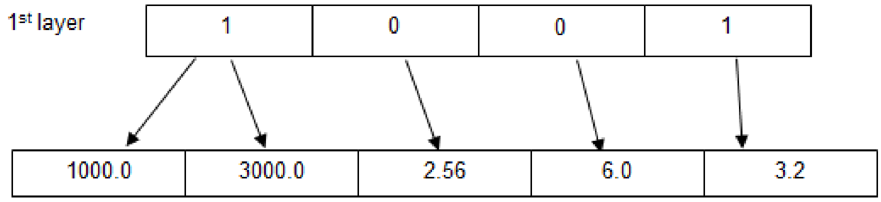


Figure 4.2. A sample encoding of the rule with 4 features.

In Figure 4.2, a sample encoding for a rule is given. It is assumed that the dataset *D* has 4 attributes excluding the class label, and the rule is for any class *A* is given. As the dataset has 4 attributes, there are 4 values in the first layer. As shown in the first layer, only the first and the fourth attributes are used in the < *conditions* > as their values are set to 1. Let the name of the first attribute be “SALARY”, and that of the fourth attribute be “AGE”. We also assume that SALARY is a real value, and AGE is a categorical attribute and takes one of the three values that are “young”, “middle” and “old”. The categorical values are encoded with real-values starting from 1.0 with 1.0 increment. As AGE has three different values 1.0 is used for “young”, 2.0 stands for “middle”, and 3.0 represents “old”. The second layer has randomly generated values for the attributes in the first layer. When we have a continuous attribute, we form a range comparison condition as for the SALARY attribute. If we have a categorical attribute, we compare the value in the second layer with the numeric encodings of the categorical values and then convert the value in the second layer to the nearest categorical value. The encoding of the food source position given in Figure 4.2 is converted to the following rule:

IF 1000.0≤SALARY≤3000.0 *AND* AGE="old" *THEN* Class *A*

As AGE has value 3.2 in the second layer, the closest value to 3.2 is 3.0 which is equal to "old". Also, the second and the third attributes are not used in the < *conditions* > part, therefore their values in the second layer are ignored.

4.1.2. Perturbation of Input Data with Differential Privacy

Differential privacy provides confidentiality guarantee for the individuals in a database while the functions are performed on the database. There exists three ways to achieve differential privacy that are input perturbation (Mivule et al., 2012; Sarwate and Chaudhuri, 2013; Ji et al., 2014; Sanchez et al., 2015), objective perturbation (Chaudhuri, 2011; Chaudhuri and Monteleoni, 2008; Rubinstein et al., 2009; Zhang et al., 2012; Ji et al., 2014; Fukuchi et al., 2017), and output perturbation (Friedman and Schuster, 2010; Bojarski et al. 2015; Fletcher and Islam, 2015; Fletcher and Islam, 2016; Gursoy et al., 2017).

According to the literature, most of the differentially private classification techniques are based on output perturbation. Although output perturbation provides strong privacy guarantee under differential privacy, output perturbation cannot be suitable for the data mining algorithms which require too many appeals to the database to perform mining process (Ji et al., 2014). On the other hand, input perturbation technique allows to release the noisy dataset while preserving the privacy, and it is independent of any data mining algorithm which is private or non-private (Antonova, 2015; Edlich, 2017).

Consequently, the input perturbation technique which provides differential privacy guarantee as in the study of Mivule et al. (2012) is adopted to perturb the training data to be mined using ABC algorithm. In this technique, perturbation process is applied to only numerical attribute values of the training data, and the algorithm of the input perturbation is given in Algorithm 4.1.

Algorithm 4.1. Input perturbation with differential privacy

```
Input: Original training data $T$, and privacy parameter $\epsilon$
Output: Differentially private training data $T_{priv}$
Begin
    for each numerical attribute $A_j$ in $T$ do
        for each numerical attribute value $v$ of $A_j$ in $T$ do
            $v \coloneqq v + Lap(\Delta f/\epsilon)$
        end for
    end for
    $T_{priv} \coloneqq T$
    return $T_{priv}$;
end
```

In Algorithm 4.1, Δf which is the sensitivity of the j^{th} numerical attribute in T is computed according to Equation (4.1).

$$\Delta f \coloneqq \, ||\, A_j(v_{max}) - A_j(v_{min})|| \tag{4.1}$$

where A_j is the jth numerical attribute of the training data T, $A_j(v_{max})$ and $A_j(v_{min})$ are the maximum and the minimum values of the jth numerical attribute in the training data, respectively.

4.1.3. Rule Similarity Measure

To discover classification rules using ABC algorithm, three methods are proposed, and they are called wLapMS ABC, sequential covering wLap ABC, and 1-rule ABC.

In wLapMS ABC, *Michigan* approach, where each food source position/chromosome (i.e., solution) represents only a single rule for a class, is adopted. This method provides a simpler structure and a shorter time to converge to the best solution (i.e., rule) for ABC algorithm (Celik et al., 2011; Shukran et al., 2011; Talebi and Abadi, 2014). However, after the extraction of the first rule for a class, it is possible that ABC algorithm converges to the same (sub)optimal classification rule for this class, that is, almost the same rules are generated for the class since we do not remove the instances covered by the discovered rules from the training data in wLapMS ABC method. To prevent from having the same or very similar rules for a class, a rule similarity measure is proposed in this thesis to embed in the greedy selection stages of ABC algorithm. Thanks to this approach, the greater similarity a candidate rule (i.e., solution) has to the previously learned rules, the lower chance it has to be selected during the selection phase of ABC algorithm. Therefore, different rules from each other are discovered from the training data for the prediction of any class. Rule similarity measure can be considered as an alternative to classical sequential covering rule induction which removes instances covered by discovered rules, and it provides to extract required number of different rules (this value is determined by the user) for each class.

To compute the similarity between a candidate rule r and all previously generated rules for the class, we propose Algorithm 4.2 which checks the intersection of the attributes used in the rule *r* and with all of the previously generated rules. Let r_j be any previously generated rule. For each common attribute in rules *r* and r_j, if the common attributes have the same value or if their ranges intersect, then similarity score of *r* is incremented by one. This computation is repeated for all previously generated rules, and cumulative similarity score for the candidate rule *r* is computed. If the resulting similarity score is high for the rule *r*, this means the candidate rule *r* covers or is covered by the previously generated rules, and it should not be selected.

Algorithm 4.2. Rule similarity measure

```
Input: Candidate rule $r$, and set of previously generated rules $RS$
Output: $SimScore_r$ which is similarity score of $r$
Begin
    $SimScore_r$ := 0
    $FS_1$ := set of valid attributes in rule $r$
    for each rule $r_j$ in $RS$ do
        $FS_2$ := set of valid attributes in rule $r_j$
        $FS := FS_1 \cap FS_2$
        for each attribute $A_i \in FS$ do
            if $A_i$ is a categorical attribute then
                if $value(r.A_i) == value(r_j.A_i)$ then $SimScore_r$ ++;
            else if $A_i$ is a continuous attribute then
                if $range(r.A_i) \cap range(r_j.A_i) \neq \emptyset$ then $SimScore_r$ ++;
        end for
    end for
    return $SimScore_r$;
end
```

4.1.4. Selection Mechanism for ABC Algorithm with Rule Similarity Measure

The proposed rule induction algorithms based on ABC algorithm search for different rules from the previously discovered best rules for the class at each iteration. In other words, different combinations of the attributes which most cover the instances of the class at hand are searched at each iteration in wLapMS ABC. Therefore, the rule similarity measure given in Algorithm 4.2 is proposed and embedded into the selection phase providing to choose the different rules from previously learned ones for the class. The selection mechanism formula is given in Equation (4.2).

$$X_i = \begin{cases} V_i, & if\ (RQM_{X_i} \leq RQM_{V_i}\ \textbf{and}\ SimScore_{V_i} \leq SimScore_{X_i}) \\ X_i, & otherwise \end{cases} \tag{4.2}$$

where X_i is the current rule in the swarm, V_i is the candidate rule, $RQM_{X_i}$and $RQM_{V_i}$are the rule quality measure values for the current rule and candidate rule respectively, and $SimScore_{X_i}$ and $SimScore_{V_i}$ are the rule similarity score values for X_i and V_i according to the previously extracted rules in the rule set of the class at hand.

4.1.5. Binary Operator of ABC for the First Layer of the Rule Encoding

The original ABC algorithm has been proposed for continuous valued optimization problems and its operators that are used to determine the new food source position in employed bee and onlooker bee phases given in Equation 3.2 are based on real-valued processes. Therefore, we use a variant of the binary operator proposed in (Zorarpacı and Özel, 2016), which has been developed to perform feature selection, for the first layer of two-tier solution structure of our ABC algorithm. This binary operator for ABC algorithm is employed to specify which conditions (i.e., attribute and value pairs) exist in a rule (i.e., food source position). The mathematical formulation of the binary operator is given in Equation (4.3).

$$V_i^{jrand} = \begin{cases} 0, & if\ \left(X_i^{jrand} == X_k^{jrand}\ \boldsymbol{and}\ F > rand(0,1)\right) \\ X_k^{jrand} & if\left(X_i^{jrand} \neq X_k^{jrand} \boldsymbol{and}\ F > rand(0,1)\right) \\ X_i^{jrand} & otherwise \end{cases} \tag{4.3}$$

where F is a real number between 0 and 1, and it controls whether $X_i^{jrand}$is changed with the difference between X_k^{jrand} and X_i^{jrand} or not. F is used for simulating the coefficient (i.e., amount of movement of $X_i^{jrand}$to X_k^{jrand}), which is rand[-1, 1] and multiplied with the difference of X_k^{jrand} and X_i^{jrand}in Equation (3.2) and for providing the convergence simultaneously for the first and second layer of our proposed ABC algorithm. Finally, $rand(0,1)$ is a random value between 0 and 1. X_i^{jrand}, X_k^{jrand}, and V_i^{jrand} are as given in Equation (3.2).

4.1.6. The Proposed Rule-based classifiers Using ABC Algorithm over Differentially Private and Non-private Data

According to the literature, ABC algorithm is used for classification only a few different ways. For the most of these algorithms, ABC is used to optimize the vital parameters of the well-known classifiers. On the other hand, a few studies use ABC to discover classification rules in the form of IF-THEN from training data (Celik et al., 2011; Shukran et al. 2011; Talebi and Abadi, 2014; Celik et al., 2016). However, none of these studies learn the rule set from a differentially private data. Therefore, to our best knowledge, this is the first study which applies ABC to learn classification rules from the differentially private data.

In our proposed method, firstly, numerical attribute values of the training data are perturbed under differential privacy guarantee by using Algorithm 4.1 to protect the sensitive data, then the classification rules are learned from this differentially private data by using the proposed rule-based classifiers employing the ABC algorithm. The flowchart of the proposed ABC algorithm to discover classification rules from training data for the both version of non-private and differentially private data is given in Figure 4.3.

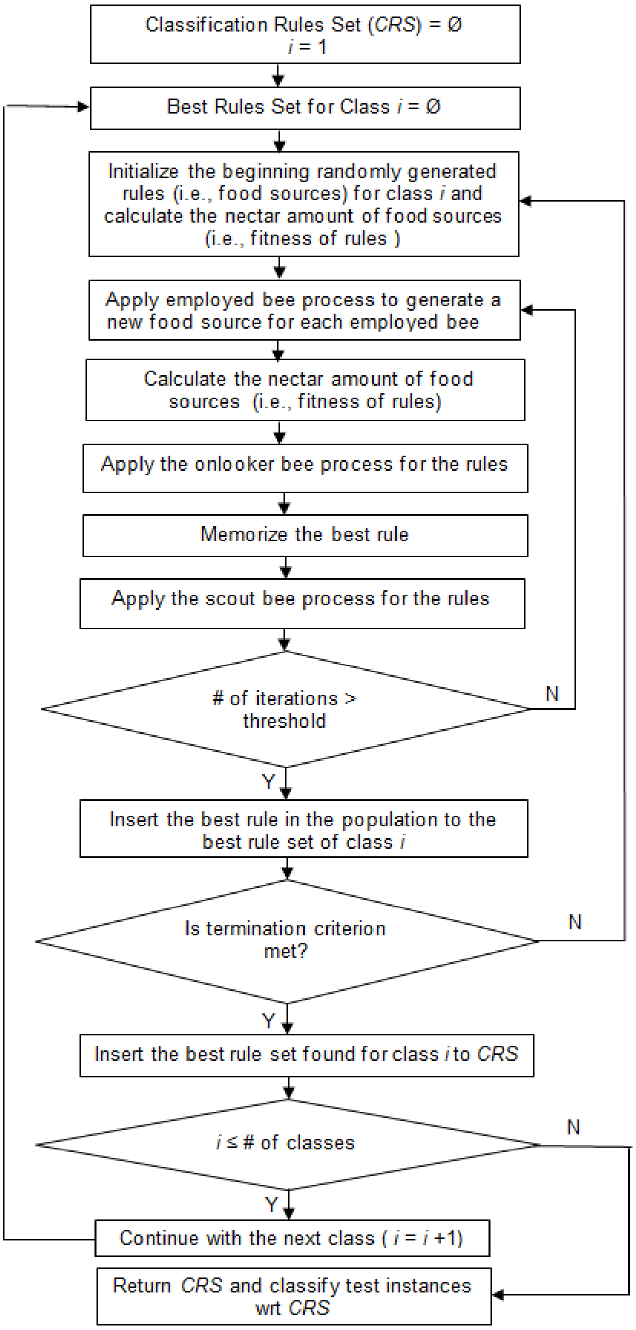

Classification Rules Set (*CRS*) = Ø
i = 1
Best Rules Set for Class *i* = Ø
Initialize the beginning randomly generated rules (i.e., food sources) for class *i* and calculate the nectar amount of food sources (i.e., fitness of rules)
Apply employed bee process to generate a new food source for each employed bee
Calculate the nectar amount of food sources (i.e., fitness of rules)
Apply the onlooker bee process for the rules
Memorize the best rule
Apply the scout bee process for the rules
# of iterations > threshold
N
Y
Insert the best rule in the population to the best rule set of class *i*
Is termination criterion met?
N
Y
Insert the best rule set found for class *i* to *CRS*
i ≤ # of classes
N
Y
Continue with the next class (*i* = *i* +1)
Return *CRS* and classify test instances wrt *CRS*

Figure 4.3. The flowchart of the proposed ABC-based classification algorithm.

In the ABC-based classifier implementation of this thesis, three rule-induction methods based-on ABC algorithm are proposed, which are called as wLapMS ABC, 1-rule ABC, and sequential covering wLap ABC, to discover classification rules over the differentially private and non-private data.

According to Figure 4.3, the proposed rule-induction methods search for a set of classification rules (i.e., IF-THEN rules) for each class in the dataset, and each classification rule for a class is learned in one by one. The best rule discovered at the end of each iteration of ABC algorithm for a class (i.e., when ABC algorithm reaches the threshold value of iterations) is stored as a classification rule for the class at hand. Searching process starts with the first class in the dataset and continues to discover rules for all classes in the training dataset.

