DIFFERENTIALLY PRIVATE ATTRIBUTE SELECTION FOR CLASSIFICATION

ESRA VAR
B.S., Computer Engineering, BAŞKENT UNIVERSITY, 2010

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

IŞIK UNIVERSITY
2015

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DIFFERENTIALLY PRIVATE ATTRIBUTE SELECTION FOR CLASSIFICATION

ESRA VAR

APPROVED BY:

Assist. Prof. Dr. Ali İnan (Thesis Supervisor)	Işık University	_____
Assoc. Prof. Dr. Olcay Taner Yıldız	Işık University	_____
Assist. Prof. Dr. Gülay Ünel	Işık University	_____

APPROVAL DATE: ..../..../....

Differentially Private Attribute Selection For Classification

Abstract

Any study on processing or analyzing large data sets that contain personally sensitive data should conform against some form of privacy protection mechanism. Otherwise, malicious people can aceess these data sets to extract private information and use this private information in agency operations, blackmail, fraud or any other harmful actions. Importance and necessity of privacy preserving data mining is increasing day by day, hence public and government lawmakers, privacy advocates and the media are drawing more and more attention to this subject daily. This thesis proposes an approach to that selects features from a data set according to the differential privacy mechanism and implements this proposed solution on a popular data mining library called WEKA.

Ayrımsal Mahremiyete Dayalı Öznitelik Seçimi

Özet

Büyük veriler üzerindeki çalışmalar ve analizler gizliliği, özellikle kişisel hassas bilgilerin gizliliğini gözetmek durumundadır. Gerekli koruma önlemleri alınmazsa kötü niyetli kişiler kritik bilgilere ulaşabilir ve bunları şantaj, dolandırıcılık gibi çeşitli zararlı amaçlı için kullanabilir. Veri güvenliği kavramının önemi ve gerekliliği günden güne artmaktadır ve halk, hükümet yetkilileri ve medya bu kavrama giderek artan bir ilgi göstermektedir. Bu tez yaygın kullanılan bir veri madenciliği kütüphanesi olan WEKA üzerinde, ayrımsal mahremiyet kavramını veri madenciliğinin bir alanı olan özellik seçimi yönünden ele alıp veri güvenliği performansını geliştiren bir yaklaşım sunmaktadır.

Acknowledgements

Many thanks to all whose research and mental or academic support helped me during this thesis. I am very grateful to my supervisor Ali İnan for listening to and answering my questions patiently even when he was busy with his personal duties or at peak times at the university. It was a great chance for me to work with him and benefit from his experience. Although they were not near me physically, I want to thank my family for their support and encouragement. And finally I want to thank my fiance who supported me in my busy and troubled times.

To my family...

Table of Contents

List of Figures

List of Tables

List of Abbreviations

VEIGNoiseT	**V**arying **E**psilon **I**nfo **G**ain Noise **T**rue
VEIGNoiseF	**V**arying **E**psilon **I**nfo **G**ain Noise **F**alse
VECSNoiseT	**V**arying **E**psilon **C**hi **S**quare Noise **T**rue
VECSNoiseF	**V**arying **E**psilon **C**hi **S**quare Noise **F**alse
VIIGNoiseT	**V**arying **I**nstance **I**nfo **G**ain Noise **T**rue
VIIGNoiseF	**V**arying **I**nstance **I**nfo **G**ain Noise **F**alse
VICSNoiseT	**V**arying **I**nstance **C**hi **S**quare Noise **T**rue
VICSNoiseF	**V**arying **I**nstance **C**hi **S**quare Noise **F**alse
VAIGNoiseT	**V**arying **A**ttribute **I**nfo **G**ain Noise **T**rue
VAIGNoiseF	**V**arying **A**ttribute **I**nfo **G**ain Noise **F**alse
VACSNoiseT	**V**arying **A**ttribute **C**hi **S**quare Noise **T**rue
VACSNoiseF	**V**arying **A**ttribute **C**hi **S**quare Noise **F**alse

Symbols

ϵ	Parameter of Sensivity
Δ	Sensivity
λ	Variable of Sensivity
D	Dataset D
S^D	Sensitivity of Dataset D
Q	Query Set
r_k	kth Row of Dataset
κ	A Randomized Function
p	Probability
E	Entropy
C_k	kth Class
#	Number
Pct.	Percentage

Chapter 1

Introduction

Since the beginning of 21th century, people have been using the Internet for their daily work such as shopping, credit card transactions, phone calls, banking actions, chatting on social networking websites. In addition to these daily routine data there are other areas that accumulate huge amounts of data such as satellite observation data, health-care records, government organizations and databases of companies.

Data is not stored only for commercial purposes but also by governments. An example of these are census data sets to that are used to increase citizen participation, collaboration, and transparency in government, which the state has to offer to the public. Census data sets hold real statistical data about given populations. Many firms take the collected data when they construct their sales methods and plans.

Taking all of this effort of data collection into consideration, we arrive at some important questions: how much our personal information is stored somewhere in a database and how much of this information is kept safely? First question can not be easily answered because it is hard to draw any boundaries. Although there are laws towards protecting personal information, people are in doubt about enforcement of law. Furthermore, there is a huge amount of data. So, it is not easy to check and control every piece of it. As for the second question, we need to consider a relatively new area, namely data security and privacy.

Data security means protection of data from unauthorized users. Data protection can be done in many different ways. Traditional protection techniques are sanitization approaches such as k-anonymity, l-diversity and t-closeness. These sanitization techniques generally try to break the link between the records and their owners. First used method among all of them is k-anonymity. K-anonymity approach aims that each record relates to at least k individual records for providing privacy [1]. Because of the inadequacy of k-anonymity, l-diversity was proposed. L-diversity provides privacy by supplying diversity on sensitive attributes of anonymity groups [2]. T-closeness is a further refinement of l-diversity which also includes maintaining the distribution of sensitive fields. It is assumed that sharing sanitized data with third parties does not harm privacy but sanitization is never perfect. This is proven in Dwork's first paper on differential privacy [3].

Second protection method is secure multi-party computation (SMC). In this technique, data is never shared explicitly. Any function of the collected data should be computed through cryptographic solutions as in [4] and [5]. These methods employ SMC protocols that require heavy use of computational power and network bandwidth.

Current state of the art in privacy preserving data mining is a new solution called differential privacy. In this approach, access to a data set is restricted to a statistical database interface. Individual privacy is protected by adding noise to query results. We explain this approach in detail in Chapter 2.

In this thesis, we try to solve the following important problem. Database D contains private data and agent A needs to build a classifier C on D without violating the privacy of the individuals whose records are stored in D. Classifier C might be used for any purpose. C could be a method of deciding whether there is an epidemic outbreak, or whether a banking customer will default on a loan to be given by a private company.

There are many studies on how C can be built on D by A in a privacy preserving manner. However, to the best of our knowledge, there is no solution that discusses how attributes of D can be selected - in a differentially private manner - before C is built.

We list our contributions below:

- We propose a differentially private method of attribute selection for classification.

- We empirically analyze the performance of this solution on various real-life data sets.
- We implement our solutions in WEKA [6].

In this thesis, our study is based on the differential privacy approach. In Chapter 2 and Chapter 3, we give general information about differential privacy and feature selection which are the techniques we use. In Chapter 4, we explain main results of our study. Experimental results of the study are presented and discussed in Chapter 5. Chapter 6 compares our work against existing studies. Finally, in Chapter 7, we sum up our discussion and present alternatives for future work.

Chapter 2

Differential Privacy Overview

In recent years, differential privacy has emerged as one of the best techniques for protecting the privacy of individuals. Differential privacy is applied before publishing the results of aggregate, statistical queries on a data set. While techniques like k-anonymity focus on overall data privacy, differential privacy is interested in the privacy of individuals. It is not affected by adding or removing data, and it still protects privacy of individuals.

In privacy techniques, data sanitization branch on two different types which are interactive and non-interactive. In non-interactive process, the privacy of related data is provided by various methods. People, who want to reach those data, study on altered data instead of the original data. In active process, there is a middle-ware between data and user. Although user studies on the original data, middle-ware sends perturbed results to the user. Differential privacy is an active privacy technique and also works as a noise generator.

Definition 2.1. [3] *A randomized function κ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\kappa)$,*

$$Pr[\kappa(D_1) \in S] \leq e^{\epsilon} \times Pr[\kappa(D_2) \in S] \tag{2.1}$$

Mechanism κ should give the same result on any two data sets that differ in only one record with probability e^{ϵ}. This means, any attack on an individual would have been successful even if his/her data were not included in the data set.