To stop the rule discovery process for a class, the termination criteria for wLapMS ABC and 1-rule ABC is chosen as the number of rules to be discovered for the class. When this threshold is reached, rule discovery process is stopped for the class at hand, and the algorithm starts to discover new rules for the next class. This threshold value is determined by the user. For the 1-rule ABC, number of rules to be discovered for each class is set to 1. For the wLapMS ABC on the other hand, the user of the algorithm determines the number of rules to be discovered for each class and this value is greater than 1. In sequential covering wLap ABC, the termination criteria is defined as the number of uncovered instances of the current class in the training data and this value is less than or equal to 5% of all instances of the class in the training data as in other ABC-based classification algorithms (Celik et al., 2011; Shukran et al., 2011; Talebi and Abadi, 2014). The other details of our proposed methods are explained in the below subsections of this thesis.

4.1.6.1. Rule Quality Measures

Developing a rule-based classifier requires a measure to determine whether the quality of the generated rule is good or not to compute nectar amount of the food sources. Measures used for computing rule quality in terms of precision and coverage have been compared in detail in (Hilderman and Hamilton, 1999; Lavrac et al., 1999; Azevedo and Jorge, 2007; Michalak et al., 2015).

Two measures are used to compute rule qualities for ABC algorithm. First, weighted Laplace measure (wLap) is used to discover rules which have high precision and low coverage (Michalak et al., 2015). It is based on decision tree rule induction algorithm and it uses Laplace estimate (*S*+*N*)/*S* which takes into account the distribution of data instances between the classes and has the best classification accuracy even for unbalanced datasets (Michalak et al., 2015). The weighted Laplace measure wLap of a rule for class *A* is computed as in Equation (4.4).

$$wLap = \frac{(p+1)(S+N)}{(p+n+2)S} \tag{4.4}$$

In Equation (4.4), if we assume that *D* is a dataset which has instances from 3 classes *A*, *B*, and *C*; and *p* expresses the number of samples of class *A* covered by the rule, *n* is the number of instances from other classes (i.e., *B* and *C*) covered by that rule, and *S* denotes the number of all samples of class *A*. *N* is the number of instances which belong to classes (i.e., *B* and *C*) except class *A*.

Another rule quality measure is Mutual Support (*MS*) (Michalak et al., 2015) which is computed as in Equation (4.5). *MS* prefers rules with high coverage.

$$MS = \frac{p}{S+n} \tag{4.5}$$

By Equation (4.6), we propose to weight *MS* with the α and β coefficients to adjust the precision and coverage of the rule. Therefore, the user can prefer the rules which have higher precision or coverage according to the data at his hand.

$$wMS = \frac{p}{\alpha\, S+\beta n} \tag{4.6}$$

The rule quality measure wLap, prefers the rules which have high precision. However, in some cases, we need to discover rules which have high coverage as well. Therefore, to adjust coverage and precision for the rules to be discovered, we propose to use the combination of wLap and *wMS* as the rule quality measure, which is called wLapMS, and its formula is given in Equation (4.7).

$$wLapMS = \frac{(p+1)(S+N)}{(p+n+2)S} \times \frac{p}{\alpha\, S+\beta n} \tag{4.7}$$

where α and β are real numbers in the range 0 and 1and they are determined by the user experimentally.

4.1.6.2. The Proposed Sequential Covering Rule-based Classifier Using ABC with wLap Rule Quality Measure (Sequential Covering wLap ABC)

There exists two main ways to learn classification rules by using metaheuristic algorithms: i) *Pittsburg* approach in which each candidate solution (i.e., food source) stands for an entire set of rules, and each solution evolves along the iterations as an exact rule base (Tan et al., 2012) such as in (De Falco et al., 2013; Li et al., 2013) or ii) *Michigan* approach where each solution represents a single rule (Chui and Hsu, 2005) such as the encoding of the rules in (Su et al., 2010; Celik et al., 2011; Shukran et al. 2011; Talebi and Abadi, 2014).

In *Michigan* approach, initially a training set *T* is taken and discovered rule set *R* is set to empty. At each iteration, the algorithm discovers a classification rule one by one. At the end of each iteration, the best discovered rule is inserted into set *R* and all samples covered by it (i.e., samples satisfying the rule conditions) are removed from *T*. This process is repeated while the number of uncovered samples left in *T* reduces. This method is known as sequential covering rule induction and in our proposed sequential covering wLap ABC, this approach is adopted as in other rule-based classifiers that are based on ABC in the literature (Celik et al., 2011; Shukran et al., 2011; Talebi and Abadi, 2014) however we use *weighted* Laplace as rule quality measure in our implementation since measures used for computing rule quality have been compared in detail in (Hilderman and Hamilton, 1999; Lavrac et al., 1999; Azevedo and Jorge, 2007; Michalak et al., 2015) and *weighted* Laplace achieves the best classification accuracy even for unbalanced datasets (Michalak et al., 2015).

4.1.6.3. The Proposed Rule-based Classifier Using ABC with wLap and *wMS* Rule Quality Measure (wLapMS ABC)

In our proposed wLapMS ABC, classification rules are learned one by one for each class as in the *Michigan* approach, however, we do not remove instances from *T* that are covered by the learned rules from the dataset. Instead, a similarity measure given in Section 4.1.3 is used for rules learned, and this measure is employed in the onlooker and employed bee processes of ABC algorithm to learn different rules from the previously learned ones. As we do not remove any instances from the training set, wLapMS ABC is able to learn more than one rule for each class, and the number of rules to be learned for each class is determined by the user.

Additionally, in wLapMS ABC the proposed rule quality measure wLapMS, which enables that the user can adjust the weights of precision and coverage, is used.

4.1.6.4. The Proposed Rule-based Classifier Using ABC with wMS Rule Quality Measure (1-rule ABC)

In our proposed 1-rule ABC, each class is represented with a single rule and ABC algorithm is run to learn only one classification rule for each class unlike the proposed wLapMS ABC and sequential covering wLap ABC in which ABC algorithm is run to discover multiple rules for each class in the training data.

The aim of 1-rule ABC is to classify instances with the minimum number of rules since having small number of rules is always preferred. Thus, for 1-rule ABC algorithm, the total number of rules to be discovered is equal to the number of classes in the dataset.

4.2. Differentially Private 1R Classification Algorithm Using ABC and DE

Some differentially private implementations of the well-known classification algorithms such as decision trees, random trees, random forests, Naïve Bayes etc. have been proposed in the literature (Vaidya et al., 2013; Friedman and Schuster, 2010; Bojarski et al. 2015; Fletcher and Islam, 2015; Fletcher and Islam, 2016). However, to our knowledge any differentially private implementation of 1R algorithm (Holte, 1993) has not been proposed in the literature so far. As 1R is a simple, but efficient and accurate classifier, a differentially private 1R classification algorithm, which employs ABC and DE meta-heuristics to reduce feature space, is proposed in this thesis. At the same time, our proposed ABC and DE based feature selection process is also applied to build a differentially private NB classifier which is used as baseline for differentially private classification in the literature. The scheme to construct the proposed differentially private classification algorithms (i.e., 1R and Naïve Bayes) is given in Figure 4.4.

In our proposed method to build a differentially private classifier (i.e., 1R and NB), the data owner applies ABC-DE based feature selection, proposed by (Zorarpacı and Özel, 2016) as the first step. The used feature selection method is explained in Section 3.3. After the feature selection process, this reduced data is located in a differentially private database which responds the count queries, that are necessary for the classification algorithms (i.e., 1R and NB). Therefore we apply output perturbation by adding Laplace noise to the actual results of the count queries sent to the private database. The reason of the usage of ABC-DE based feature selection method in the data owner side is to appeal the private database as few as possible to reduce noise added to the results of the count queries send by the classifiers. If the number of features of the dataset is reduced, the number of queries to be sent to the database also decreases.

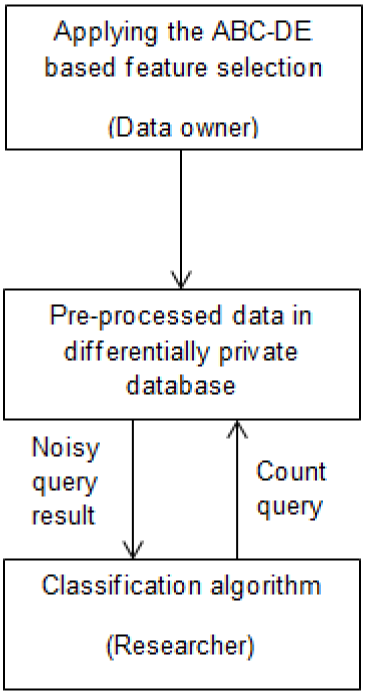

Applying the ABC-DE based feature selection
(Data owner)
Pre-processed data in differentially private database
Noisy query result
Count query
Classification algorithm
(Researcher)

Figure 4.4. The scheme of the proposed differentially private classification algorithms.

In our proposed differentially private classification algorithm, we need to access to the database for only count queries. The number of these queries is equal to $class_{number} \times \sum_{j=1}^{n} \sum_{i} 1$, here n represents the number of attributes in the dataset, j is the j^{th} attribute (i.e., predictor) of the dataset, and i is the i^{th} value of the attribute j. Accordingly, it is clear that the reduction of n (i.e., the number of attributes) decreases the number of count queries sent to the differentially private database. Hence, in this study, we propose to apply ABC-DE based feature selection method as a pre-processing step, which reduces the number of attributes on a large scale, in the data owner side, and this pre-processed data is located in a database which uses Laplace mechanism to guarantee differential privacy to respond the count queries, and called differentially private database. Then, 1R and NB classifiers are built using sequential composition property of differential privacy. The classification algorithms used are described in Algorithm 4.3 and 4.4 respectively. Sequential composition property is used since we keep our data as a whole in one database. Indeed, it is the worst-case scenario for the classifiers as it results in the minimum classification accuracy. The classification accuracies of the proposed differentially private classifier can be improved by applying data partitioning and parallel composition property of differential privacy as in Definition 3.5.5 given in Section 3.5, however, we don’t apply these techniques in this thesis study since we make a general assessment for the proposed method, these improvements may be done as a future work. Therefore, we consider the worst case scenario and sequential composition property for our proposed method.

Algorithm 4.3. Differentially private 1R classification algorithm

```
Input: Privacy parameter $\epsilon$, differentially private database $D$
Output: IF-THEN classification rules of differentially private 1R classifier
Begin
    $\epsilon' := \frac{\epsilon}{class_{number} \times \sum_{j=1}^{n} \sum_{i} 1}$
    $\gamma := \Delta f / \epsilon'$
    for each attribute $A_j$ in $D$ do
        for each attribute value $v_i$ in $A_j$
            Count how often appears in each class and set this value to $n_{ji}$
            Perturb $n_{ji}$ as $n'_{ji}$=$n_{ji}$+ $Lap(0, \gamma)$
        end for
        Detect the most frequent class of $v_i$ by using $n'_{ji}$ values
        Make an IF-THEN rule assigning the most frequent class to $v_i$
        Calculate the total classification error of the rules of $A_j$
    end for
    Choose the best attribute $A_{best}$ of which IF-THEN rules that have the smallest
total error among all $A_j$
    return the IF-THEN rules of $A_{best;}$
end
```

According to Algorithms 4.3 and 4.4, the noisy count query results are used to build differentially private 1R and NB classification algorithms unlike classical 1R and NB classification algorithms.

In Algorithms 4.3 and 4.4, the sensitivity Δf, is equal to 1 since the type of the queries sent to the differentially private database D is count query. ϵ is the total budget to guarantee differential privacy. ϵ' is the budget per each count query and is equal to $\frac{\epsilon}{class_{number} \times \sum_{j=1}^{n} \sum_{i} 1}$ due to sequential composition property of differential privacy on ϵ parameter given in Definition 3.5.5 in Section 3.5.

$Lap(0,\gamma)$ represents the noise drawn from Laplace distribution with mean 0 and standard deviation γ where $\gamma=\frac{\Delta f}{\epsilon'}$.

Algorithm 4.4. Differentially private NB classification algorithm

```
Input: Database $D$, Class label set $C$, the privacy parameter $\epsilon$
Output: Class label $C_{NB}$
Begin
    $\gamma := \Delta f / \epsilon'$;
    for each class label $C_k \in C$ do
        Compute $n_k$ which is total number of instances with class label $C_k$, $n'_k=n_k+ Lap(0,\gamma)$
        for each attribute $A_j$ in $D$ do
            for each attribute value $v_i$ in $A_j$
                Compute $n_{ji}$, $n'_{ji}=n_{ji}+ Lap(0,\gamma)$, which is equal to how often $v_i$ appears for $C_k$
                Use $n'_{ji}$ to compute $P(A_{ji}|C_k)$
            end for
        end for
        Use $n'_k$ to compute $P(C_k)$
    end for
    Find $C_{NB}= argmax_{C_k \in C}\left(P(C_k)\prod P(A_{ji}|C_k)\right)$
    return $C_{NB}$;
end
```

5. RESULTS AND DISCUSSIONS

In this section, the experimental results are given and the performances of the proposed methods and other techniques in the literature are compared.

We implemented the proposed methods in Java programming language under NetbeansIDE 8.0.2 platform. The experiments were run on a PC which has Windows 7 Home Premium operating system, 4 GB of RAM, Intel Core i5-2430M 2.4 GHz processor. Weka data mining tool is utilized to perform the other classification methods such as J48, MLP, NB etc.

5.1. Performance Metrics for Evaluation of Classifiers

In this study, the accuracy value is utilized to compare the algorithms since this metric is used as performance metric for most of the studies in the literature. The accuracy value for a classifier is computed by using the equation given in Equation (5.1).

$$Accuracy = \frac{n_{correct}}{|D|} \tag{5.1}$$

where $|D|$ is the total number of instances in the dataset, $n_{correct}$ is the number of instances correctly classified by the classifier.

5.2. Experimental Results for the ABC-based classifiers

In the experiments, we investigate the performance of the proposed rule-based classifiers using ABC in two cases which are for non-private and differentially private data. Eleven well-known classification algorithms that are J48, NB, BN, MLP, IBk, Kstar, 1R, PART, RTree, Bagging, and RIPPER from the Weka data mining tool are used to make comparison with the our proposed methods. Among the classifiers C4.5(Quinlan, 2014), PART (Frank and Witten, 1998), and RTree (Breiman, 2001) are decision tree algorithms whereas PART is a partial decision tree algorithm instead of a whole decision tree.

RIPPER is a sequential covering rule induction algorithm which learns one rule at a time and remove the instances covered by this rule from the dataset as in other ABC-based classification algorithms in the literature (Celik et al., 2011; Shukran et al. 2011; Talebi and Abadi, 2014).

NB (Murphy, 2006) is a statistical method based on Bayes theorem. A Bayesian Network (Jensen, 1996) utilizes graph-based structure to represent the probabilistic relationships between the predictors (i.e., features) of the data and deduces probabilistic results by using these relations. MLP (Rumelhart et al., 1986) is a neural network based classification method.

IBk (Aha et al., 1991) and Kstar (Cleary and Trigg, 1995) are the implementations of *k* nearest neighbor algorithm where only the *k* value used is different. 1R (Holte, 1993) is a rule-based classification algorithm that acquires one rule for each attribute (i.e., predictor), and the rule which has the minimum total error is selected as “One Rule”.

Bagging (Breimann, 1996) is an ensemble method. An ensemble method consists of a series of *n* trained base classifiers. Ensemble has better classification results than those of its base classifiers.

In the experiments, REPTree (Reduces Error Pruning Tree) (Quinlan, 2014) is selected as base classifiers of Bagging algorithm. REPTree is a fast decision tree model and based on C4.5 algorithm because it takes advantages of information gain and entropy. To apply these well-known classification techniques, the default parameter values of the algorithms in Weka have been used.

10-fold cross validation is utilized to test the performance of the classifiers. In 10-fold cross validation, the whole dataset is split into almost equal ten partitions. One partition is taken as the test data and the rest of the partitions are used as training data for each run of the algorithms. In the experiments, we apply 10 times of 10 fold cross validation to the datasets over both of non-private and differentially private data. Average values at the end of 100 runs are presented for the proposed rule-based classifiers using ABC algorithm and the well-known classification techniques.

5.2.1. Performance Evaluation of Rule-based classifiers Using ABC Algorithm over Non-private Data

In this section, the experimental results of the proposed rule-based classifiers using ABC algorithm (i.e., 1-rule ABC, wLapMS ABC, and sequential covering wLap ABC) over non-private datasets are presented. The best results are given in bold face and the rankings of the algorithms are specified in brackets in the below of the average results.

In Table 5.1, the average classification accuracies of 10 runs of 10 fold cross validation with standard deviations are given for the datasets Breast-w, Ecoli, Heart-statlog, Iris, and Sonar. In Table 5.2, average number of rules to achieve the average classification accuracies in Table 5.1 are presented.

Table 5.1. Average classification accuracies for classifiers over non-private data

Method	**Dataset**				
	Breast-w	Ecoli	Heart	Iris	Sonar
J48	0.932±0.02 (7)	0.828±0.04 (4)	0.778±0.06 (8)	0.952±0.05 (5)	0.722±0.05 (9)
NB	0.949±0.02 (2)	**0.859**±0.03 (1)	**0.849**±0.05 (1)	0.954±0.05 (4)	0.676±0.06 (11)
Bayes Net.	0.947±0.02 (3)	0.813±0.03 (7)	0.840±0.06 (3)	0.933±0.06 (11)	0.756±0.05 (6)
Part	0.923±0.02 (10)	0.835±0.04 (3)	0.783±0.07 (7)	0.945±0.06 (7)	0.742±0.05 (7)
Ripper	0.926±0.02 (8)	0.817±0.03 (6)	0.791±0.06 (5)	0.932±0.06 (12)	0.735±0.04 (8)
1R	0.921±0.02 (11)	0.653±0.05 (12)	0.708±0.07 (13)	0.938±0.05 (10)	0.618±0.05 (13)
Kstar	0.932±0.02 (7)	0.817±0.03 (6)	0.771±0.07 (9)	0.945±0.05 (7)	0.833±0.04 (2)
IBk	0.933±0.02 (6)	0.808±0.04 (8)	0.750±0.06 (10)	0.950±0.05 (6)	**0.856**±0.04 (1)
RT	0.923±0.02 (10)	0.793±0.04 (10)	0.740±0.07 (11)	0.939±0.05 (9)	0.718±0.06 (10)
Bagging	0.937±0.02 (4)	0.821±0.04 (5)	0.818±0.06 (4)	0.942±0.05 (8)	0.761±0.05 (5)
MLP	0.936±0.02 (5)	0.852±0.03 (2)	0.787±0.07 (6)	0.960±0.04 (3)	0.825±0.04 (3)
wLap MS ABC	0.925±0.03 (9)	0.801±0.04 (9)	0.841±0.05 (2)	**0.986**±0.01 (1)	0.821±0.05 (4)
1-rule ABC	0.914±0.02 (12)	0.703±0.04 (11)	0.735±0.06 (12)	0.852±0.06 (13)	0.665±0.08 (12)
Seq. cov. wLap ABC	**0.966**±0.01 (1)	0.852±0.02 (2)	0.841±0.01 (2)	0.961±0.01 (2)	0.833±0.00 (2)

According to Table 5.1, sequential covering wLap ABC classification algorithm has the best classification accuracy for the dataset Breast-w and the second best results for other datasets that are Ecoli, Heart-statlog, Iris, and Sonar. On the other hand, the best result belongs to wLapMS ABC for the dataset Iris and wLapMS ABC outperforms the other rule-based classifiers such as Part, Ripper and 1R. At the same time, 1-rule ABC has similar or higher classification performance to 1R which is short but effective and well-known classification technique making classification task by using the values of a single attribute.

Additionally, when analyzed the standard deviations of the classification results for the proposed ABC-based classification algorithms (i.e., sequential covering wLap ABC, 1-rule ABC, and wLapMS ABC), the standard deviations of the proposed methods range between ±0.00 and ±0.08. The minimum standard deviation of the classification accuracies for the proposed sequential covering wLap ABC is ±0.00 while the maximum standard deviation of the classification accuracies for this method is ±0.02. On the other hand, the well-known classification techniques achieve minimum ±0.02 standard deviation value while the maximum standard deviation ±0.07 value is obtained. As a result, the proposed ABC-based classification algorithm (i.e., sequential covering wLap ABC and wLapMS ABC) shows a more stable performance with respect to other well-known classification techniques.

Table 5.2. Average # of rules to achieve classification accuracies over non-private data

Dataset	**wLap MS ABC**	**Part**	**Ripper**	**Seq. Cov. wLap ABC**
Breast-w	6.0	6.45	4.1	14.11
Ecoli	16.0	13.5	9.15	55.46
Heart	8.0	17.4	3.95	42.67
Iris	18.0	4.14	3.62	14.41
Sonar	20.0	7.27	4.56	37.02

Looking at the Table 5.2 to analyze the number of rules to provide the average classification accuracies in Table 5.1, it can be seen that the proposed sequential covering wLap ABC classify instances with higher number of rules with respect to the other rule-based classifiers for the datasets Ecoli and Heart-statlog, which stems from the rule quality measure wLap that prefers the rules with higher precision against coverage. As a result, the number of rules discovered by sequential covering wLap ABC algorithm for these datasets are surplus. On the other hand, it can be seen that wLapMS ABC reaches the close accuracy values to those of sequential covering wLap ABC algorithm with the number of rules 2 per class and 4 per class for these datasets. Therefore, wLapMS taking account of coverage can be used as the rule quality measure in the sequential covering wLap for these datasets to attain similar accuracy values with smaller number of rules. However, wLapMS ABC classify instances with smaller number of rules in which the number of rules are specified by the user according to data at hand.

5.2.2. Performance Evaluation of Rule-based classifiers Using ABC Algorithm over Differentially Private Data

In this section, the experimental results of the proposed rule-based classifiers using ABC algorithm (i.e., 1-rule ABC, wLapMS ABC, and sequential covering wLap ABC) over differentially private data with varying values of privacy parameter ϵ are given. In Table 5.3, the average classification accuracies over differentially private datasets, which are perturbed with Laplace noise to provide privacy with ϵ=1 (i.e., high level privacy), are presented. On the other hand, the average number of rules to satisfy these classification accuracies are given in Table 5.4.

Table 5.3. Average classification accuracies for ϵ=1 over differentially private data

Method	**Dataset**				
	Breast-w	Ecoli	Heart	Iris	Sonar
J48	0.817±0.06 (8)	0.387±0.11 (8)	0.646±0.09 (7)	0.551±0.14 (5)	0.520±0.06 (7)
NB	0.803±0.09 (9)	0.391±0.08 (7)	0.654±0.09 (6)	0.477±0.15 (9)	0.508±0.08 (11)
Bayes Net.	0.829±0.06 (6)	0.427±0.05 (4)	0.619±0.09 (9)	0.311±0.16 (13)	0.510±0.05 (10)
Part	0.823±0.06 (7)	0.383±0.11 (10)	0.658±0.08 (5)	0.509±0.14 (7)	0.515±0.07 (9)
Ripper	0.830±0.05 (5)	0.436±0.06 (3)	0.694±0.07 (4)	0.459±0.15 (12)	0.532±0.07 (6)
1R	0.608±0.12 (13)	0.419±0.07 (6)	0.564±0.09 (14)	0.500±0.18 (8)	0.502±0.06 (13)
Kstar	0.721±0.10 (11)	0.320±0.12 (14)	0.581±0.09 (11)	0.472±0.15 (10)	0.506±0.07 (12)
IBk	0.705±0.10 (12)	0.321±0.11 (13)	0.575±0.08 (12)	0.467±0.14 (11)	0.486±0.06 (14)
RT	0.774±0.07 (10)	0.332±0.10 (11)	0.624±0.08 (8)	0.500±0.15 (8)	0.516±0.07 (8)
Bagging	0.863±0.04 (3)	**0.467**±0.08 **(1)**	0.704±0.07 (3)	0.605±0.13 (4)	0.543±0.07 (5)
MLP	0.849±0.06 (4)	0.425±0.10 (5)	0.613±0.08 (10)	0.547±0.13 (6)	0.582±0.08 (4)
wLap MS ABC	**0.905**±0.04 (1)	0.325±0.12 (12)	0.758±0.10 (2)	**0.744**±0.14 **(1)**	0.781±0.11 (2)
1-rule ABC	0.867±0.08 (2)	0.446±0.11 (2)	0.570±0.14 (13)	0.710±0.15 (3)	0.638±0.18 (3)
Seq. cov. wLap ABC	0.867±0.04 (2)	0.385±0.15 (9)	**0.776**±0.11 (1)	0.726±0.11 (2)	**0.829**±0.09 (1)

When Table 5.3 is examined, the proposed rule-based classifiers such as wLapMS ABC and sequential covering wLap ABC achieve satisfactory classification results for the datasets except Ecoli while the well-known classification techniques do not yield good enough classification results. On the other hand, none of the algorithms used in experiments have satisfactory classification accuracies, but the best result belongs to Bagging classification algorithm with 46.7% accuracy for this dataset. However, the second best result for this dataset is achieved by 1-rule ABC algorithm with 44.6% accuracy.

Table 5.4. Average # of rules to achieve classification accuracies for ϵ=1 over differentially private data

Dataset	**wLap MS ABC**	**Part**	**Ripper**	**Seq. Cov. wLap**
Breast-w	6.0	2.55	2.94	32.56
Ecoli	16.0	45.3	2.09	65.65
Heart	8.0	3.55	2.52	19.39
Iris	18.0	5.11	2.99	26.34
Sonar	20.0	7.24	2.55	30.95

In Table 5.5, the average classification accuracies over differentially private datasets perturbed according to privacy parameter ϵ=2 are introduced and the average number of rules to provide these results are given in Table 5.6. When analyzed Table 5.5, it is clear that sequential covering wLap ABC performs quite well over the datasets except Ecoli. On the other hand, the best classification result for this dataset belongs to 1-rule ABC algorithm and it can be inferred that wLapMS ABC is the second best classifier according to the rankings of the algorithms in Table 5.5.

Table 5.5. Average classification accuracies for ϵ=2 over differentially private data

Meth.	**Dataset**				
	Breast-w	Ecoli	Heart	Iris	Sonar
J48	0.884±0.04 (6)	0.533±0.08 (5)	0.722±0.06 (9)	0.743±0.10 (6)	0.550±0.07 (10)
NB	0.881±0.04 (7)	0.499±0.08 (8)	0.786±0.05 (3)	0.693±0.11 (9)	0.580±0.08 (7)
Bayes Net.	0.894±0.03 (4)	0.515±0.08 (6)	0.758±0.05 (5)	0.708±0.11 (8)	0.522±0.06 (13)
Part	0.886±0.04 (5)	0.515±0.08 (6)	0.727±0.06 (8)	0.717±0.10 (7)	0.557±0.07 (9)
Ripper	0.880±0.04 (8)	0.493±0.07 (9)	0.735±0.05 (6)	0.693±0.11 (9)	0.591±0.07 (6)
1R	0.772±0.06 (12)	0.483±0.07 (10)	0.624±0.09 (14)	0.670±0.14 (11)	0.511±0.06 (14)
Kstar	0.821±0.05 (11)	0.424±0.11 (13)	0.641±0.08 (13)	0.665±0.11 (12)	0.543±0.07 (11)
IBk	0.834±0.02 (10)	0.428±0.11 (12)	0.660±0.06 (12)	0.648±0.11 (13)	0.530±0.07 (12)
RT	0.864±0.04 (9)	0.472±0.08 (11)	0.684±0.06 (11)	0.691±0.11 (10)	0.565±0.06 (8)
Bag.	0.902±0.03 (3)	0.586±0.06 (3)	0.774±0.06 (4)	0.801±0.08 (4)	0.607±0.07 (4)
MLP	0.919±0.03 (2)	0.591±0.08 (2)	0.733±0.07 (7)	0.776±0.09 (5)	0.660±0.07 (3)
wLap MS ABC	0.902±0.03 (3)	0.511±0.09 (7)	0.795±0.08 (2)	**0.887**±0.06 **(1)**	0.850±0.06 (2)
1-rule ABC	0.834±0.05 (10)	**0.610**±0.08 **(1)**	0.701±0.08 (10)	0.803±0.10 (3)	0.600±0.14 (5)
Seq. cov. wLap ABC	**0.961**±0.02 (1)	0.537±0.12 (4)	**0.860**±0.06 (1)	0.864±0.08 (2)	**0.857**±0.06 (1)

Table 5.6. Average # of rules to achieve classification accuracies for ϵ=2 over differentially private data

Dataset	wLap MS ABC	Part	Ripper	Seq. Cov. wLap
Breast-w	6.0	4.34	2.93	25.351
Ecoli	16.0	41.8	3.17	61.85
Heart	8.0	5.77	3.27	17.97
Iris	18.0	7.21	3.57	23.87
Sonar	20.0	8.29	2.73	29.88

In Table 5.7, the average classification accuracies over differentially private datasets that are perturbed according to privacy parameter ϵ=3 are listed, and the average number of rules to provide these results are given in Table 5.8.

When Table 5.7 is examined, sequential covering wLap ABC shows quite good performance compared to other algorithms in the experiments over the differentially private datasets for ϵ=3. Following sequential covering wLap ABC, the best performance belongs to wLapMS ABC algorithm. On the other hand, Bagging algorithm has the best classification accuracies over the datasets for ϵ=3 with respect to other well-known classification techniques and 1-rule ABC algorithm shows very close performance to Bagging algorithm according to the experimental results.

Considering Table 5.4, 5.6, and 5.8, when a general assessment is made in terms of the number of rules of sequential covering wLap ABC, it can be inferred that the number of rules discovered by sequential covering wLap ABC algorithm with ϵ=1, ϵ=2, and ϵ=3 are very close to each other for the datasets, but these numbers decrease with the increase of ϵ values according to the experimental results.