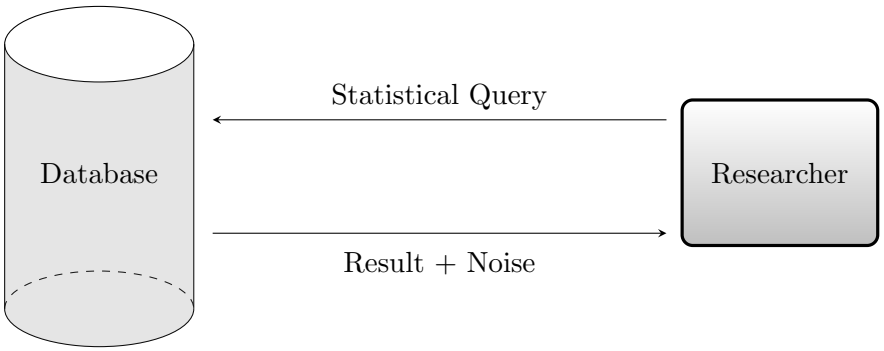


Figure 2.1: Differential privacy

A statistical query cannot be used to directly reach the contents of the tables. Instead, with statistical queries, one can only obtain aggregate results like `"select count(*)"`. Other types of queries will be disregarded.

Based on these pieces of information D_1 and D_2 sibling data sets with one individual difference, differential privacy guarantees that the distance between results of a query is less than e^{ε} for two data sets and also ε is adjustable. Assigning a small value to ε causes a slight difference in query results for that data sets. This ensures that malicious people will be unable to predict whether the result is derived from D_1 or D_2. This makes malicious people confused to extract private information via queries. Thus, differential privacy provides privacy by masking presence or absence of the record.

Figure 2.1 shows that noise is added to query results to mask real results before a user receives them. Noise amount changes with respected to the desired level of privacy. As noise gets larger, privacy protection gets stronger and results will be less predictable. Noise amount changes with the sensitivity rate of the query results.

Most common usage of differential privacy is adding random noise with Laplace distribution to the query results.

2.1 Sensitivity

Sensitivity is the largest difference between query results of any two sibling data sets. In other words, this is the maximum amount, over the domain of f, that any single argument to f, that is, any single row in the database, can change the output [7].

Definition 2.2. [7] *For query set* $Q = \{Q_1, Q_2, \ldots, Q_n\}$ *let* Δ *denote the sensitivity of* Q *such that:*

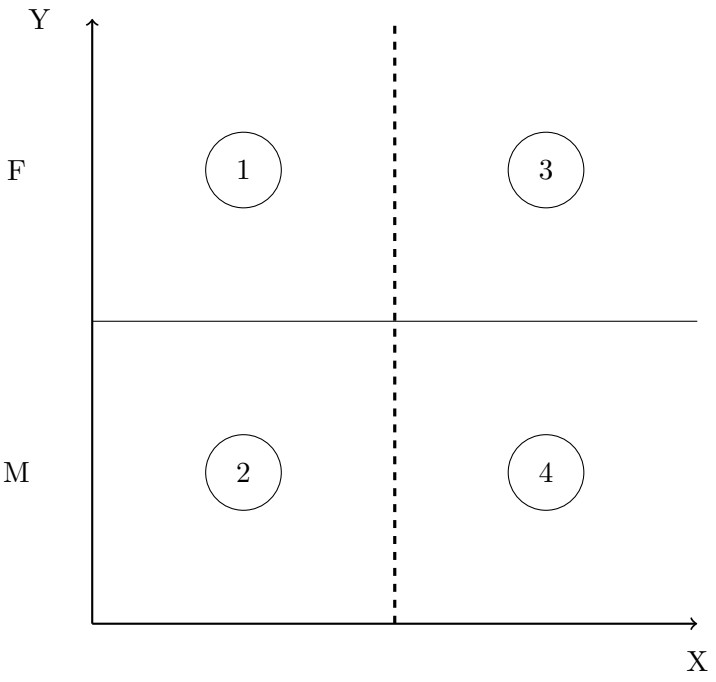
$$\Delta = \forall max \| S^D(Q) - S^{D'}(Q) \|_1 \tag{2.2}$$

If we consider all query sets, this formula will become:

$$\Delta = \sum_{i=1}^{n} \| S^D(Q_i) - S^{D'}(Q_i) \|_1 \tag{2.3}$$

For example,

$$Q = \begin{cases} \texttt{SELECT COUNT(*) FROM T WHERE X = 'M'}, \\ \texttt{SELECT COUNT(*) FROM T WHERE Y > 40 }, \end{cases} \tag{2.4}$$



Consider a data set $D = \{r_1, r_2, \ldots, r_k\}$ and consider a sibling D' of D: $D' = \{r_1, r_2, \ldots, r'_k\}$ There will be 16 different combinations considering the single record on which D and D' differ.

- $r_k \in 1$ & $r_k{}' \in 1$, same result for Q1 and Q2
- $r_k \in 1$ & $r_k{}' \in 2$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 1$ & $r_k{}' \in 3$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 1$ & $r_k{}' \in 4$, $Q_1^{D'} = Q_1^D + 2$
- $r_k \in 2$ & $r_k{}' \in 1$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 2$ & $r_k{}' \in 2$, same result for Q1 and Q2
- $r_k \in 2$ & $r_k{}' \in 3$, $Q_1^{D'} = Q_1^D + 2$
- $r_k \in 2$ & $r_k{}' \in 4$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 3$ & $r_k{}' \in 1$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 3$ & $r_k{}' \in 2$, $Q_1^{D'} = Q_1^D + 2$
- $r_k \in 3$ & $r_k{}' \in 3$, same result for Q1 and Q2
- $r_k \in 3$ & $r_k{}' \in 4$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 4$ & $r_k{}' \in 1$, $Q_1^{D'} = Q_1^D + 2$
- $r_k \in 4$ & $r_k{}' \in 2$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 4$ & $r_k{}' \in 3$, $Q_1^{D'} = Q_1^D + 1$
- $r_k \in 4$ & $r_k{}' \in 4$, same result for Q1 and Q2

Within all of these combinations, maximum difference is obtained by $r_k \in 1$ & $r_k{}' \in 4$ and $r_k \in 2$ & $r_k{}' \in 3$ or $r_k \in 4$ & $r_k{}' \in 1$ and $r_k \in 3$ & $r_k{}' \in 2$. So, the difference of query results will be at most 2. Therefore, sensitivity is also 2 in this example.

Notice that Δ is independent of the data set in use. Δ is defined on all pairs of sibling data sets. This means, Δ is a function of the query set and not the underlying data set. This shows us that differential privacy is a strong protection mechanism, because it works over the entire universe of all data sets based on the semantics of the queries issued to the database.

Sensitivity computation is central in the application of differential privacy. Unfortunately, for a given query set, computing the sensitivity is a proven to be NP-hard [8].

Chapter 3

Feature Selection

Feature selection is the process to select the best subset of features towards building a classifier on a data set. Using all of the features can often be an unnecessary approach because the data set has lots of features and some of them may be coding the same information redundantly and some of them may be irrelevant for the classification task. Data miner should pick the most useful features to decrease the execution time and maximize classification accuracy.

Another motivation for feature selection is that we want to create the smallest model which ignores input features that has little effect on our output. By ignoring unnecessary features we decrease our model size.

Feature selection process can be done with three different approaches. Those approaches are filter method, wrapper method and embedded method.

3.1 Filter Approach

Filter approach does not handle the effect of the selected feature subset on the induction algorithm. Also in this approach, a statistical measure is applied for assigning a score to each feature. The features are ranked by their scores. Features should be removed are chosen based on this ranking.

Filter methods are interested in features that are independent of each other. A few examples of scoring functions are the chi-squared test, information gain and correlation coefficient scores.

3.2 Wrapper Approach

Unlike the filter method, wrapper method associates the induction algorithm with the feature selection step. In wrapper method [9], the induction algorithm is used as a "black box" by the subset selection algorithm. Wrapper approach handles selection of a subset of features as a search problem. Almost every possible combination is evaluated and compared with others. For example, recursive feature elimination algorithm is a commonly used wrapper method.

3.3 Embedded Approach

In embedded approach, similarly to wrapper methods, feature selection is linked to the classification stage. This link, which is being in this case much stronger as the feature selection in embedded methods, is included into the classifier construction[10]. Example of these approaches can be LASSO, Elastic Net and Ridge Regression.

After feature selection, we have to check convenience by controlling the accuracy of the resulting model. We divide our data set into test and train data sets to make this control and if we want to better approximate real results, the data set is divided into more parts and a stratified cross validation is employed.