Table 5.7. Average classification accuracies for ϵ=3 over differentially private data

Meth.	**Dataset**				
	Breast-w	Ecoli	Heart	Iris	Sonar
J48	0.906±0.02 (5)	0.623±0.06 (7)	0.741±0.06 (9)	0.804±0.08 (8)	0.609±0.07 (9)
NB	0.893±0.03 (10)	0.624±0.05 (6)	0.817±0.04 (2)	0.831±0.07 (6)	0.647±0.07 (6)
Bayes Net.	0.907±0.03 (4)	0.629±0.06 (5)	0.811±0.05 (3)	0.761±0.09 (14)	0.578±0.08 (13)
Part	0.905±0.03 (6)	0.606±0.07 (8)	0.759±0.05 (7)	0.808±0.08 (7)	0.621±0.07 (8)
Ripper	0.899±0.03 (9)	0.569±0.07 (10)	0.754±0.05 (8)	0.792±0.11 (9)	0.642±0.06 (7)
1R	0.823±0.04 (13)	0.549±0.06 (12)	0.670±0.09 (13)	0.787±0.10 (10)	0.524±0.06 (14)
Kstar	0.859±0.04 (14)	0.505±0.09 (14)	0.688±0.07 (12)	0.758±0.09 (12)	0.583±0.07 (12)
IBk	0.878±0.03 (12)	0.517±0.09 (13)	0.698±0.06 (11)	0.743±0.09 (13)	0.594±0.06 (10)
RT	0.900±0.03 (8)	0.559±0.08 (11)	0.724±0.06 (10)	0.764±0.08 (11)	0.589±0.07 (11)
Bag.	0.915±0.02 (3)	0.675±0.05 (3)	0.782±0.05 (5)	0.875±0.06 (3)	0.660±0.06 (5)
MLP	0.932±0.02 (2)	0.688±0.06 (2)	0.767±0.04 (6)	0.866±0.07 (4)	0.693±0.06 (4)
wLap MS ABC	0.904±0.02 (7)	0.586±0.08 (9)	0.796±0.06 (4)	**0.924**±0.05 **(1)**	0.835±0.07 (2)
1-rule ABC	0.861±0.03 (13)	0.653±0.07 (4)	0.724±0.06 (10)	0.840±0.08 (5)	0.703±0.09 (3)
Seq. cov. wLap ABC	**0.972**±0.01 (1)	**0.698**±0.08 (1)	**0.879**±0.04 (1)	0.902±0.04 (2)	**0.853**±0.05 (1)

Table 5.8. Average # of rules to achieve classification accuracies for ϵ=3 over differentially private data

Dataset	**wLap MS ABC**	**Part**	**Ripper**	**Seq. Cov. wLap**
Breast-w	6.0	5.84	3.4	20.34
Ecoli	16.0	38.4	4.48	57.12
Heart	8.0	6.51	3.62	17.22
Iris	18.0	8.21	4.07	22.11
Sonar	20.0	9.2	2.94	29.85

In the experiments, $\boldsymbol{\alpha}$=0.4 and $\boldsymbol{\beta}$=0.6 are used in the ABC based classification algorithms (i.e., wLapMS ABC and 1-rule ABC) for all over non-private datasets since the different combinations of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters have been tested and the best results which provide both high precision and high coverage observed with $\boldsymbol{\alpha}$=0.4 and $\boldsymbol{\beta}$=0.6 for the datasets except Sonar and Heart-statlog. For the dataset Sonar dataset, these values are determined as $\boldsymbol{\alpha}$=0.3 and $\boldsymbol{\beta}$=0.7 in 1-rule ABC and $\boldsymbol{\alpha}$=0.05 and $\boldsymbol{\beta}$=0.95 in wLapMS ABC. For the dataset Heart-statlog, these values are determined as $\boldsymbol{\alpha}$=0.3 and $\boldsymbol{\beta}$=0.7 in wLapMS ABC and 1-rule ABC. On the other hand, the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters for the differentially private datasets with varying values of $\boldsymbol{\epsilon}$ parameters are given in Table 5.9.

According to Table 5.9 and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameter values yet mentioned above for non-private datasets, the weights for precision are higher for differentially private datasets of $\boldsymbol{\epsilon}$=1and $\boldsymbol{\epsilon}$=2 especially with respect to the non-private datasets in general. Because it has been observed that the precision values of the rules discovered by the proposed rule-based classifiers (i.e., 1-rule ABC and wLapMS ABC) may decrease for differentially private datasets compared to those for non-private datasets. As a result of this situation, the weights of precision have been increased for differentially private datasets.

Table 5.9. α and β values used for 1-rule ABC and wLapMS ABC with ϵ parameters

Dataset	**1-rule ABC**			**wLapMS ABC**		
	ϵ=1	ϵ=2	ϵ=3	ϵ=1	ϵ=2	ϵ=3
Ecoli	α=0.2 β=0.8	α=0.3 β=0.7	α =0.3 β=0.7	α=0.2 β=0.8	α=0.3 β=0.7	α =0.3 β=0.7
Iris	α =0.3 β=0.7	α =0.3 β=0.7	α =0.3 β=0.7	α =0.3 β=0.7	α =0.3 β=0.7	α =0.3 β=0.7
Heart	α=0.1 β=0.9	α=0.2 β=0.8	α=0.3 β=0.7	α=0.1 β=0.9	α=0.1 β=0.9	α=0.1 β=0.9
Sonar	α=0.1 β=0.9	α=0.1 β=0.9	α=0.2 β=0.8	α=0.05 β=0.95	α=0.05 β=0.95	α=0.05 β=0.95
Breast	α=0.2 β=0.8	α=0.2 β=0.8	α=0.2 β=0.8	α=0.2 β=0.8	α=0.2 β=0.8	α=0.2 β=0.8

The number of rules are determined as 2, 6, 4, 3, and 10 per class for the non-private and differentially private version of the datasets Ecoli, Iris, Heart-statlog, Breast-w and Sonar in wLapMS ABC. For sequential covering wLap ABC, no input parameter as the number of rules for the classes are taken from the user since the sequential covering wLap ABC terminates the discovery of classification rules for a class when the number of uncovered instances of the class in the training data is equal to or lower than 5% of instances of the class in the training data as in other some meta-heuristic based classification algorithms (Parpinelli et al., 2002; Celik et al., 2011; Shukran et al., 2011; Talebi and Abadi, 2014).

The parameter values of the proposed ABC algorithm are given in Table 5.10. The values of exceed limit, threshold of iterations, and swarm size are set to the values of parameters used in the experiments of some ABC based rule discovery algorithms in the literature (Celik et al., 2011; Shukran et al., 2011).

To investigate the impact of *F* parameter for the proposed ABC based classification algorithm, the average accuracy values achieved by our proposed 1-rule ABC classification algorithm with different *F* values over some datasets are presented in Table 5.11. When analyzed Table 5.11, *F*=0.6 provides good results for the datasets in terms of accuracy values and standard deviation in general. As a result, in the experiments F value is set to 0.6.

Table 5.10. The parameter values used for the proposed ABC algorithm in the experiments

Parameter	Value
F	0.6
Exceed limit	100
Threshold of iterations	500
Swarm size	20

Table 5.11. The average classification accuracies of 1-rule ABC algorithm with different *F* values at the end of 20 runs

F	Dataset		
	Breast-w	Ecoli	Iris
0.1	0.910±0.03	0.719±0.05	0.862±0.06
0.2	0.937±0.02	0.734±0.04	0.870±0.06
0.3	0.931±0.02	0.712±0.04	0.882±0.05
0.4	0.934±0.03	0.725±0.05	0.873±0.06
0.5	0.921±0.03	0.734±0.04	0.878±0.06
0.6	0.929±0.02	0.758±0.04	0.889±0.05
0.7	0.922±0.03	0.737±0.05	0.880±0.06
0.8	0.927±0.03	0.707±0.05	0.884±0.05
0.9	0.921±0.03	0.741±0.04	0.878±0.06
1.0	0.916±0.03	0.739±0.05	0.884±0.05

In addition to experimental results, in Table 5.12, 5.13, 5.14, and 5.15, we present the statistical properties of the original data and the differentially private version of it for each attribute of one of the datasets which is Iris dataset to show the correlation between the two versions of the dataset. The values in Table 5.12, 5.13, 5.14, and 5.15 are computed by using only one fold of the dataset for different values of ϵ parameter.

To show the relationships between the original and the perturbed versions of the Iris dataset, for each attribute in the dataset we compute the mean, standard deviation, and variance values of the original and perturbed (i.e., differentially private) data. Covariance and correlation between these two versions of the data are also given as well in Table 5.12, 5.13, 5.14, and 5.15.

The covariance of two random variables *X* and *Y* (i.e., the attribute values of original data and differentially private data), *Cov(X, Y)*, measures how these two random variables change in common. If *Cov(X, Y)* is positive, it is said that *X* and *Y* grow meanwhile. If the covariance is negative, then either *X* grows and *Y* reduces, or *Y* grows while *X* reduces. If the covariance is 0, the random variables are uncorrelated (Crawley, 2005; Mivule et al., 2012).

The correlation evaluates the statistical dependency between two random variables *X* and *Y*. The widely used measure is Pearson's correlation coefficient, which is Cr_{xy} = *Cov(X, Y)* /($\sigma_x\sigma_y$). When Cr_{xy} is equal to +1, then it is said that a positive linear relationship is between *X* and *Y* . In other words, when one of *X* and *Y* moves, the other moves in the same direction proportionally. When Cr_{xy} is equal to -1, a negative linear relationship is between *X* and *Y*. In other words, they move in opposite direction with respect to the mean. Values of Cr_{xy} between 1 and -1 show the grade of the relationship between *X* and *Y*, when Cr_{xy} = 0 it is said that *X* and *Y* are uncorrelated (Crawley, 2005; Mivule et al., 2012).

Table 5.12. The statistical properties of the original data and differentially private data for the attribute *Sepal length*

Statistical property	**Original Data**	**Differentially private data**		
		ϵ=1	ϵ=2	ϵ=3
Mean	5.840	5.806	5.823	5.828
Std. dev.	0.821	4.738	2.472	1.758
Variance	0.674	22.454	6.115	3.090
Covariance		0.665	0.670	0.671
Corelation		0.170	0.329	0.465

Table 5.13. The statistical properties of the original data and differentially private data for the attribute *Sepal width*

Statistical property	**Original Data**	**Differentially private data**		
		ϵ=1	ϵ=2	ϵ=3
Mean	3.041	3.192	3.116	3.091
Std. dev.	0.427	2.990	1.542	1.078

Variance	0.182	8.941	2.380	1.162
Covariance		0.198	0.190	0.187
Corelation		0.155	0.288	0.407

Table 5.14. The statistical properties of the original data and differentially private data for the attribute *Petal length*

Statistical property	Original Data	Differentially private data		
		ϵ=1	ϵ=2	ϵ=3
Mean	3.791	4.880	4.336	4.154
Std. dev.	1.753	9.002	4.654	3.307
Variance	3.074	81.046	21.668	10.938
Covariance		1.275	2.175	2.474
Corelation		0.080	0.266	0.426

Table 5.15. The statistical properties of the original data and differentially private data for the attribute *Petal width*

Statistical property	Original Data	Differentially private data		
		ϵ=1	ϵ=2	ϵ=3
Mean	1.212	1.561	1.386	1.328
Std. dev.	0.762	3.468	1.869	1.378
Variance	0.581	12.027	3.495	1.900
Covariance		0.686	0.634	0.616
Corelation		0.259	0.444	0.586

When the values given in the tables are analysed, it is clear that the standard deviation and variance values of the attributes are the highest for ϵ=1where we have high level privacy with respect to those for ϵ=2 and 3 since the σ of Laplace noise ($\sqrt{\sigma} \geq \Delta f/\epsilon$) increases with the decreasing of ϵ values, which results in the lower classification accuracies for our proposed ABC-based classification algorithm and the well-known classification techniques. According to the covariance and correlation values in the tables, the differentially private data and original data are positively correlated for all ϵ parameters. When paid attention, correlation is getting close to +1 which is perfect positive linear relationship with the increase in ϵ parameter since the σ of Laplace noise decreases.

According to the experimental results over differentially private datasets, the low values of ϵ mean more privacy since the σ of Laplace noise added to the training data increases with the low values of ϵ. On the other hand, the proposed ABC-based classification algorithm (i.e., wLapMS ABC, and sequential covering wLap ABC) outperforms the other classification techniques even for low ϵ parameter value (i.e., ϵ=1) over differentially private datasets. At the same time, it shows similar or higher performances to the eleven well-known classification techniques categorized as rule-based, instance-based, artificial neural networks, and decision trees over non-private datasets.

Consequently, it can be inferred from the experimental results that the proposed ABC-based algorithm (i.e., wLapMS ABC, sequential covering wLap ABC, and 1-rule ABC) can be efficiently used to discover classification rules from both of differentially private and non-private datasets.

Finally, as an example, we list the discovered rules by the proposed rule-based classifiers using ABC algorithm for the dataset Iris with the varying values of ϵ parameter that are 1, 2, and 3 in the below.

Discovered rules by 1-rule ABC algorithm for the original data

- IF sepalwidth between (2.825656805542188 and 4.2) and petal length between (1.0275483035881807 and 3.4325862983723074) THEN Class=Iris-setosa
- IF petalwidth between (0.7555154980873141 and 1.591550285750013) THEN Class=Iris-versicolor
- IF petalwidth between (1.6615598596118841 and 2.485139771412621) THEN Class=Iris-virginica

Discovered rules by 1-rule ABC algorithm for the differentially private data of ϵ=1

- IF sepallength between (-0.4657818588095203 and 7.81798912159113) and sepalwidth between (3.4367024522923835 and 13.475396811557221) THEN Class=Iris-setosa
- IF sepallength between (1.7965738479553544 AND 9.947735892333764) and petallength between (-4.877351209861676 and 14.840946422824832) AND petalwidth between (-3.900207983745684 and 15.622057863092135) THEN Class=Iris-versicolor
- IF sepalwidth between (-4.203676566884612 and 5.330409266611276) and petalwidth between (-4.811544342717559 and 16.70518809175706) THEN Class=Iris-virginica

Discovered rules by 1-rule ABC algorithm for the differentially private data of ϵ=2

- IF sepallength between (2.3094805964812286 and 6.005589174191702) and sepalwidth between (3.234663956450949 and 8.78769840577861) THEN Class=Iris-setosa
- IF sepallength between (3.4030086658775294 and 9.11888118695996) and sepalwidth between (-0.6018382834423059 and 5.741130437833356) and petallength between (-0.3790992274166598 and 9.759411034290457) and petalwidth between (-1.4895164507309424 and 8.894713733436117) THEN Class=Iris-versicolor
- IF sepallength between (5.992270489484784 and 13.136278208584546) and sepalwidth between (-0.6018382834423059 and 4.354073994865576) THEN Class=Iris-virginica

Discovered rules by 1-rule ABC algorithm for the differentially private data of ϵ=3

- IF sepallength between (3.0483066594909327 and 5.783856366773856) and sepalwidth between (3.0627592599764184 and 7.225132270519073) THEN Class=Iris-setosa
- IF sepallength between (4.330658877138605 and 7.065085617224864) and sepalwidth between (1.7847105475181175 and 3.9939193533737156) and petalwidth between (-0.2824398576581886 and 2.707199907327946) THEN Class=Iris-versicolor
- IF sepallength between (6.094370460007446 and 10.75751880572303) THEN Class=Iris-virginica

Discovered rules by wLapMS **ABC algorithm for the original data**

- IF sepalwidth between (2.6128858620464 and 4.198770961405614) and petallength between (1.0275483035881807 and 3.4325862983723074) THEN Class=Iris-setosa
- IF petalwidth between (0.658884937089109 and 1.551490481995595) THEN Class=Iris-versicolor
- IF petallength between (4.822724916657451 and 6.547640728124049) THEN Class=Iris-virginica

Discovered rules by wLapMS **ABC algorithm for the differentially private data of ϵ=1**

- IF sepallength between (-0.5741553252266216 and 7.710032367446601) and sepalwidth between (3.4797248176118107 and 13.475396811557221) THEN Class=Iris-setosa
- IF sepallength between (5.418760983485363 and 9.427021051218444) and sepalwidth between (-1.6437412863867822 and 5.975961990452238) and petallength between (-19.647877181396954 and 14.008070900745839) and petalwidth between (-5.860144566110757 and 10.067470834776852) THEN Class=Iris-versicolor
- IF sepalwidth between (-4.203676566884612 and -1.1521252437127623) THEN Class=Iris-virginica

Discovered rules by wLapMS **ABC algorithm for the differentially private data of ϵ=2**

- IF sepallength between (2.1998721360616327 and 5.790014787240827) and sepalwidth between (3.225933019052612 and 8.78769840577861) THEN Class=Iris-setosa
- IF sepallength between (6.883861327777364 and 7.58343387377861) and sepalwidth between (-0.6018382834423059 and 8.78769840577861) THEN Class=Iris-versicolor
- IF sepallength between (6.007146152847497 and 13.136278208584546) and sepalwidth between (-0.6018382834423059 and 4.755415016682328) THEN Class=Iris-virginica

Discovered rules by wLapMS **ABC algorithm for the differentially private data of ϵ=3**

- IF sepallength between (3.143206826520672 and 5.785105682539721) and sepalwidth between (3.272372481038399 and 7.225132270519073) THEN Class=Iris-setosa
- IF sepallength between (6.746448637996943 and 7.155786451881849) THEN Class=Iris-versicolor
- IF sepallength between (6.060417393517042 and 6.706299946590372) THEN Class=Iris-virginica

Discovered rules by sequential covering wLap **ABC algorithm for the original data**

- IF sepallength between (4.3 and 5.801151410237652) and sepalwidth between (2.706037396581697 and 4.2) THEN Class=Iris-setosa
- IF petalwidth between (0.883686444779515 and 1.7856694833214435) THEN Class=Iris-versicolor
- IF petallength between (5.167505565293222 and 6.853649257039449) THEN Class=Iris-virginica

Discovered rules by sequential covering wLap **ABC algorithm for the differentially private data of ϵ=1**

- IF sepallength between (-3.9287920637011675 and 4.606983879501613) THEN Class=Iris-setosa
- IF sepallength between (5.878619308509025 and 10.2954141867077) and sepalwidth between (-4.092185911660381 and 4.49181973967097) THEN Class=Iris-versicolor
- IF sepalwidth between (-4.203676566884612 and -1.1530481351473827) and petallength between (-17.497395794425906 and

21.300966480020417) and petalwidth between (-1.310405722324477 and 4.42195720704856) THEN Class=Iris-virginica

Discovered rules by sequential covering wLap **ABC algorithm for the differentially private data of ϵ=2**

- IF sepalwidth between (3.2236922900515608 and 8.78769840577861) THEN Class=Iris-setosa
- IF sepallength between (5.690474903618018 and 13.136278208584546) and sepalwidth between (-0.6018382834423059 and 3.6117980138789365) and petalwidth between (0.7038950193364588 and 3.3519712178353247) THEN Class=Iris-versicolor
- IF sepallength between (6.085264251527051 and 11.255370021339598) and sepalwidth between (1.7025860342741872 and 8.78769840577861) and petalwidth between (2.9543058257335995 and 7.600426515113706) THEN Class=Iris-virginica

Discovered rules by sequential covering wLap **ABC algorithm for the differentially private data of ϵ=3**

- IF sepalwidth between (0.5987744777051294 and 7.225132270519073) and petalwidth between (-2.8660888700145297 and 0.3672278063761216) THEN Class=Iris-setosa
- IF sepallength between (5.662152412291221 and 7.6596053649338645) and sepalwidth between (2.293427446038246 and 3.992792167552733) THEN Class=Iris-versicolor
- IF sepallength between (6.082623037589031 and 6.667082032103514) and sepalwidth between (2.2464673011335567 and 3.338132684527776) THEN Class=Iris-virginica

5.3. Experimental Results for Privacy Preserving 1R Classifier

In this section, the experimental results of the proposed differentially private 1R classification algorithm with and without applying ABC-DE based feature selection are given and compared with the performance of differentially private NB classification algorithm.

In the experiments, we investigate the performance of the proposed differentially private 1R algorithm for the different values of privacy parameter ϵ. High values of this parameter mean less privacy while the low values of that provide high level privacy. However, the lower the values of ϵ parameter are, the lower classification accuracies are observed but the more privacy is provided; while the higher the values of ϵ parameter are, the higher classification accuracies are obtained but having less privacy. Therefore, it is expected that with the decreasing values of ϵ parameter for a differentially private classifier accuracy also decreases (Rubinstein, 2009; Vaidya et al., 2013; Friedman and Schuster, 2010; Mivule et al., 2012; Bojarski et al. 2015; Fletcher and Islam, 2015; Fletcher and Islam, 2016; Gursoy et al., 2017).

Additionally, we implement both our proposed model, which is differentially private 1R algorithm, and differentially private NB classifier for the performance comparison since Naïve Bayes classifier has been used as the baseline classifier in the literature (Vaidya et al., 2013; Gursoy et al., 2017), and its construction process is very similar to 1R algorithm in terms of usage of the count queries. Therefore, we use the same privacy mechanism to implement both classifiers in this study.

In the experiments, we analyse the classification performances of the proposed differentially private 1R and NB classification algorithms for the values of ϵ parameter that are 0.1, 0.25, 0.5, 1, 2, and 3. We run the classifiers for 10 times with 10 fold cross validation and give the average accuracy values with the standard deviations for the different ϵ parameters in Table 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, and 5.22 for the datasets Cong. votes, Mushroom, Heart-statlog, Spect-h, Credit, Breast-w, and Nursery respectively.

Table 5.16. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Congressional votes

Epsilon value	Non private	3	2	1	0.5	0.25	0.1
1R with F.S.	0.956 ±2E-16	0.950 ±0.004	0.944 ±0.008	0.903 ±0.021	0.839 ±0.034	0.721 ±0.038	0.656 ±0.078
NB with F.S.	0.933 ±0.001	0.928 ±0.005	0.927 ±0.006	0.908 ±0.006	0.900 ±0.014	0.844 ±0.027	0.652 ±0.055
1R without F.S.	0.956 ±2E-16	0.886 ±0.014	0.800 ±0.017	0.739 ±0.048	0.625 ±0.038	0.558 ±0.033	0.570 ±0.046
NB without F.S.	0.901 ±0.001	0.893 ±0.004	0.886 ±0.006	0.866 ±0.009	0.799 ±0.025	0.701 ±0.033	0.603 ±0.063

According to the experimental results in Table 5.16, it is clear that differentially private 1R algorithm shows very close performance to differentially private NB algorithm for the values of ϵ parameter that are 3, 2, 1, and 0.5. However, differentially private NB with 84.4% accuracy outperforms differentially private 1R with 72.1% accuracy for ϵ=0.25.

Table 5.17. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Mushroom

Epsilon value	Non private	3	2	1	0.5	0.25	0.1
1R with F.S.	0.985 ±2E-16	0.972 ±0.011	0.971 ±0.011	0.969 ±0.012	0.931 ±0.018	0.893 ±0.029	0.770 ±0.042
NB with	0.979 ±0.006	0.967 ±0.008	0.960 ±0.009	0.950 ±0.008	0.939 ±0.008	0.900 ±0.011	0.831 ±0.035

F.S.							
1R without F.S.	0.950 ±0.008	0.968 ±0.007	0.946 ±0.010	0.862 ±0.016	0.760 ±0.029	0.641 ±0.025	0.560 ±0.028
NB without F.S.	0.957 ±2E-16	0.929 ±0.001	0.926 ±0.004	0.911 ±0.002	0.873 ±0.006	0.803 ±0.008	0.688 ±0.020

When examined Table 5.17, it can be seen that differentially private 1R algorithm shows higher performance than differentially private NB algorithm for the values of ϵ parameter that are 3, 2, 1, 0.5, and 0.25. However, differentially private NB with 83.1% accuracy outperforms differentially private 1R with 77% accuracy for ϵ=0.1.

Table 5.18. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Heart-statlog

Epsilon value	**Non private**	$\mathbf{3}$	$\mathbf{2}$	$\mathbf{1}$	$\mathbf{0.5}$	$\mathbf{0.25}$	$\mathbf{0.1}$
1R with F.S.	0.722 ±0.01	0.617 ±0.01	0.575 ±0.03	0.550 ±0.03	0.502 ±0.02	0.532 ±0.03	0.530 ±0.02
NB with F.S.	0.845 ±0.002	0.645 ±0.02	0.637 ±0.02	0.574 ±0.02	0.533 ±0.02	0.537 ±0.03	0.506 ±0.04
1R without F.S.	0.722 ±0.01	0.535 ±0.02	0.517 ±0.03	0.512 ±0.03	0.503 ±0.03	0.515 ±0.02	0.506 ±0.02
NB without F.S.	0.835 ±0.006	0.564 ±0.03	0.550 ±0.03	0.540 ±0.02	0.508 ±0.04	0.503 ±0.03	0.492 ±0.02

According to the experimental results in Table 5.18, differentially private 1R and differentially private NB have quite close classification results, but NB has slightly higher classification accuracies with respect to 1R algorithm.

In Table 5.19, differentially private 1R and differentially private NB performs very similar, but 1R yields slightly higher classification results than those of NB algorithm.

Table 5.19. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Spect-h

Epsilon value	Non private	3	2	1	0.5	0.25	0.1
1R with F.S.	0.723 ±0.001	0.690 ±0.01	0.662 ±0.01	0.610 ±0.02	0.592 ±0.03	0.539 ±0.03	0.512 ±0.03
NB with F.S.	0.693 ±0.02	0.643 ±0.02	0.606 ±0.02	0.565 ±0.02	0.548 ±0.02	0.521 ±0.02	0.509 ±0.02
1R without F.S.	0.723 ±0.001	0.571 ±0.02	0.564 ±0.02	0.526 ±0.01	0.510 ±0.03	0.504 ±0.02	0.515 ±0.02
NB without F.S.	0.681 ±0.03	0.619 ±0.02	0.605 ±0.01	0.568 ±0.02	0.512 ±0.01	0.515 ±0.03	0.484 ±0.03

Table 5.20. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Credit

Epsilon value	Non private	3	2	1	0.5	0.25	0.1
1R with F.S.	0.855 ±1E-15	0.772 ±0.01	0.733 ±0.01	0.680 ±0.02	0.611 ±0.02	0.540 ±0.03	0.514 ±0.02
NB with F.S.	0.693 ±0.02	0.589 ±0.01	0.566 ±0.02	0.533 ±0.02	0.521 ±0.02	0.501 ±0.01	0.503 ±0.01
1R without F.S.	0.855 ±1E-15	0.689 ±0.01	0.628 ±0.01	0.588 ±0.01	0.554 ±0.02	0.516 ±0.02	0.511 ±0.01
NB without F.S.	0.681 ±0.03	0.535 ±0.02	0.530 ±0.01	0.514 ±0.01	0.495 ±0.01	0.501 ±0.01	0.490 ±0.01

According to the experimental results in Table 5.20, it can be seen easily that differentially private 1R outperforms differentially private NB algorithm for Credit dataset.

Table 5. 21. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Breast-w

Epsilon value	Non private	3	2	1	0.5	0.25	0.1

1R with F.S.	0.917 ±0.003	0.861 ±0.01	0.835 ±0.01	0.817 ±0.01	0.768 ±0.02	0.655 ±0.06	0.595 ±0.05
NB with F.S.	0.960 ±0.001	0.924 ±0.006	0.915 ±0.006	0.857 ±0.01	0.792 ±0.02	0.636 ±0.03	0.570 ±0.05
1R without F.S.	0.917 ±0.003	0.772 ±0.02	0.748 ±0.02	0.675 ±0.03	0.614 ±0.04	0.576 ±0.03	0.536 ±0.06
NB without F.S.	0.973 ±4E-4	0.893 ±0.004	0.843 ±0.01	0.779 ±0.01	0.668 ±0.03	0.590 ±0.05	0.545 ±0.03

In Table 5.21, the experimental results of differentially private NB algorithm outperforms 1R algorithm for $\epsilon = 3$ and $\epsilon = 2$ for the dataset Breast-w. However, NB shows very close performance to 1R algorithm for ϵ=1, ϵ=0.5, ϵ=0.25 and ϵ =0.1.

Table 5.22. Average classification accuracies of differentially private 1R and NB according to different ϵ parameters for the dataset Nursery

Epsilon value	**Non private**	**3**	**2**	**1**	**0.5**	**0.25**	**0.1**
1R with F.S.	0.709 ±2E-16	0.709 ±2E-16	0.709 ±2E-16	0.708 ±0.001	0.695 ±0.007	0.550 ±0.039	0.283 ±0.019
NB with F.S.	0.898 ±2E-4	0.894 ±0.001	0.887 ±0.002	0.859 ±0.004	0.758 ±0.015	0.534 ±0.014	0.387 ±0.018
1R without F.S.	0.709 ±2E-16	0.709 ±2E-16	0.709 ±2E-16	0.707 ±0.001	0.691 ±0.008	0.545 ±0.023	0.347 ±0.026
NB without F.S.	0.902 ±2E-4	0.895 ±0.001	0.886 ±0.002	0.854 ±0.004	0.740 ±0.011	0.549 ±0.016	0.386 ±0.014

According to the experimental results in Table 5.22, NB outperforms 1R algorithm for Nursery dataset.

When a general classification performance assessment is made for the proposed differentially private 1R algorithm, it can be inferred that the proposed method shows similar performance to differentially private NB which is used as a baseline for differentially private classification in the literature (Vaidya et al., 2013; Gursoy et al., 2017), and the proposed differentially private 1R algorithm can be used as an efficient private rule-based classifier.

In our proposed method, we apply ABC-DE based feature selection method as a pre-processing step to reduce the number of required count queries to build differentially private 1R and NB classification algorithms. According to the experimental results, it has been demonstrated that the classification accuracies of differentially private 1R and NB algorithms increase for all values of ϵ parameter for the majority of the datasets when ABC-DE based feature selection is applied. The number of attributes determined by applying the ABC-DE based feature selection method are presented in Table 5.23.

According to the values in Table 5.23, the number of count queries required to build differentially private 1R and NB classification algorithms decreases at least 70%, 70.5%, 60%, 57.1%, 77.2%, 12.5%, and 56.5% with applying of ABC-DE based feature selection method as a pre-processing step, which enables to have higher level of privacy with more accuracy. As an example, differentially private NB classification algorithm yields 84.4% accuracy with applying of ABC-DE based feature selection for ϵ=0.25 (i.e., quite high level privacy) over Cong. votes. dataset while it achieves 70.1% accuracy without applying the feature selection.