3.4 Cross Validation

The data set is divided into n parts with n-fold cross validation. $n-1$ parts are used for creating our model for training. This model is used for estimating the accuracy on an arbitrary data set using the part that is reserved for testing. This process continues n times and accuracy rate of model is calculated by averaging the n accuracy measurements.

A common value for n of n-fold cross validation is 10. Below, we show how cross validation is applied when $n = 3$:

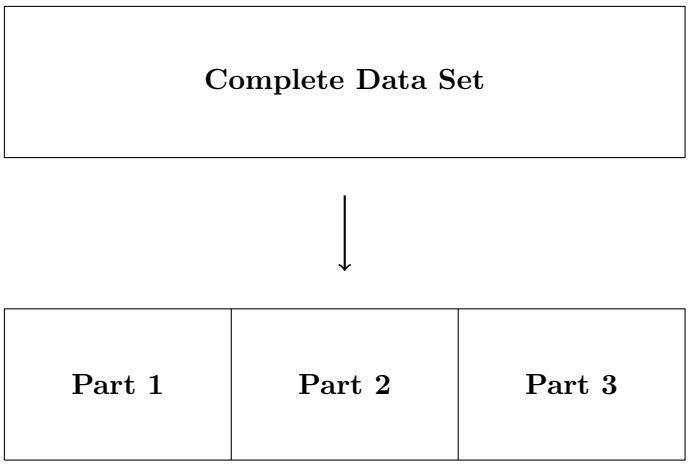


Figure 3.1: First step of cross validation

We divide our data set into three parts as shown in Figure 3.1. For it contains three parts it is named 3-fold cross validation.

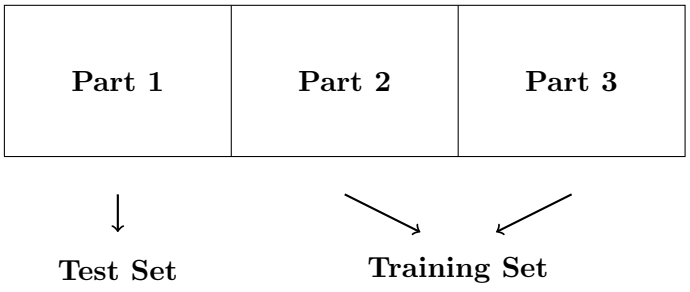


Figure 3.2: Second step of the cross validation

We select one of these parts for testing and other parts will be used for training. This is shown in Figure 3.2.

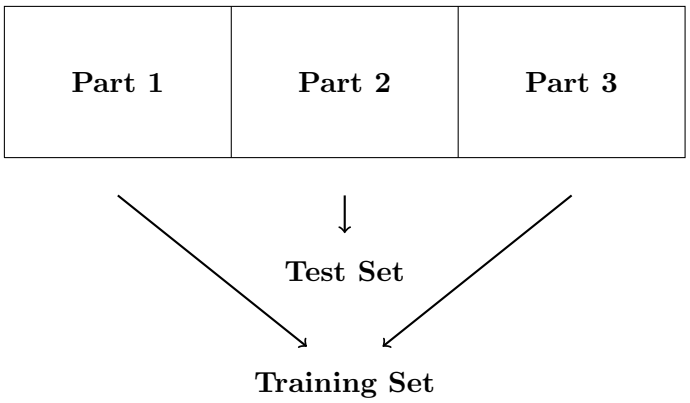


Figure 3.3: Third step of the cross validation

After executing the previous partition, we change our selection about test and train sets as shown in Figure 3.3 and Figure 3.4.

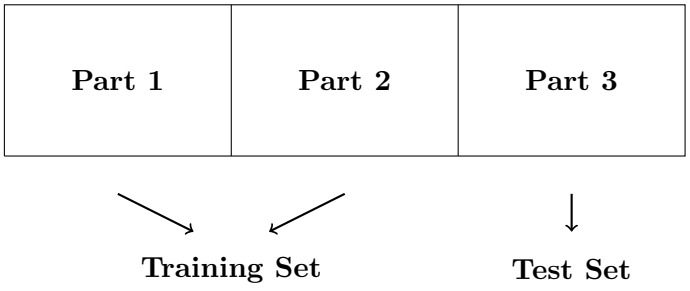


Figure 3.4: Fourth step of the cross validation

At the end of this process, we examine the performance with 3 different training data sets 3 times. Every time test and training sets have different element. General accuracy and error rate of the model is calculated as the average of each separate steps' result.

Chapter 4

Proposed Solution

Consider a data set that is protected by differential privacy. Any access to this data set is restricted to statistical queries, responses to which will be perturbed based on the sensitivity of the query set. We attack exactly the problem of selecting a relevant set of features from this data set without violating differential privacy. The problem of building a classifier according to differential privacy is left out. On this aspect of the problem we refer readers to existing studies covered in Chapter 6.

In Section 4.1, we discuss the pre-processing steps applied on our data sets used in our experimental analysis. In Section 4.2, we explain the steps of selecting an appropriate attribute evaluator and search method. In Section 4.3, we analyze the sensitivity of our attribute selection methods. Section 4.4 explains the filtering step and finally in Section 4.5, we explain how we apply classification.

4.1 Preparing the Data Set

In this process, we arrange our data set for preparing it to run our code. The data set has missing or out of range values. There are many ways to handle missing data. We choose removing records with missing values because there is a small amount of records with missing values in our data sets. Alternative techniques for handling missing values are imputation, partial imputation, partial deletion, full analysis or interpolation.

We use list-wise deletion method which is a subtopic of partial deletion method. In list-wise deletion, an entire record is excluded from analysis if any single value of the record is missing [11]. Here is an example to explain list-wise deletion on our mushroom data set:

veil-color	ring-no	ring-type	spore-print-color	population	habitat	class
w	t	p	r	v	m	p
w	t	e	w	c	w	e
w	o	?	h	v	g	p
w	t	e	w	c	w	e
w	o	l	h	y	p	p
w	o	p	h	v	?	p
w	o	l	h	v	d	p
w	o	l	h	v	d	p

Table 4.1: Before applying list-wise deletion

In the mushroom data set there are 22 attributes and one class attribute. We choose a few of these attributes to show list-wise deletion execution of our subject. Class attribute shows which mushroom is edible or poisonous in this data set and in the example, every attribute is represented with the first letter of domain value. Using list-wise deletion, we remove rows 3 and 6 from the sample before performing any further analysis.

veil-color	ring-no	ring-type	spore-print-color	population	habitat	class
w	t	p	r	v	m	p
w	t	e	w	c	w	e
w	t	e	w	c	w	e
w	o	l	h	y	p	p
w	o	l	h	v	d	p
w	o	l	h	v	d	p

Table 4.2: After applying listwise deletion

4.2 Attribute Selection

In the WEKA library, there are two steps of attribute/feature selection:

- Selecting appropriate attribute evaluator, and
- Selecting appropriate search method.

4.2.1 Selecting an Attribute Evaluator

In WEKA, for making attribute selection we have to select an attribute evaluator which computes an individual feature's necessity and importance for that data set. There are 15 different attribute/subset evaluators. We choose ChiSquaredAttributeEval and InfoGainAttributeEval attribute evaluators because these methods can be easily translated to a query set over a statistical database model and their sensitivity level is lower compared to more complex attribute selection methods.

4.2.1.1 ChiSquaredAttributeEval

A chi-squared test aims at analyzing categorical data. Before applying this test, a contingency table has to be built. This contingency table contains count values of each category across distinct class values. Therefore, data should be counted and divided into categories at the first step.

Chi-squared test does not work with parametric or continuous attributes. An example of a continuous attribute can be weight of a person or temperature of the weather. Even though temperature of the weather would not be appropriate for a Chi-square test, arranging the weather into categories as "rainy" and "sunny" would be a sufficient transformation. This process is called discretization.

An example contingency table from the mushroom data set is given below:

	Poison	Edible	Total
White	465	1035	1500
Brown	680	320	1000
Total	1145	1355	2500

Table 4.3: Test table on mushroom dataset

The estimated value for each cell is calculated by multiplication of the column total value and sum of row values and dividing this result with total value of the table. Therefore, for our test table the expected value of white poisonous mushroom is $(1500 * 1145)/2500$ or 687 and test table with expected values will be like this:

	Poison	Edible	Total
White	465 (687)	1035 (813)	1500
Brown	680 (458)	320 (542)	1000
Total	1145	1355	2500

Table 4.4: Test table on mushroom dataset with expected values

4.2.1.2 InfoGainAttributeEval

Before explaining the information gain attribute evaluator, we need to discuss the subject of attribute entropy. Entropy characterizes the (im)purity of an arbitrary collection of examples [12].

$$E(S) = -\sum_{i=1}^{n} p_i \log_2 (p_i) \tag{4.1}$$

Entropy is based on the information theory. A large decrease in entropy means larger information gain. After computing the total entropy of data sets partitioned on each attribute, we can find how much information can be gained upon partitioning a data set on an attributes. While selecting an attribute A, our information gain is calculated with this formula:

$$Gain(A) \quad = E(p, n) - E(A) \tag{4.2}$$

To select an attribute using information gain, $Gain(A)$ is computed for each attribute A and the one that provides the highest gain in information content is selected. In the formula above, $E(p, n)$ represents total information gain of class.