Table 5.23. # of attributes selected with ABC-DE based Feature Selection Method

Dataset	# of Attributes	# of Attributes with F.S.
Breast-w	10	3
Cong. Vot.	17	5
Credit	15	6
Heart	14	6
Mushroom	22	5
Nursery	8	7
Spect-h	23	10

Finally, when a general comparison is made between differentially private 1R and NB classification algorithms for ϵ=1 which is a most commonly used privacy value, 1R algorithm outperforms NB for 3 datasets that are Credit, Mushroom, and Spect-h among 7 datasets. However, 1R algorithm achieves 0.903, 0.550, and 0.817 average accuracies while NB reaches 0.908, 0.574, and 0.857 average accuracies for the datasets Cong. votes, Heart-statlog, and Breast-w. The differences of average accuracies for the algorithms are only 0.005, 0.024, and 0.04 for these datasets. Therefore, differentially private 1R algorithm has quite similar performance to differentially private NB classification algorithm for these datasets. Consequently, differentially private 1R algorithm can also be efficiently used for privacy preserving classification.

5.4. Comparison of the Proposed Methods with Recent Studies in the Literature

In this section, the proposed methods are compared with the recent studies in the literature. Therefore, the section is divided into two subsections. The first subsection includes the comparison with the recent studies of rule-based classifiers using ABC while the second subsection contains the comparison with differentially private classification algorithms in the literature.

5.4.1. Comparison of the Proposed Rule-based Classifiers Using ABC with Recent Rule-based Classifiers Using ABC over Non-private Data

In this subsection, the classification results of the proposed rule-based classifiers using ABC (i.e., 1-rule ABC, wLapMS ABC, and sequential covering wLap ABC) and the other rule-based classifiers using ABC in the literature are compared in Table 5.24 for the common datasets used in the experiments.

Table 5.24. Classification accuracies for our proposed rule-based classifiers using ABC and the other rule-based classifiers using ABC in the literature

Dataset	Celik et al. (2011)	Shukran et al. (2011)	Talebi and Abadi (2014)	1-rule ABC	wLap MS ABC	Seq. Cov. wLap ABC
Breast	95.42%	-	-	91.4%	92.5%	**96.6**%
Ecoli	-	-	83.39%	70.3%	80.1%	**85.2**%
Iris	-	94.8%	-	85.2%	**98.6**%	96.1%
Sonar	-	-	72.1%	66.5%	82.1%	**83.3**%

According to the classification results given in Table 5.24, the proposed ABC-based classification algorithms (i.e., wLapMS ABC and sequential covering wLap ABC) have very close or higher classification accuracies to the other ABC-based classication algorithms in the literature for the datasets Breast-w, Ecoli, and Iris. However, higher classification accuracies are achieved by our proposed rule-based classification algorithm (i.e., sequential covering wLap ABC and wLapMS ABC) for the dataset Sonar.

5.4.2. Comparison of the Proposed Rule-based Classifiers with Recent Differentially Private Classification Algorithms for the used Datasets

In Table 5.25, the average classification accuracies of our proposed classifiers and the average accuracy values reached by other differentially private classification algorithms in the literature over the most commonly used datasets that are Cong. votes, Mushroom, and Nursery are given.

When Table 5.25 is examined, the proposed differentially private 1R algorithm achieves satisfactory classification results considering the classification results of the other differentially private classification algorithms in the literature (Friedman and Schuster, 2010; Jagannathan et al., 2012; Vaidya et al., 2013; Fletcher and Islam, 2015) for the datasets Mushroom, Nursery, and Cong. votes.

According to Table 5.25, differentially private ID3 is an indirect rule-based classification algorithm and achieves 83.0% and 40.4% classification accuracies for the datasets Mushroom and Nursery while 86.2% and 70.7% of classification accuracies are attained by the proposed differentially private 1R algorithm. On the other hand, by applying the ABC-DE based feature selection method over the dataset Mushroom, differentially private 1R classification algorithm reaches 96.9% classification accuracy which is the highest value of accuracy values reported in the literature for this dataset.

Briefly, the proposed differentially private 1R algorithm achieves satisfactory classification results considering the classification results of the other differentially private classification algorithms in the literature (Friedman and Schuster, 2010; Jagannathan et al., 2012; Vaidya et al., 2013; Fletcher and Islam, 2015) over the most commonly used datasets in the literature.

As a result, it can be inferred that the proposed differentially private 1R algorithm can be used as a differentially private rule-based classification algorithm taking account of our experimental results and the results of other differentially private classification algorithms in the literature. Also, the classification accuracy of the proposed differentially private 1R algorithm can be improved by applying the ABC-DE feature selection as a pre-processing step.

Table 5.25. Classification accuracies with ϵ=1for the proposed 1R classifier and the other differentially private classifiers in the literature.

Method	**Dataset**		
	Cong. votes	Mushroom	Nursery
Diff.Priv. ID3 (Friedman and Schuster, 2010)	-	83.0%	40.4%
Diff. Priv. RT (Jagannathan et al., 2012)	**90.0**%	92.2%	63.9%
Diff. Priv. NB (Vaidya et al., 2013)	86.6%	91.1%	85.4%
Diff.Priv. Random Forest (Fletcher and Islam, 2015)	-	93.5%	69.0%
Diff. Priv. 1R with F.S.	**90.0**%	**96.9**%	70.8%
Diff. Priv. NB with F.S.	**90.0**%	95.0%	**85.9**%
Diff. Priv. 1R without F.S.	73.9%	86.2%	70.7%
Diff. Priv. NB without F.S.	86.6%	91.1%	85.4%

6. CONCLUSION

Privacy preserving data mining is a sub-field of data mining and its goal is to protect the privacy of individuals while making possible to apply data mining techniques. Recently, differential privacy has been proposed to present maximum security to the statistical databases by minimizing the chances for the disclosure of the sensitive information of records. Therefore, some implementations of classification algorithms such as decision trees, random trees, random forests, NB, *k*-NN etc. with differential privacy have been performed in the literature. Although the success of the rule-based classifiers using meta-heuristics such as Ant-Miner, Bee-miner etc. in data mining has been demonstrated, any implementation of these classification algorithms with differential privacy has not been studied in the literature to our best knowledge.

Motivated by this, we investigate the performance of some rule-based classifiers using meta-heuristics under differential privacy guarantee in the first implementation of this thesis. To make performance comparison, eleven well-known classification techniques categorized as rule-based method, instance-based technique, artificial neural networks, and decision trees are used. The experiments are performed over both of non-private and differentially private datasets. The proposed rule-based classifiers especially wLapMS ABC, and sequential covering wLap ABC outperform the other well-known classification techniques when high level privacy (i.e., ϵ=1) is applied. The experimental results show that the proposed rule-based classifiers can be efficiently used to discover classification rules from both of the differentially private and non-private databases.

In the second differential privacy implementation of this thesis, differentially private 1R classifier is developed to cover the gap for the lack of that implementation of 1R classification algorithm with differential privacy which has not been developed in the literature to our best knowledge. On the other hand, 1R is a simple classification algorithm and it discovers the rules which result in slightly lower accuracy with respect to the state of the art classification algorithms, but its rules are of a small number and very easy to interpret for humans.

In this second rule-based implementation of differential privacy with meta-heuristics, an ABC-DE based feature selection method proposed by Zorarpacı and Özel (2016) is applied as a pre-processing step to reduce the required count queries sent to the differentially private database on a large scale during the construction of 1R.

Additionally, for the performance evaluation of the proposed differentially private 1R, the same privacy preserving model is used for both differentially private 1R and differentially private NB classifier. For the performance comparison of the proposed differentially private 1R classifier, differentially private NB classifier is utilized as it is the baseline for differentially private classification in the literature.

The experimental results show that the proposed differentially private 1R classification algorithm has very similar or higher performances with respect to differentially private NB classification algorithm for the different values of privacy parameter ϵ. Also, the accuracy values of the differentially private 1R and NB classifiers can be increased for all values of ϵ parameter by applying ABC-DE based feature selection.

In this thesis, we investigate the performance of rule-based classifiers using ABC meta-heuristic algorithm for the implementation. However other meta-heuristics or evolutionary algorithms, or hybrid approaches of these algorithms can be considered to develop rule-based classifiers under differential privacy guarantee as future work. Additionally differentially private feature selection method may be developed to build differentially private 1R and NB classification algorithms as future work as well. 1R algorithm may be updated to develop a more accurate version of differentially private 1R classifier.

REFERENCES

Adam N. R., and Worthmann J. C., 1989. Security-control methods for statistical databases: A comparative study. ACM Computing Surveys., vol. 21., no. 4., pp. 515-556.

Aha D. W., Kibler D., and Albert M. K., 1991. Instance-based learning algorithms. Machine Learning., vol. 6, no. 1, pp. 37-66.

Antonova D., 2015. Practical Differential Privacy in High Dimensions. The University of Edinburhg, Institute for Adaptive and Neural Computation, School of Informatics, Master of Philosophy.

Azevedo P. J. and Jorge A. M., 2007. Comparing rule measures for predictive association rules. In Pro. 18th European Conference on Machine Learning, Warsaw, Poland, pp. 510–517.

Blum A., Dwork C., Mcsherry F., and Nissim K., 2005. Practical privacy: The SuLQ framework," in Proc. International Conference on Principles of Data Systems, Baltimore, Maryland.

Brieman L., 1996. Bagging predictors. Machine Learning., vol. 24, no. 2, pp. 123-140.

Brieman L., 2001. Random forests. Machine Learning., vol. 45, no. 1, pp. 5-32.

Bojarski M., Choromanska A., and Choromanski K., 2015. Differentially-and non-differentially-private random decision trees. arXiv preprint arXiv:1410.6973v2.

Celik M., Karaboğa D., and Köylü F., 2011. Artificial bee colony data miner (ABC-miner)," In Proc. International Symposium on Innovations and Intelligent Systems and Applications(INISTA).

Celik M., Köylü F., and Karaboğa D., 2016. CoABCMiner: An algorithm for cooperative rule classification system based on artificial bee colony. International Journal on Artificial Intelligence Tools, 25(1).

Chaudhuri K., and Monteleoni C., 2008. Privacy preserving logistic regression. Advances in Neural Information Processing Systems, pp. 289-296.

Chaudhuri K., Monteleoni C., and Sarwate A. D., 2011. Differentially private empirical risk minimization. Journal of Machine Learning Research, vol. 12, pp. 1069-1109.

Chui C., and Hsu P. L., 2005. A constraint-based genetic algorithm approach for mining classification rules. IEEE Transactions on Systems, Man and Cybernetics., vol. 35, no. 2, pp. 205-220.

Clark P. and Niblett T., 1989. The CN2 induction algorithm. Machine Learning., vol. 3, no. 4, pp. 261-283.

Cleary J. E. and Trigg L. E., 1995. K*: An instance based learner using an entropic distance measure. In Proc. 12th International Conference on Machine Learning, Thaoe, California, pp. 108–114.

Cohen W. W., 1995. Fast effective rule induction. In Proc. Twelfth International Conference on Machine Learning, Tahoe, California.

Crawley M. J., 2005. Statistics: An introduction using R. John Wiley and Sons, pp. 93-95.

Debie E., Shafi K., Lokan C., and Merrick K., 2013. Investigating differential evolution based rule discovery in learning classifier systems. In Proc. IEEE Symposium on Differential Evolution (SDE), pp. 77–84.

De Falco I., 2013. Differential evolution for automatic rule extraction from medical databases. Applied Soft Computing., vol. 13, no. 2, pp. 1265-1283.

Deng F., Chen J., Wang Y., and Gong K., 2013. Measurement and calibration method for an optical encoder based on adaptive differential evolution-fourier neural networks. Measurement Science and Technology, 24(5).

Duch W., Jankowski N., Grabczewski K., and Adamczak R., 2000. Optimization and interpretation of rule-based classifiers. In Proc. Intelligent Information Systems Symposium, Bystra, Poland, pp. 1–13.

Dwork C., McSherry F., Nissim K., and Smith A., 2006. Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography., pp. 265-284, Springer.

Dwork C., 2008. Differential privacy: A survey of results. In Proc. 5th International Conference on Theory and Applications of Models of Computation, Xi'an, China.

Dwork C. and Roth A., 2014. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, Vol. 9, Nos. 3-4 pp. 211-407.

Edlich T., 2017. Differential privacy: Robust data mining techniques. Seminar, Munchen Technique University, Department of Informatics Data Mining and Analytics.

Fidelis M. V., Lopes H. S., and Freitas A. A., 2000. Discovering comprehensible classification rules with a genetic algorithm. In Proc. Congress on Evolutionary Computation, San Diego, CA, pp. 805–810.

Fletcher S., and Islam M. Z., 2015. A differentially private decision forest. In Proc. 13th Australasian Data Mining Conference, Sydney, Australia.

Fletcher S., and Islam M. Z., 2016. Decision tree classification with differential privacy: A survey. arXiv preprint arXiv:1611.01919v1.

Fletcher S., and Islam M. Z., 2016. Decision tree classification with differential privacy: A survey. ACM Computing Surveys.

Fletcher S., and Islam M. Z., 2017. Differentially private random decision forests using smooth sensitivity. Expert Systems with Applications.

Frank E. and Witten I. H., 1998. Generating accurate rule sets without global optimization. In Proc. Fifteenth International Conference on Machine Learning, pp. 144-151.

Friedman J. H., Kohavi R., and Yun Y., 1996. Lazy decision trees. In: AAAI/IAAI., vol. 1, pp.717-724.

Friedman A., and Schuster A., 2010. Data mining with differential privacy. In Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA.

Fukuchi K., Tran Q. K., and Sakuma J., 2017. Differentially private empirical risk minimization with input perturbation. arXiv preprint arXiv:1710.07425v1.

Gao L., Ye M., and Wu C., 2017. Cancer classification based on support vector machine optimized by particle swarm optimization and artificial bee colony**.** Molecules, 22(12).

Goldreich O., 2004. Foundations of Cryptography, vol. II, Cambridge University Press.

Gursoy M. E., Inan A., Nergiz M. E., and Saygın Y., 2017. Differentially private nearest neighbor classification. Data Mining and Knowledge Discovery, vol. 31, no. 5, pp. 1544-1575.

Hilderman R. J. and Hamilton H. J., 1999, Knowledge Discovery and Interestingness Measures: A survey, Technical Report, University of Regina, Canada. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.588.4570and rep=rep1andtype=pdf

Holte R. C., 1993. Very simple classification rules perform well on most commonly used datasets. Machine Learning., vol. 11, pp. 63-90.

Hyma J. et al., 2018. Heteregenous data distortion for privacy preserving SVM classification. Smart Intelligent Computing and Applications, pp. 459-468.

Inan A., Kaya S. V., Saygın Y., Savaş E., Hintoğlu A. A., and Levi A., 2007. Privacy preserving clustering on horizontally partioned data. Data and Knowledge Engineering, 63(3), pp.646-666.

Jagannathan G., Pillaipakkamnatt K., and Wright R. N., 2012. A practical differentially private random decision tree classifier. Transactions on Data Privacy., no. 5, pp. 273-295.

Jagannathan G., Monteleoni C., and Pillaipakkamnatt K., 2013. A semi-supervised learning approach to differential privacy. In Proc. 13th International Conference on Data Mining Workshops, TX, USA.

Jensen F., 1996. An Introduction to Bayesian Networks. UCL Press/Springer-Verlag, New York.

Ji Z., Lipton Z. C., and Elkan C., 2014. Differential privacy and machine learning: A survey and review. arXiv preprint arXiv:1412.7584v1.