4.2.2 Selecting Search Method

There are 10 search algorithms in the WEKA library. We use the ranker search algorithm. Ranker search ranks attributes by their individual evaluations. As discussed in Chapter 5, we will select the best n-features of a data set, where n will be an experiment parameter.

Ranker search computes the score of each attribute only once and then lists all attributes in increasing order of their scores. This search method is more suitable to our problem setup due to its rather low sensitivity compared to other search algorithms.

When building our solution, we have also considered other search algorithms like subset evaluation. However, sensitivity of these algorithms were computed to be prohibitive. In the next section, we present our analysis for the ranker algorithm. Any work of ours on subset evaluation algorithm are omitted.

4.3 Adding Noise

We choose Laplace distribution which produces a random noise depending on a given seed and lambda value. In Chapter 3, we explained noise, noise types, Laplace noise and formulas.

As previously mentioned, the amount of added noise depends on the sensitivity of the issued query set. Once sensitivity is available, the noise is sampled using Laplace distribution and $\lambda = \frac{\Delta}{\epsilon}$ is decided as the magnitude of the distribution for ϵ-differential privacy.

In Chapter 3, we also defined sensitivity showed how it is calculated on an example. In our study, we extract a formulation for generalizing sensitivity calculation for all data sets. To give an another example for two attributes A_1 and A_2;

A_1

	Y	N
Y	a	c
N	b	d

A_2

	Y	N
Y	e	f
N	g	h

Figure 4.1: Contingency tables of A_1 and A_2

Suppose each cell is queried by itself.

$$Q = \begin{cases} \texttt{Select count(*) from T group by Class, A1} \\ \texttt{Select count(*) from T group by Class, A2} \end{cases} \tag{4.3}$$

In this example maximum difference for A_1, A_2, and class attribute *Class* will be obtained by changing (Y, Y, N) to (N, N, P). This is because every attribute has 2

sensitivity here and there are two attributes so total sensitivity is sum of their sensitivity values and this is 4.

If we generalize this calculation our formulation will be like below:

$$\Delta = (\text{Number Of Attributes} - 1) \times 2 \tag{4.4}$$

Minus 1 difference above is for eliminating the class attribute from this calculation.

4.4 Filtering

There are lots of filters in the WEKA library under the subcategories supervised and unsupervised. We choose the remove filter to get rid of attributes that are not selected. Once selected attributes are chosen based on infogain and chisquare, we remove any attributes that are not selected using the unsupervised remove filter of WEKA. This filter takes as input, the list of attributes to be removed and the sensitivity of this operation is 0.

4.5 Classification

Classification has two main steps. The first step is creating a model based on labeled data. This step is also called the learning step. In the second step, the model built is tested over the remaining labeled data and if the predicted accuracy of the model is at an acceptable level, the model is adapted.

To give an example about classification, some decision making processes in the health care domain can be considered. At the end of test results obtained from patient records, we can decide whether a new person is sick or not based on her/his tests results. There are several techniques that can be used for classification such as decision tree induction, neural networks and Bayesian methods.

Our classification method is the Naive Bayes Classifier method. There is no specific reason to select this classifier, all of our experiments can also be run with other classifiers. We did not vary classification methods because of our focus in this work is to capture effect of attribute selection on classification accuracy.

4.5.1 Naive Bayes Classifier

Naive bayes is a probabilistic classifier based on the Bayes theorem. In Naive Bayes classifier, all features are considered to be independent of the value of any other features. Bayes theorem is formulated below:

$$p(A|B) = \frac{p(A)p(B|A)}{p(B)} \tag{4.5}$$

$P(A|B)$: conditional probability of A given B.
$P(B|A)$:conditional probability of B given A.
$P(x)$: prior probability of event x.

Naive Bayes decides on the class value using this probabilistic calculations on data set. In the first step, system learns and models presented data. In the second step, accuracy of the model is tested using the test data set.

For a data set with n features Naive Bayes formulation is given below:

$$p(C_k|x) \times p(x) = p(C_k) \times p(x|C_k) \tag{4.6}$$

In Naive Bayes formulation x represents features vector between 1 to n $x = x_1, x_2, \ldots, x_n$. If p(x) represented as a feature vector:

$$p(x) = \sum_{i=1}^{L} p(x|C_k)p(C_k) \tag{4.7}$$

Class of new data is decided by probabiliy's results. Data is classified as the biggest result's class label.

Chapter 5

Experimental Results

In this chapter, we analyze our study processed on three different data sets which have different number of instances and attributes and then we empirically analyze our solution based on results from those different data sets. In Table 5.1, we describe the data sets we used in our experiments. We have picked one data set that has a large number of records (adult), one that has a large number of attributes (mushroom) and one that contains a hard-to-learn classification model.

Data set Name	# of Attributes	# of Instances	# of Classes	NB Error Rate
Mushroom	22	8124	2	2.71
Nursery	8	12960	2	9.03
Adult	14	48841	2	16.98

Table 5.1: Dataset descriptions

There are 3 important factors that may affect the performance of our solution. These are:

i Privacy parameter epsilon,

ii Number of instances,

iii Number of attributes to be selected.

Naturally, we evaluate the performance of our solution based on error rates of classification. We do not analyze the complexity of our solution because adding noise to counts that already are generated is minimal, especially in large data sets.

Default values of the parameters for each data set are listed below:

Data set Name	Epsilon	Pct. of Instance	Attribute
Mushroom	0.3	100%	11
Nursery	0.15	100%	4
Adult	0.05	100%	7

Table 5.2: Default parameters

In what follows, we will look at the effects of each of these parameters one by one. We start with privacy parameter epsilon, since its default value was set during these set of experiments. For each data set, default value of epsilon is decided through a search mechanism.

We plot our results for each data set separately. Each section below contains 3 figures for our 3 data sets and separate discussions.

5.1 Effects of Changing Epsilon

As below, there are graphs which show effects of varying epsilon value on our data sets, features of which were represented before.

Figure 5.1 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing epsilon values. Noisy and clear data of mushroom data set are used for this experiment. As we can see, noise effects are more visible in the beginning. The reason is our noise formulation, $\lambda = \frac{\Delta}{\epsilon}$. Small epsilon value causes big lambda value. So it means bigger noise amount. While epsilon value is increasing, noise effect disappears, noisy lines get closer to the noiseless and Naive Bayes lines that the prove formulation again.

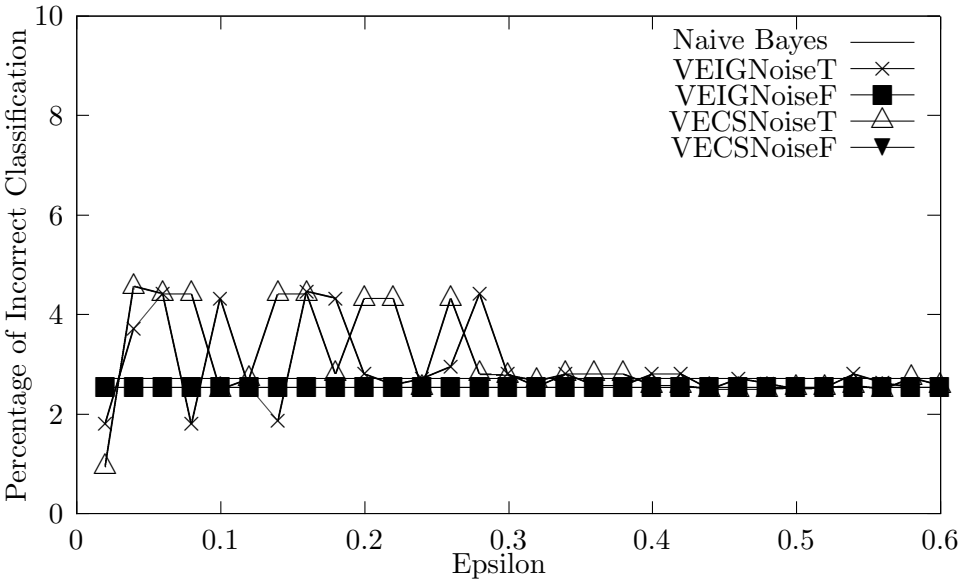


Figure 5.1: Varying epsilon value graph on mushroom data set

Figure 5.2 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing epsilon values on the nursery data set. Noise deviations are visible up to 0.15 value but on continued epsilon values after 0.15, epsilon value increases, noisy lines join to the noiseless and Naive Bayes lines. This form of graph is also similar to the mushroom data set on varying epsilon values.

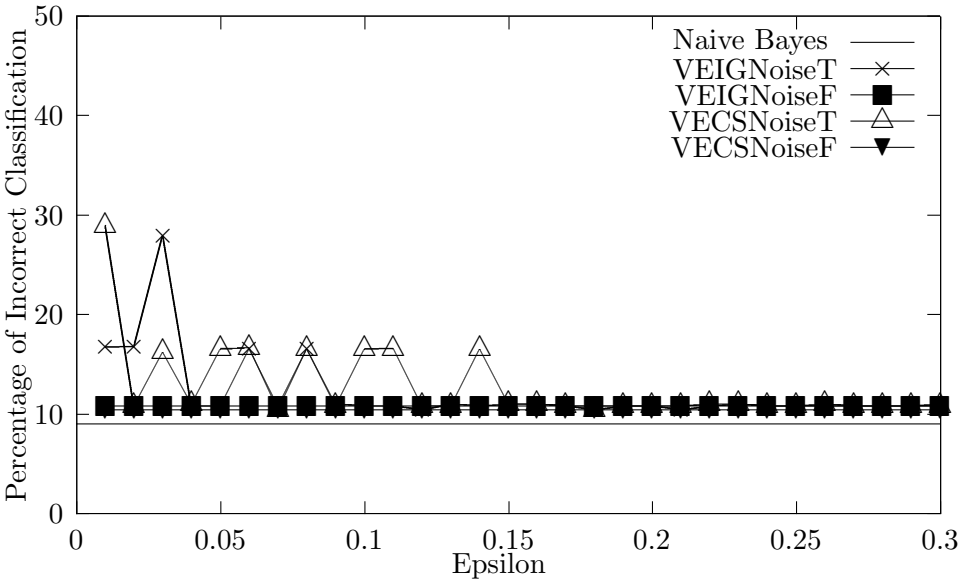


Figure 5.2: Varying epsilon value graph on nursery dataset

Figure 5.3 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing epsilon values. In this graph we can see that although the noiseless results have straight line and beside Naive Bayes error rate, noisy results have zigzag structure between 0 to near 0.5 and effects of noise are active in this interval.

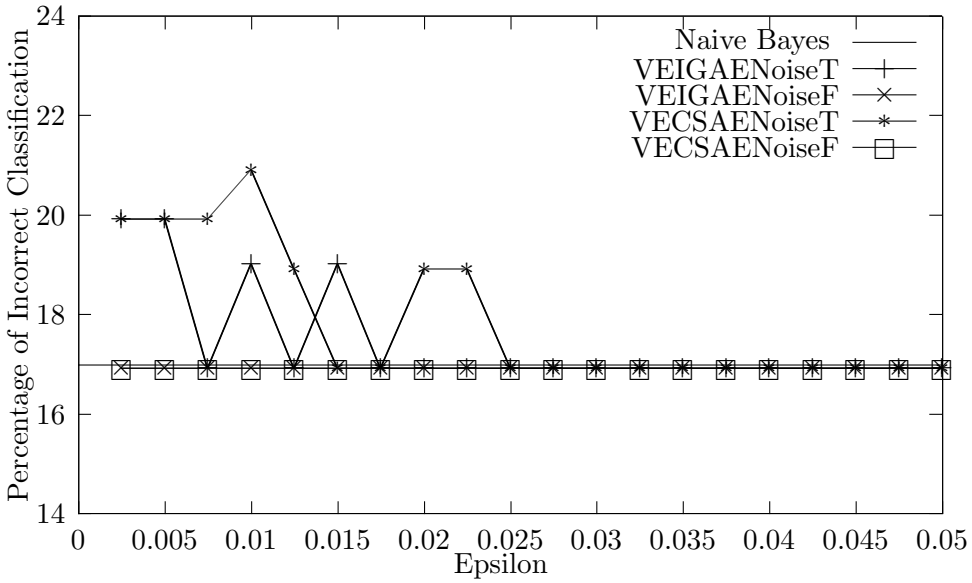


Figure 5.3: Varying epsilon value graph on adult dataset

5.2 Effects of Changing Instance

Figure 5.4 shows effect on error rate for classification of changing instances with different subset evaluators, which are chisquared and infogain, applied to perturbed and noiseless data of the mushroom data set taken from UCI[13].

As we see that noisy situations lead a parallel path to results of noiseless chisquare and infogain evaluators but there is a small difference value of error rate between them. Error rate of noisy results both provide us to find out expected effect with tracing the noiseless results in a similar form with little diversion and finally gradually approaching to it.

In every steps, path of noisy lines are connected with path of noiseless lines in many points and changing interval is fairly low. As instance percentage increases, all results error is decreasing and come close to Naive Bayes result. This behaviour is expected because making noise on data sets cannot create same effect while number of instance is increasing. Increasing the number of instances increases stability of classification because of increasing test data.

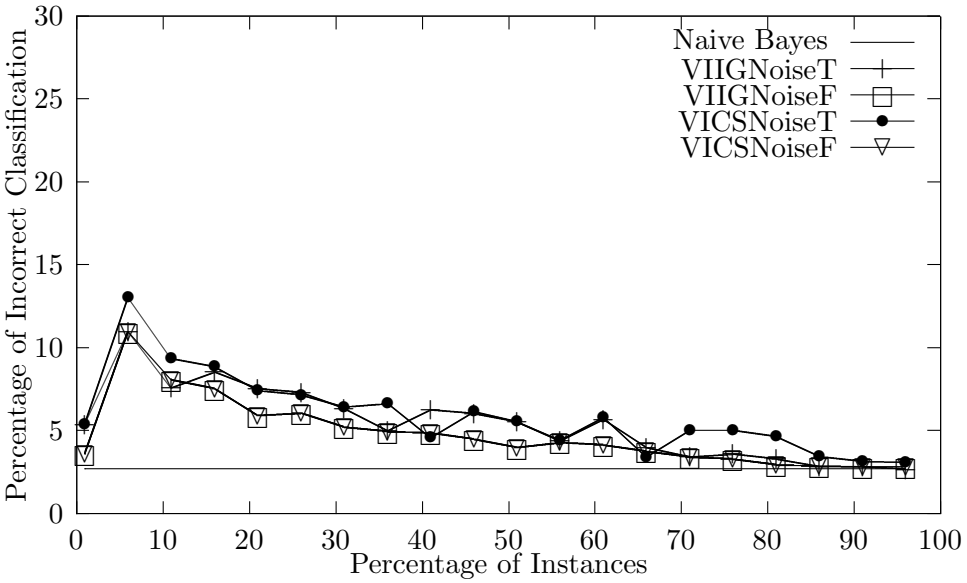


Figure 5.4: Varying percentage of instance graph on the mushroom data set

In Figure 5.5, we aimed to analyze results in different perspectives using box plot diagrams because investigating significance of the difference of noisy and noiseless results can be difficult at some points as we are operating our instance graphs with using little intervals of percentages. We also apply this process to other instance graphs of data sets. Due to 5.5, we can extract that mean and max values of noisy data is larger than noiseless data. This is the reason of why lengths of noiseless boxplot results are more taller than lengths of noisy boxplot. Repeated experiments are showing the same results as they should be in 5.7 and 5.9