Karaboğa D., 2005. An idea based on honey bee swarm for numerical optimization. Erciyes University, Engineering Faculty, Computer Engineering Department. Technical report.

Kantarcıoğlu M., and Clifton C., 2004. Privately computing a distributed k-nn classifier. In Proc. European Conference on Principles of Data Mining and Knowledge Discovery., PKDD, pp. 279–290.

Lavrač N., Flach P., and Zupan B., 1999. Rule evaluation measures: A unifying view. In Pro. International Conference on Inductive Logic Programming, Bled, Slovenia, pp. 174–185.

Li J., H., Guo K. Z., Wang D., and Hu J., 2013. Differential evolution for rule extraction and its application in recognizing oil reservoir. Journal of Digital Information and Management, vol. 11, no. 6, pp. 435-440.

Lindell Y. and Pinkas B., 2002. Privacy preserving data mining. J. Cryptology, vol. 15, no. 3, pp. 177-206.

Liu L., Kantarcıoğlu M., and Thuraisingham B., 2008. The applicability of the perturbation based privacy preserving data mining for real-world data. Data and Knowledge Engineering, 65, pp. 5-21.

Liu L., Kantarcıoğlu M., and Thuraisingham B., 2009. Privacy preserving decision tree mining from perturbed data. In Proc. 42nd Hawaii International Conference on System Sciences, pp. 1-10.

Li N., Li T., and Venkatasubramanian S., 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In Proc. IEEE 23rd Int. Conf. Data Eng. (ICDE), pp. 106-115.

Machanavajhala A., Kifer D., Gehrke J., and Venkitasubramaniam M., 2007. l-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data, vol. 1, no. 1.

Michalak M., Sikora M., and Wrobel L., 2015. Rule quality measures settings in a sequential covering rule induction algorithm-An empirical approach. In Proc. Federated Conference on Computer Science and Information Systems, Lodz, Poland, pp. 109–118.

Mivule K., Turner C., and Ji S. Y., 2012. Towards a differential privacy and utility preserving machine learning classifier. Procedia Computer Science.

Mohamed A. W., Sabry H. Z., and Korshid M., 2012. An alternative differential evolution algorithmfor global optimization. Journal of Advanced Research, 3(2), pp. 149-165.

Mohana S., and Arul Mary S. A., 2017. Heuristics for privacy preserving data mining: An evaluation**.** In Proc. International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies.

Murphy K. P., 2006. Naïve Bayes classifiers. Technical Report. [Online]. Available: https://datajobsboard.com/wp-content/uploads/2017/01/Naive-Bayes-Kevin-Murphy.pdf

Omran M. G. H., Engelbrecht A. P., and Salman A., 2005. Differential evolution methods for unsupervised image classification. In Proc. IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, pp. 966–973.

Palanisamy S., and Kanmani S., 2012. Classifier ensemble design using artificial bee colony based feature selection. International Journal of Computer Science Issues, 9(3).

Parpinelli R. S., Lopes H. S., and Freitas A. A., 2002. Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation., vol. 6, no. 4, pp. 321-332.

Patil A., and Singh S., 2014. Differential private random forest. In Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, India.

Preethi P., K., Kumar P., Ullhaq M. R., Naveen A., and Janapana H., 2018. Privacy preserving data clustering using a heteregeneous data distortion. Smart Intelligent Computing and Applications, pp. 477-486.

Quinlan J. R., 1993. C4.5: Programs for Machine learning. Morgan Kaufmann Publishers.

Rana S., Gupta S. K., and Venkatesh S., 2015. Differentially private random forest with high utility. In Proc. International Conference on Data Mining, Atlantic City, NJ, USA.

Rangasamy R. R., and Duraisamy R., 2018. Ensemble of artificial bee colony optimization and random forest technique for feature selection and classification of protein function family prediction. Soft Computing in Data Analytics.

Ravi A. T., and Chitra S., 2014. Privacy preserving data mining using differential evolution-artificial bee colony algorithm. International Journal of Applied Engineering Research, 9(23).

Ravi A. T., and Chitra S., 2015. Privacy preserving data mining. Research Journal of Applied Sciences, Engineering and Technology, 9(8), pp. 616-621.

Ravi V., Naveen N., and Rao C. R., 2012. Privacy preserving data mining using particle swarm optimisation trained auto-associative neural network:

an application to bankruptcy prediction in banks. International Journal of Data Mining, Modeklling and Management, 4(1).

Rubinstein B. I. P., Bartlett P. L., Huang L., and Taft N., 2009. Learning in a large function space: Privacy preserving mechanisms for SVM learning. Computing Research Repository.

Rumelhart D. E., Hinton G. E., and Williams R. J., 1986. Learning representation by back propogation errors. Nature., vol. 323, pp. 533-536.

Sanchez D., Domingo-Ferrer J., Martinez S., and Soria-Comas J., 2015. Utility-preserving differentially private data releases via individual ranking microaggregation. Information Fusion.

Samarati P. and Sweeney L., 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In Proc. IEEE Symp. Res. Secur. Privacy, pp. 384-393.

Samarati P. and Sweeney L., 1998. Generalizing data to provide anonymity when disclosing information. In Proc. PODS.

Samarati P., 2001. Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 16, pp. 1010-1027.

Sarwate A. D., and Chaudhuri K., 2013. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. IEEE Signal Processing Magazine, 30(5), pp. 86-94.

Shah H., Herawan T., Ghazali R., and Naseem R., 2014. An improved gbest guided artificial bee colonyalgorithm for classification and prediction tasks. Lecture Notes in Computer Science.

Storn R. and Price K., 1997. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization., vol. 11, no. 4, pp. 341-359.

Su D., Cao J., Li N., Bertino E., and Jin H., 2015. Differentially private k-means clustering. arXiv preprint arXiv:1504.05998v1.

Su H., Yang Y., and Zhao L., 2010. Classification rule discovery with DE/QDE algorithm. Expert Systems with Applications., vol. 37, no. 2, pp. 1216-1222.

Shukran M. A. M., Chung Y. Y., Yeh W., Wahid N., and Zaidi A. M. A., 2011. Artificial colony based data mining algorithms for classification tasks. Modern Applied Science, 5(4).

Tan C. H., Yap K. S., and Yap H. J., 2012. Application of genetic algorithm for fuzzy rules optimization on semi expert judgment automation using Pittsburg approach. Applied Soft Computing., vol. 12, no. 8, pp. 2168-2177.

Triguero I., García S., and Herrera F., 2010. A preliminary study on the use of differential evolution for adjusting the position of examples in nearest neighbor classification. In Proc. IEEE Congress on Evolutionary Computation, Barcelona, Spain, pp. 1–8.

Tsang S., Kao B., Yip K. Y., Ho W., and Lee S. D., 2011. Decision trees for uncertain data. IEEE Transactions on Knowledge and Data Engineering, 23(1), pp. 64-78.

Vaghashia H. and Ganatra A., 2015. A survey: Privacy preservation techniques in data mining. International Journal of Computer Applications., vol. 119, no. 4, pp. 20-26.

Vaidya J., Shafiq B., Basu A., and Hong Y., 2013. Differentially private naïve bayes classification. In Proc. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies, pp. 571–576.

Vijarayani S., and Prabha M. S., 2011. Association rule hiding using artificial bee colony algorithm. International Journal of Computer Applications, 33(2), pp. 41-47.

Vijayarani S., and Janakiram M., 2016. Genetic algorithm based confidential data protection in privacy preserving data mining. International Journal of Advanced Research in Computer and Communication Engineering, 5(4), pp. 158-164.

Yao Y. Y. and Zhong N., 1999. An analysis of quantitative measures associated with rules. Lecture Notes in Computer Science., vol. 1574, pp. 479-488. doi: 10.1007/3-540-48912-6_64, [Online]. Avaliable: {https://doi.org/10.1007/3-540-48912-6_64.}.

Zhang J., Zhang Z., Xiao X., Yang Y., and Winslett M., 2012. Functional mechanism: Regression analysis under differential privacy. In Proc. International Conference on Very Large Databases, pp. 1364-1375.

Zhang Y. and Wu L., 2011. Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach. Entropy, vol 13, no. 4, p. 841-859.

Zorarpacı E. and Özel S. A., 2016. A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Systems with Applications., vol. 62, pp. 91-103.

CURRICULUM VITAE

Ezgi ZORARPACI was born in Ankara, in 1986. She received the B.E in Computer Engineering from Selçuk University in Konya, Turkey, in 2011. She received the MSc degree from Computer Engineering, Çukurova University in 2014. Since 2012, she has been working as a research assistant.

APPENDIX

Original data for Iris dataset

sepallength	sepalwidth	petallength	petalwidth
5,10	3,50	1,40	0,20
4,90	3,00	1,40	0,20
4,70	3,20	1,30	0,20
4,60	3,10	1,50	0,20
5,00	3,60	1,40	0,20
5,40	3,90	1,70	0,40
4,60	3,40	1,40	0,30
5,00	3,40	1,50	0,20
4,40	2,90	1,40	0,20
4,90	3,10	1,50	0,10
5,40	3,70	1,50	0,20
4,80	3,40	1,60	0,20
4,80	3,00	1,40	0,10
4,30	3,00	1,10	0,10
5,80	4,00	1,20	0,20
5,40	3,90	1,30	0,40
5,10	3,50	1,40	0,30
5,70	3,80	1,70	0,30
5,10	3,80	1,50	0,30
5,40	3,40	1,70	0,20
5,10	3,70	1,50	0,40
4,60	3,60	1,00	0,20
4,80	3,40	1,90	0,20
5,00	3,00	1,60	0,20
5,20	3,50	1,50	0,20
5,20	3,40	1,40	0,20
4,80	3,10	1,60	0,20
5,40	3,40	1,50	0,40
5,20	4,10	1,50	0,10

5,50	4,20	1,40	0,20
4,90	3,10	1,50	0,10
5,00	3,20	1,20	0,20
5,50	3,50	1,30	0,20
4,90	3,10	1,50	0,10
4,40	3,00	1,30	0,20
5,00	3,50	1,30	0,30
4,50	2,30	1,30	0,30
4,40	3,20	1,30	0,20
5,00	3,50	1,60	0,60
5,10	3,80	1,60	0,20
4,60	3,20	1,40	0,20
5,30	3,70	1,50	0,20
5,00	3,30	1,40	0,20
7,00	3,20	4,70	1,40
6,40	3,20	4,50	1,50
6,90	3,10	4,90	1,50
6,50	2,80	4,60	1,50
5,70	2,80	4,50	1,30
6,30	3,30	4,70	1,60
4,90	2,40	3,30	1,00
5,20	2,70	3,90	1,40
5,00	2,00	3,50	1,00
5,90	3,00	4,20	1,50
6,00	2,20	4,00	1,00
6,10	2,90	4,70	1,40
5,60	2,90	3,60	1,30
6,70	3,10	4,40	1,40
5,60	3,00	4,50	1,50
5,80	2,70	4,10	1,00
6,20	2,20	4,50	1,50

5,60	2,50	3,90	1,10
5,90	3,20	4,80	1,80
6,10	2,80	4,00	1,30
6,30	2,50	4,90	1,50
6,10	2,80	4,70	1,20
6,40	2,90	4,30	1,30
6,70	3,00	5,00	1,70
6,00	2,90	4,50	1,50
5,70	2,60	3,50	1,00
5,50	2,40	3,80	1,10
5,50	2,40	3,70	1,00
5,80	2,70	3,90	1,20
5,40	3,00	4,50	1,50
6,00	3,40	4,50	1,60
6,70	3,10	4,70	1,50
6,30	2,30	4,40	1,30
5,60	3,00	4,10	1,30
5,50	2,50	4,00	1,30
5,50	2,60	4,40	1,20
6,10	3,00	4,60	1,40
5,80	2,60	4,00	1,20
5,00	2,30	3,30	1,00
5,60	2,70	4,20	1,30
5,70	3,00	4,20	1,20
5,70	2,90	4,20	1,30
6,20	2,90	4,30	1,30
5,10	2,50	3,00	1,10
5,70	2,80	4,10	1,30
6,30	3,30	6,00	2,50
5,80	2,70	5,10	1,90
7,10	3,00	5,90	2,10

6,30	2,90	5,60	1,80
6,50	3,00	5,80	2,20
4,90	2,50	4,50	1,70
7,30	2,90	6,30	1,80
6,70	2,50	5,80	1,80
7,20	3,60	6,10	2,50
6,50	3,20	5,10	2,00
6,40	2,70	5,30	1,90
5,70	2,50	5,00	2,00
5,80	2,80	5,10	2,40
6,40	3,20	5,30	2,30
6,50	3,00	5,50	1,80
7,70	3,80	6,70	2,20
7,70	2,60	6,90	2,30
6,00	2,20	5,00	1,50
6,90	3,20	5,70	2,30
5,60	2,80	4,90	2,00
7,70	2,80	6,70	2,00
6,30	2,70	4,90	1,80
6,70	3,30	5,70	2,10
7,20	3,20	6,00	1,80
6,20	2,80	4,80	1,80
6,10	3,00	4,90	1,80
6,40	2,80	5,60	2,10
7,20	3,00	5,80	1,60
7,40	2,80	6,10	1,90
7,90	3,80	6,40	2,00
6,40	2,80	5,60	2,20
6,30	2,80	5,10	1,50
6,10	2,60	5,60	1,40
7,70	3,00	6,10	2,30

6,30	3,40	5,60	2,40
6,40	3,10	5,50	1,80
6,00	3,00	4,80	1,80
6,90	3,10	5,40	2,10
6,70	3,10	5,60	2,40
6,90	3,10	5,10	2,30
5,80	2,70	5,10	1,90
6,80	3,20	5,90	2,30
6,70	3,00	5,20	2,30
6,30	2,50	5,00	1,90
6,50	3,00	5,20	2,00
6,20	3,40	5,40	2,30
5,90	3,00	5,10	1,80

Differentially private data with ϵ=1 for Iris dataset

sepallength	**sepalwidth**	**petallength**	**petalwidth**
4,75	4,07	-2,22	3,16
0,26	3,89	9,06	4,28
3,96	-1,14	11,67	-0,13
4,38	9,58	28,81	-0,07
-0,26	9,68	5,45	-1,65
14,96	3,56	4,30	2,89
3,97	3,67	-9,67	7,17
2,67	4,56	-0,66	1,53
5,88	-0,84	1,40	-4,90
10,44	0,67	3,12	0,41
5,68	3,97	12,44	4,72
5,86	2,40	0,83	-6,58
4,44	4,19	-6,25	-0,53
0,09	2,26	8,17	1,60
7,23	4,73	-11,42	1,37