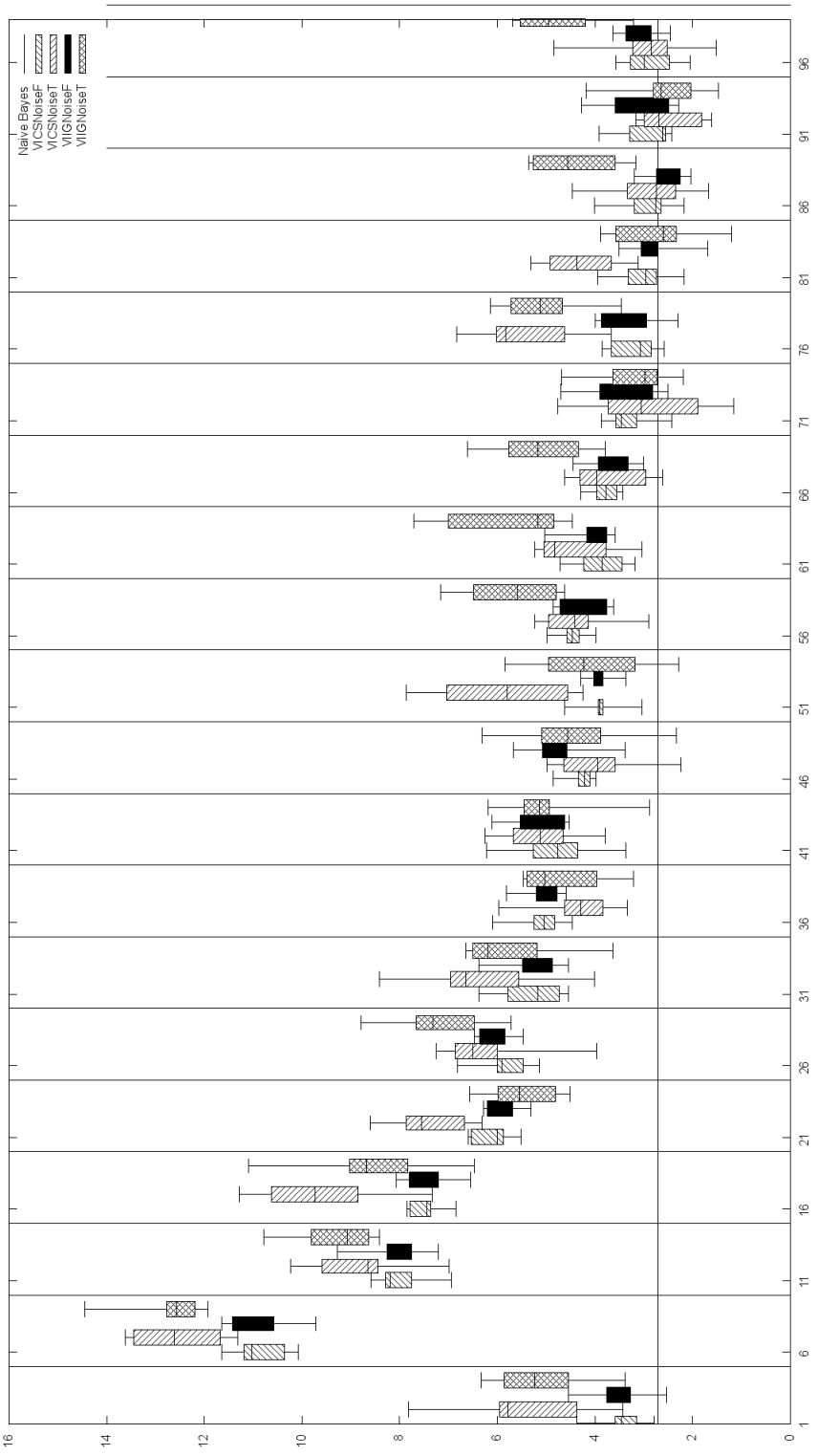


Figure 5.5: Varying percentage of instance boxplot graph on mushroom data set

Figure 5.6 shows effect on error rate on classification of changing instances with different subset evaluators, which are chisquared and infogain, applied to perturbed and noiseless data of the nursery data set. As it seen, the more instances are taken into evaluator the more stable (lower) pct. of error rate are produced for the nursery data set.

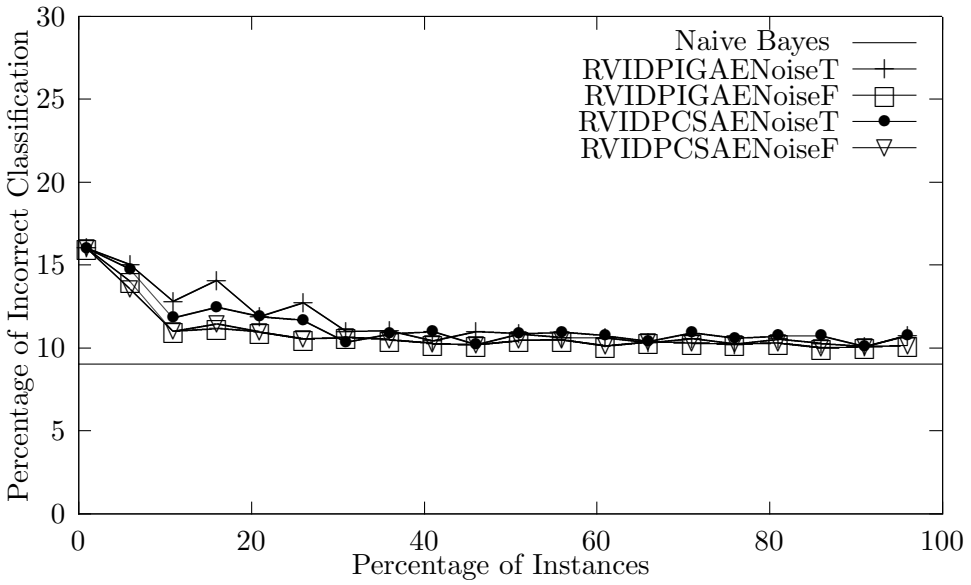


Figure 5.6: Varying percentage of instance graph on the nursery data set

Although selected epsilon value is closer to used in analyzing of the mushroom data set, given noise seems less because the instance number of the nursery data set is much more than mushroom. With more instance, training data number also increases. Increasing training data makes classification easier and cause to decrease error rate.

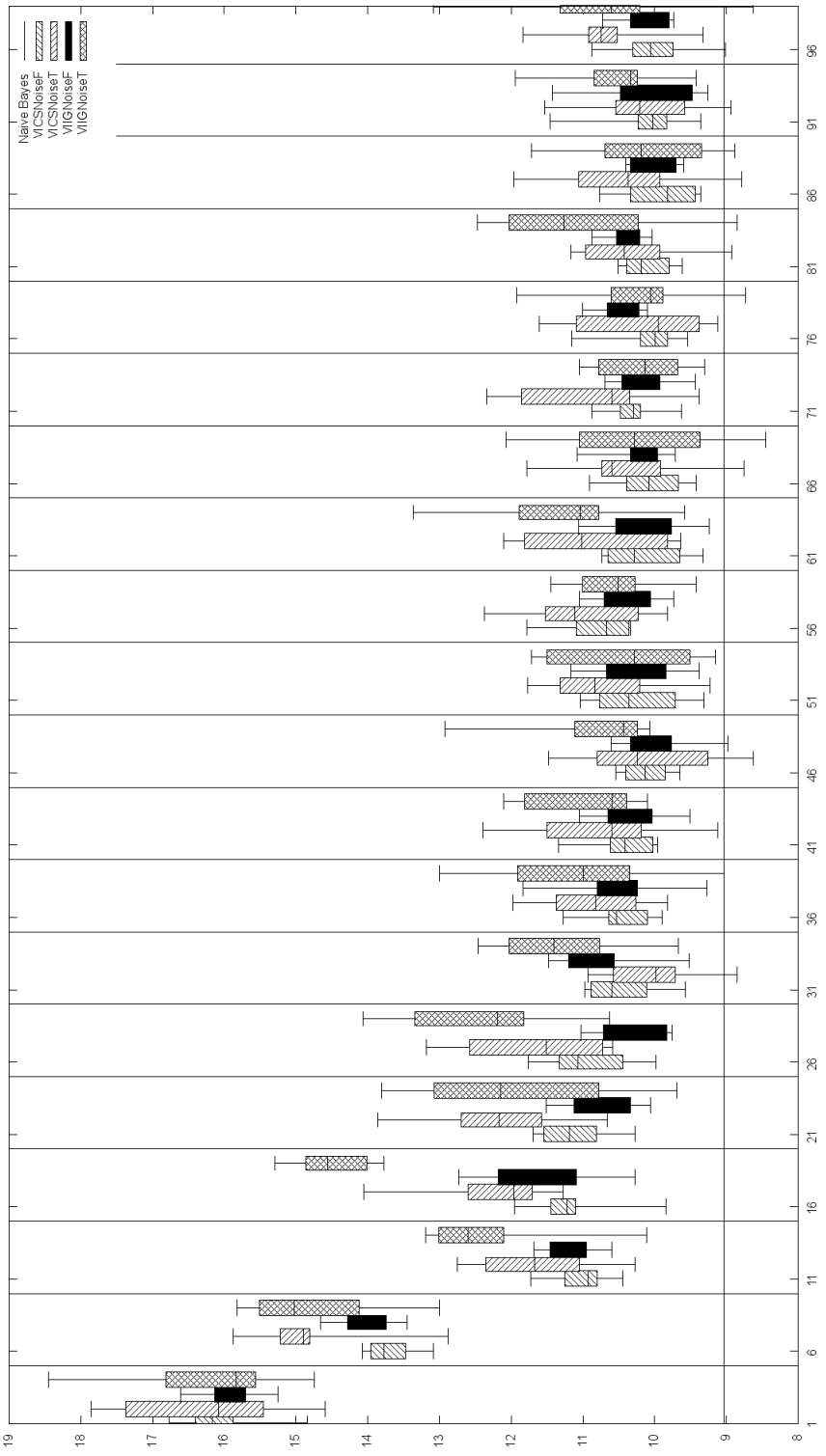


Figure 5.7: Varying percentage of instance boxplot graph on the nursery data set

Figure 5.8 shows effect on error rate on classification of changing instances with different subset evaluators, which are chisquared and infogain, applied to perturbed and noiseless data of the adult data set. In this example, noisy chisquare and infogain results also have parallel form with noiseless chisquare and infogain results but only little difference between them.