-2,90	2,56	4,54	3,09
5,38	6,63	13,55	-1,84
-3,88	2,00	1,44	-0,26
5,24	8,09	-3,26	-2,33
3,55	4,34	2,55	-0,87
6,70	5,74	-0,02	0,15
4,70	1,78	17,44	-1,32
2,82	3,66	18,39	11,14
5,63	3,99	18,48	-0,07
4,37	6,24	-8,21	4,68
4,50	2,90	5,88	-1,39
7,56	9,45	6,89	-1,94
4,05	-0,45	1,50	-12,06
5,70	13,48	-7,08	-3,11
4,99	3,50	9,18	-1,16
13,66	2,56	7,80	-0,14
0,48	5,19	1,58	0,06
2,32	5,97	2,86	-2,15
15,87	1,93	1,29	-1,38
5,00	0,53	2,55	1,86
5,18	2,97	-2,60	-0,76
2,88	2,41	19,90	4,00
4,88	-1,02	-13,18	2,01
6,52	4,50	2,37	8,51
6,88	4,58	13,30	-5,09
6,06	9,77	-1,38	3,08
3,14	-0,35	2,82	8,10
5,73	1,19	14,72	1,62
6,95	2,42	-4,12	0,19
7,46	2,95	-2,64	4,28
5,87	4,12	6,32	1,82

3,28	0,58	6,63	-0,15
6,62	5,58	-2,32	1,29
8,35	0,88	5,65	2,05
4,42	6,87	-5,85	4,24
6,61	3,66	-1,85	-1,20
5,07	-1,37	-1,56	4,67
12,19	7,71	7,23	-3,42
6,77	8,74	3,94	2,38
-0,31	3,23	6,31	2,72
9,11	6,03	26,53	0,94
11,93	2,53	-5,20	3,52
-2,35	4,76	-2,80	7,42
-2,66	4,35	17,89	2,39
5,50	2,57	4,21	2,14
3,79	1,55	8,40	3,03
3,23	4,45	3,34	0,12
8,57	2,60	12,52	2,27
6,89	-1,03	8,62	1,31
7,93	5,95	2,59	0,74
11,12	8,25	19,05	1,20
7,72	-0,15	8,03	4,55
9,37	-0,83	4,96	5,46
15,60	3,10	-26,73	-5,07
18,63	6,34	2,59	1,91
3,40	1,62	7,13	-0,33
-1,56	3,29	8,95	0,62
4,22	2,25	4,39	0,02
5,97	-0,86	3,00	1,66
6,98	3,14	3,37	2,57
5,47	3,75	2,77	1,99
2,74	0,77	6,29	1,84

4,69	5,83	12,34	2,75
2,32	3,76	3,58	4,45
14,89	2,29	8,95	2,81
5,62	2,92	11,80	1,46
6,68	-0,89	11,55	0,49
2,53	2,74	-17,29	3,05
4,13	5,78	-0,54	-1,28
8,82	1,15	3,55	0,65
7,85	3,42	-12,54	3,16
2,32	6,86	2,96	-1,13
2,13	1,23	4,91	3,52
15,09	5,05	-4,23	6,21
-3,01	4,71	13,72	2,99
15,68	-4,20	15,45	1,34
6,75	1,30	14,69	4,64
6,80	1,88	10,68	-0,78
-0,68	1,38	23,30	-1,68
3,80	-2,60	8,44	0,25
6,36	-2,07	7,60	1,26
8,06	0,95	0,73	13,19
6,89	1,95	8,68	1,56
6,60	1,84	1,22	1,89
-0,90	2,52	2,65	-0,72
4,11	1,07	-1,61	1,00
7,13	-1,83	-6,16	4,34
5,95	2,37	3,27	0,41
14,04	4,85	3,83	0,79
10,82	-1,18	2,09	3,26
20,27	2,41	6,35	-0,12
4,90	7,18	23,92	0,43
-3,05	5,05	7,24	0,60

8,02	2,68	15,69	0,53
14,92	2,93	-8,16	1,87
-1,73	3,14	14,70	-0,01
0,65	1,85	8,21	2,40
10,07	-0,27	3,66	1,67
15,44	4,70	3,46	8,74
8,94	6,28	-10,53	-1,84
-0,94	1,25	5,48	1,46
4,67	4,17	2,93	0,52
5,68	2,81	9,47	1,73
3,50	6,58	-18,13	2,21
4,18	1,73	21,32	-0,02
12,43	4,49	-3,27	-4,61
1,96	4,82	9,70	17,70
7,04	6,03	28,28	3,63
6,54	1,12	0,98	4,16
3,35	3,42	-5,47	1,97
4,74	3,03	1,62	1,70
5,49	1,09	13,49	0,19
5,85	8,59	1,13	6,58
13,11	10,13	10,81	0,89
16,17	-3,54	4,54	1,76
9,59	3,49	16,58	6,10
-3,93	-1,51	11,88	2,01
-0,77	4,51	3,78	1,99
-1,01	-2,54	6,23	0,23
6,15	3,10	13,46	1,53

Differentially private data with $\epsilon=2$ for Iris dataset

sepallength	sepalwidth	petallength	petalwidth
4,92	3,79	-0,41	1,68
2,58	3,44	5,23	2,24
4,33	1,03	6,48	0,04
4,49	6,34	15,15	0,07
2,37	6,64	3,42	-0,72
10,18	3,73	3,00	1,65
4,28	3,53	-4,13	3,74
3,84	3,98	0,42	0,87
5,14	1,03	1,40	-2,35
7,67	1,89	2,31	0,26
5,54	3,84	6,97	2,46
5,33	2,90	1,21	-3,19
4,62	3,60	-2,43	-0,21
2,20	2,63	4,64	0,85
6,52	4,36	-5,11	0,78
1,25	3,23	2,92	1,75
5,24	5,06	7,48	-0,77
0,91	2,90	1,57	0,02
5,17	5,94	-0,88	-1,02
4,47	3,87	2,12	-0,33
5,90	4,72	0,74	0,27
4,65	2,69	9,22	-0,56
3,81	3,53	10,15	5,67
5,32	3,50	10,04	0,06
4,78	4,87	-3,36	2,44
4,85	3,15	3,64	-0,59
6,18	6,27	4,25	-0,87
4,73	1,48	1,50	-5,83
5,45	8,79	-2,79	-1,50

5,24	3,85	5,29	-0,48
9,28	2,83	4,65	-0,02
2,74	4,20	1,39	0,13
3,91	4,74	2,08	-0,98
10,39	2,52	1,39	-0,64
4,70	1,76	1,92	1,03
5,09	3,24	-0,65	-0,23
3,69	2,35	10,60	2,15
4,64	1,09	-5,94	1,11
5,76	4,00	1,99	4,56
5,99	4,19	7,45	-2,45
5,33	6,49	0,01	1,64
4,22	1,67	2,16	4,15
5,36	2,25	8,06	0,91
6,98	2,81	0,29	0,79
6,93	3,08	0,93	2,89
6,39	3,61	5,61	1,66
4,89	1,69	5,62	0,68
6,16	4,19	1,09	1,30
7,33	2,09	5,18	1,82
4,66	4,63	-1,27	2,62
5,91	3,18	1,03	0,10
5,04	0,31	0,97	2,84
9,04	5,36	5,72	-0,96
6,38	5,47	3,97	1,69
2,89	3,07	5,51	2,06
7,36	4,46	15,07	1,12
9,31	2,82	-0,40	2,46
1,62	3,88	0,85	4,46
1,57	3,52	10,99	1,70
5,85	2,39	4,36	1,82

4,70	2,02	6,15	2,06
4,56	3,82	4,07	0,96
7,34	2,70	8,26	1,78
6,60	0,73	6,76	1,41
7,01	4,38	3,64	0,97
8,76	5,57	11,67	1,25
7,21	1,42	6,51	3,12
7,69	1,04	4,73	3,48
10,65	2,85	-11,61	-2,04
12,07	4,37	3,19	1,51
4,45	2,01	5,41	0,34
2,12	2,99	6,43	0,91
4,81	2,62	4,45	0,76
5,99	1,27	3,75	1,63
6,84	3,12	4,04	2,03
5,88	3,03	3,58	1,64
4,17	1,88	5,19	1,57
5,10	4,16	8,17	2,02
3,91	3,18	3,99	2,82
10,49	2,64	6,78	2,10
5,71	2,76	7,90	1,33
5,84	0,71	7,42	0,74
4,06	2,72	-6,55	2,18
4,92	4,39	1,83	-0,04
7,26	2,03	3,87	0,98
7,02	3,16	-4,12	2,23
3,71	4,68	2,98	-0,02
3,91	2,02	4,50	2,41
10,69	4,17	0,88	4,35
1,40	3,71	9,41	2,45
11,39	-0,60	10,67	1,72

6,52	2,10	10,14	3,22
6,65	2,44	8,24	0,71
2,11	1,94	13,90	0,01
5,55	0,15	7,37	1,03
6,53	0,22	6,70	1,53
7,63	2,27	3,41	7,84
6,70	2,58	6,89	1,78
6,50	2,27	3,26	1,90
2,40	2,51	3,83	0,64
4,95	1,93	1,74	1,70
6,77	0,69	-0,43	3,32
6,23	2,68	4,38	1,11
10,87	4,32	5,27	1,50
9,26	0,71	4,49	2,78
13,14	2,30	5,67	0,69
5,90	5,19	14,81	1,37
1,28	3,92	6,07	1,30
7,86	2,74	11,20	1,26
10,61	2,81	-1,63	1,84
2,48	3,22	10,20	1,05
3,92	2,52	7,11	2,10
8,13	1,26	4,23	1,73
10,77	3,85	4,18	5,27
7,67	4,54	-2,46	0,13
3,13	2,12	5,64	1,53
6,03	3,48	4,52	1,21
6,79	3,31	7,94	1,86
4,95	4,69	-6,26	2,21
5,24	2,26	13,21	0,74
9,27	3,55	1,16	-1,60
4,83	3,91	7,90	10,00

6,67	4,71	16,94	3,02
6,47	2,11	3,24	2,98
4,67	3,21	-0,34	1,89
5,82	3,06	3,51	1,90
6,09	2,10	9,55	1,29
6,37	5,84	3,11	4,44
9,45	6,41	7,96	1,39
11,48	-0,17	5,22	2,03
8,14	3,25	10,89	4,20
1,19	0,50	8,44	1,95
2,87	3,76	4,49	1,99
2,60	0,43	5,81	1,26
6,03	3,05	9,28	1,66

Differentially private data with ϵ=3 for Iris dataset

sepallength	sepalwidth	petallength	petalwidth
4,98	3,69	0,19	1,19
3,35	3,30	3,95	1,56
4,45	1,75	4,76	0,09
4,53	5,26	10,60	0,11
3,25	5,63	2,75	-0,42
8,59	3,79	2,57	1,23
4,39	3,49	-2,29	2,59
4,22	3,79	0,78	0,64
4,89	1,65	1,40	-1,50
6,75	2,29	2,04	0,20
5,49	3,79	5,15	1,71
5,15	3,07	1,34	-2,06
4,68	3,40	-1,15	-0,11
2,90	2,75	3,46	0,60
6,28	4,24	-3,01	0,59

2,63	3,45	2,38	1,30
5,19	4,54	5,45	-0,41
2,51	3,20	1,61	0,11
5,15	5,23	-0,09	-0,58
4,78	3,71	1,98	-0,16
5,63	4,38	0,99	0,32
4,63	2,99	6,48	-0,31
4,14	3,49	7,40	3,85
5,21	3,33	7,23	0,11
4,92	4,41	-1,74	1,69
4,97	3,23	2,89	-0,33
5,72	5,22	3,36	-0,51
4,95	2,12	1,50	-3,75
5,37	7,23	-1,36	-0,97
5,33	3,97	3,99	-0,25
7,82	2,92	3,60	0,02
3,49	3,86	1,33	0,15
4,44	4,32	1,82	-0,58
8,56	2,71	1,43	-0,39
4,60	2,18	1,72	0,75
5,06	3,32	0,00	-0,05
3,96	2,34	7,50	1,53
4,56	1,79	-3,53	0,80
5,51	3,83	1,86	3,24
5,69	4,06	5,50	-1,56
5,09	5,39	0,47	1,16
4,58	2,35	1,94	2,83
5,24	2,60	5,84	0,67
6,98	2,94	1,76	1,00
6,75	3,12	2,12	2,43
6,56	3,44	5,37	1,61

5,43	2,06	5,28	0,95
6,01	3,73	2,23	1,30
6,98	2,49	5,02	1,75
4,74	3,89	0,25	2,08
5,67	3,02	1,98	0,53
5,02	0,88	1,81	2,22
8,00	4,57	5,21	-0,14
6,26	4,38	3,98	1,46
3,96	3,01	5,24	1,84
6,77	3,94	11,24	1,18
8,44	2,91	1,20	2,11
2,95	3,59	2,07	3,47
2,98	3,25	8,70	1,46
5,97	2,32	4,40	1,71
5,00	2,18	5,40	1,74
5,01	3,62	4,31	1,24
6,92	2,73	6,84	1,62
6,50	1,32	6,14	1,44
6,71	3,85	4,00	1,05
7,97	4,68	9,22	1,27
7,04	1,95	6,01	2,65
7,12	1,66	4,65	2,82
9,00	2,77	-6,58	-1,02
9,88	3,71	3,40	1,37
4,80	2,14	4,84	0,56
3,35	2,90	5,58	1,01
5,01	2,75	4,46	1,01
5,99	1,98	4,00	1,62
6,79	3,11	4,26	1,86
6,02	2,78	3,86	1,53
4,65	2,26	4,83	1,48

5,23	3,61	6,78	1,78
4,44	2,99	4,13	2,28
9,03	2,76	6,05	1,87
5,74	2,71	6,60	1,29
5,56	1,24	6,05	0,83
4,58	2,71	-2,96	1,88
5,18	3,93	2,62	0,37
6,74	2,32	3,98	1,08
6,75	3,07	-1,31	1,92
4,17	3,95	2,99	0,36
4,51	2,28	4,37	2,04
9,23	3,88	2,59	3,74
2,86	3,37	7,97	2,26
9,96	0,60	9,08	1,85
6,45	2,37	8,63	2,75
6,60	2,63	7,43	1,21
3,04	2,13	10,77	0,57
6,13	1,07	7,01	1,28
6,59	0,98	6,40	1,62
7,49	2,72	4,31	6,06
6,63	2,78	6,29	1,85
6,47	2,41	3,94	1,90
3,50	2,51	4,22	1,09
5,24	2,22	2,86	1,93
6,64	1,52	1,48	2,98
6,32	2,79	4,76	1,34
9,81	4,15	5,74	1,73
8,74	1,34	5,30	2,62
10,76	2,27	5,45	0,96
6,23	4,53	11,77	1,68
2,72	3,55	5,68	1,53

7,81	2,76	9,70	1,51
9,17	2,78	0,55	1,82
3,89	3,25	8,70	1,40
5,02	2,75	6,74	2,00
7,49	1,78	4,42	1,76
9,21	3,57	4,42	4,11
7,25	3,96	0,22	0,79
4,49	2,42	5,69	1,55
6,49	3,26	5,04	1,44
7,16	3,47	7,42	1,91
5,43	4,06	-2,31	2,20
5,59	2,44	10,51	0,99
8,21	3,23	2,64	-0,60
5,79	3,61	7,30	7,43
6,55	4,28	13,16	2,81
6,45	2,44	3,99	2,59
5,12	3,14	1,38	1,86
6,18	3,08	4,14	1,97
6,30	2,43	8,23	1,66
6,55	4,93	3,78	3,73
8,24	5,18	7,00	1,56
9,92	0,95	5,45	2,12
7,66	3,16	8,99	3,57
2,89	1,16	7,29	1,94
4,08	3,50	4,73	2,00
3,80	1,42	5,68	1,61
5,98	3,03	7,89	1,71