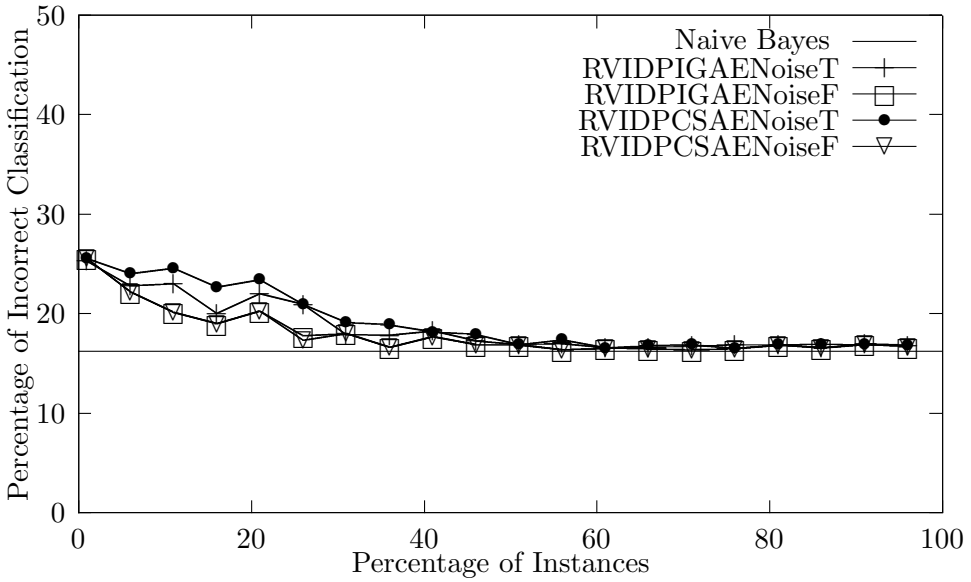


Figure 5.8: Varying percentage of instance graph on the adult data set

This data set is a very big data set with 48841 instance so to see effects of noise on this data set more difficult but we can see changing effects in the beginning of the graph and this effects disappearing close to the 100%. We set default epsilon value less than the mushroom and nursery data set's default epsilon value. Because less epsilon value, makes noise greater and more instance number want more noise adding to see the result effect.In the beginning less instance count(5%) cause to classifier's find true classes but increasing instance number it's difficult to select true classes.

Noisy results can be seen above the noiseless lines. Noisy results error rate both provide wanted effect with little diversion and close to the noiseless results with similar form.

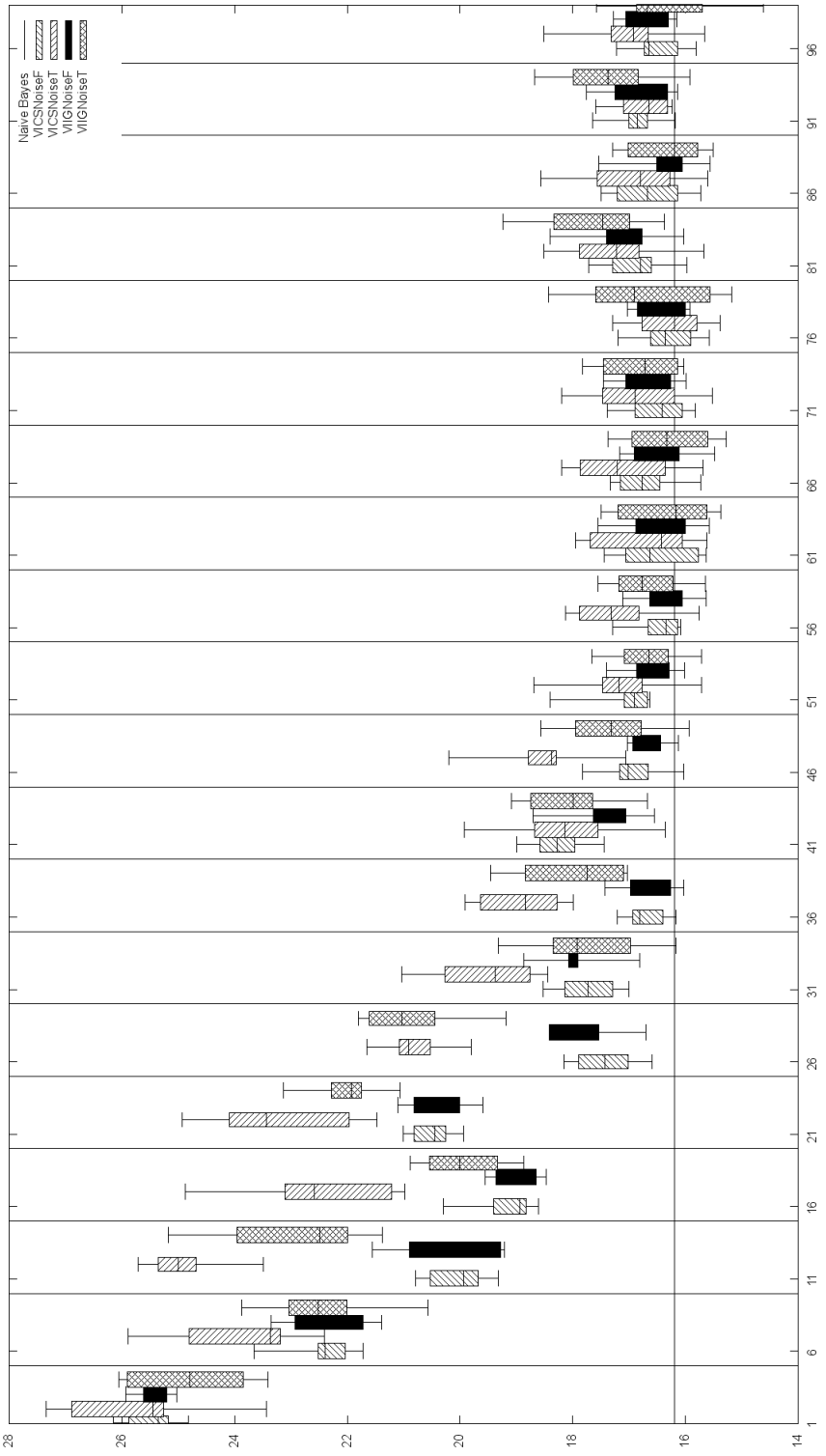


Figure 5.9: Varying percentage of instance boxplot graph on the adult data set

5.3 Effects of Changing Attribute

Figure 5.10 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing attribute numbers on the mushroom data set. Selecting less attribute cause to classification more easy first part(0-10). After 10 attribute both noiseless and noisy results error rates increasing because of more attributes mislead classifier while choosing right class. Through last attributes noisy and noiseless lines get close to Naive Bayes line as in the end we select every attribute.

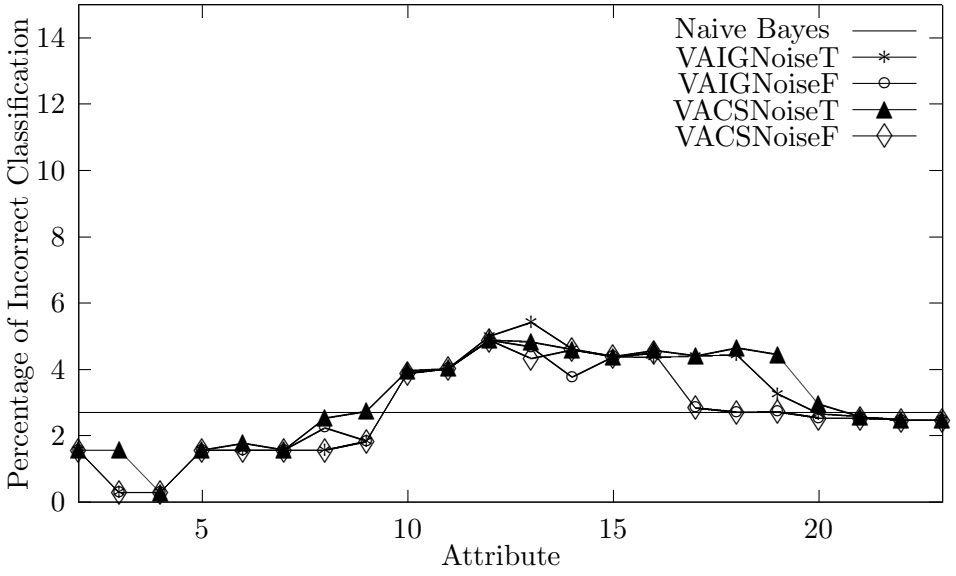


Figure 5.10: Varying number of attribute graph on the mushroom data set

Figure 5.11 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing attribute values on the nursery data set. This result form is little different from mushroom result. We increase attribute number every step and selecting is in order. Mushroom's first attributes more dominant on classification. While # of attributes are increasing, noise effects are appearing. Towards the end of graph, all lines (noisy and noiseless) get close Naive Bayes pct error line.

Mushroom has a special situation with class distribution and difference's reason is that. Mushroom classes pct is very similar each other (edible 51.8%, poison 48.2%) and adding noise with little attribute influence in a better way, cause less error rate first part of mushroom varying attributes graph.

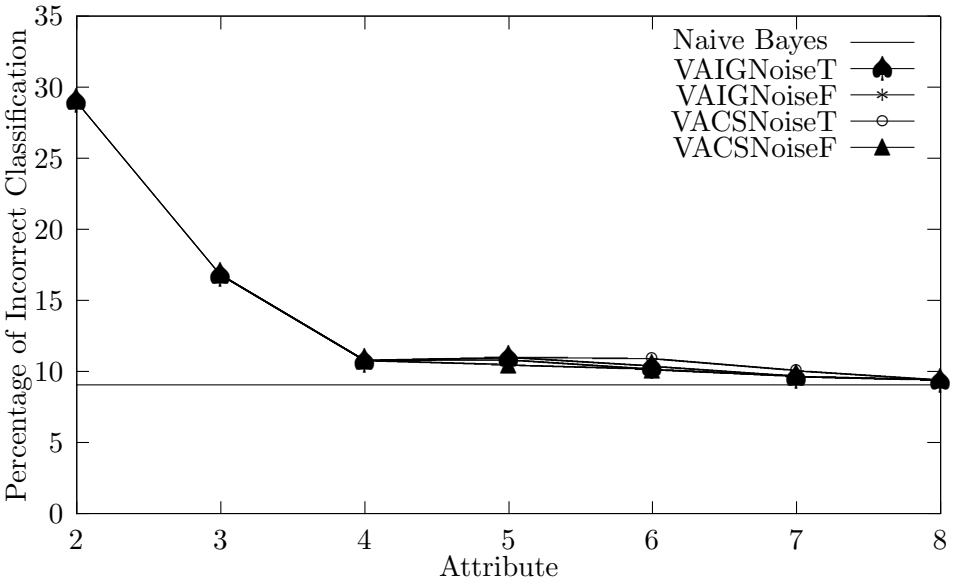


Figure 5.11: Varying number of attribute graph on the nursery data set

Figure 5.12 shows percentage of error rate on classification using two subset evaluators (chisquared, infogain) for changing attributes on the adult data set. This graph look alike to Figure 5.11, varying attribute graph the nursery data set but more softer than. The Adult data set has more instance two times from the nursery data set and has also more attribute this make adult graph's transitions softer.

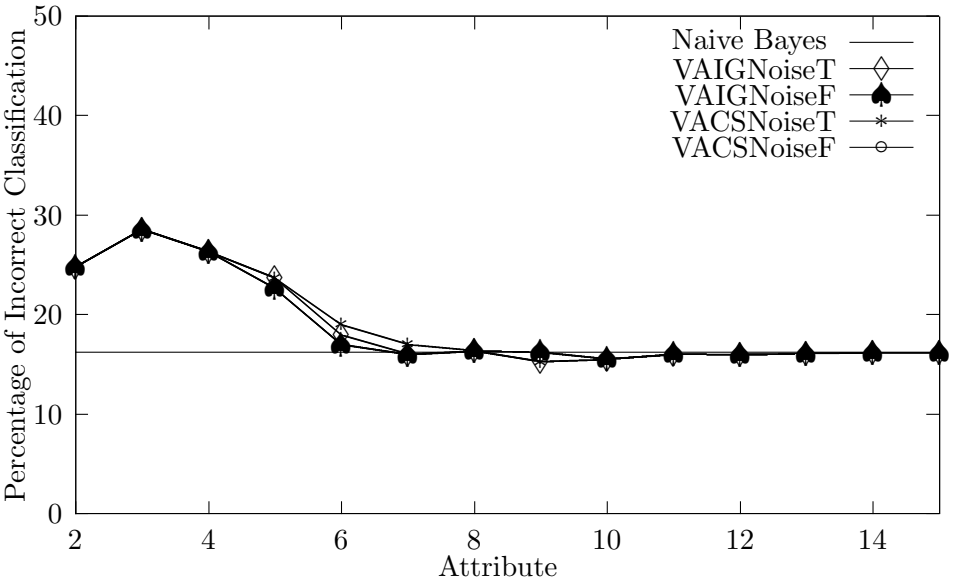


Figure 5.12: Varying number of attributes graph on the adult data set

Chapter 6

Related Work

Many studies has been done in the field of data security and privacy. As we mentioned in Chapter 1, data sanitization techniques are one of them. Sweeney [1] made a study about combining generalization and suppression to achieve k-anonymity. Aggarwal showed ineffectiveness of k-anonymity (difficulty of anonymization) when data set has a large number of attributes [14]. After several attacks on k-anonymity, l-diversity notation was introduced. Machanavajjhala et al. [2] showed the ineffectiveness of k-anonymity and proposed a new privacy definition called l-diversity. L-diversity, in addition to k-anonymity, is also concerned with the number of distinct values on sensitive attributes within each anonymization group. L-diversity requires that each anonymization group exhibit at least l different values on a sensitive attribute. Li et al. [15] further limits this constraint by requiring that the distribution of a sensitive value within an anonymization group resemble that of the entire population as closely as possible.

Differential privacy is provided by data perturbation and data perturbation includes data swapping, adding noise to the values, adding noise to the result of the query and sampling. Zou et al. [16] handle data modification and data swapping. They represent a new method that makes data swapping steps easier by reducing execution time on large data sets. Evans et al. [17] discusses how noise may be added to micro-data. Shlomo et al. [18] focus on sampling perturbation with using probabilistic differential privacy on their study. The work by Kadampur et al. [19] shows how decision trees can be constructed according to differential privacy.

Feature selection methods can be categorized as filter methods, wrapper methods and embedded methods. Most popular method of all is the filter method. Filter method, which also we used in our study, calculates scores according to evaluator functions for each attribute and selects the best n features by order of their scores. Yang et al. [20]

shows that information gain and chi-square are the most effective methods of feature selection on filter method studies.

Soria-Comas et al. [21] studied a different branch of differential privacy. Their study is on finding the optimal data-independent noise distribution that achieves ϵ-differential privacy and gives better results on multi-varied query functions.

There are also lots of studies on specific application areas of differential privacy like patient's data security. Guang et al.[22] applies differential privacy to patient data sets over the framework PINQ to give better results. Lee et al.[23] studied mobile privacy of wellness in health care services with feature selection. Divanis et al.[24] presents a survey of algorithms which provide privacy preserving of electronic health records while publishing.

Chapter 7

Conclusion

We aimed to bring rates of possible classification errors to an optimal level by using differential privacy with feature selection. We reached successful results by maintaining utility and privacy at the same time. Generally in experimental results, lines that show noisy and noiseless results are close to each other and it gives results strayed from clear data. We also show Naive Bayes error rates in our figures to e selection, no privacy case. We applied our study to a lot of data sets but listed here only a subset of results for representative data sets.

In future work, we plan to investigate alternative querying strategies for infogain and chisquare attribute evaluation strategies. We will also expand the set of evaluators and experiment with other classification methods such as decision tree induction. In addition to these, another research alternative would be applying differential privacy to other pre-processing tasks of classification such as supervised discretization.

Bibliography

[1] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, October 2002. ISSN 0218-4885. doi: 10.1142/S021848850200165X. URL `http://dx.doi.org/10.1142/S021848850200165X`.

[2] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217302. URL `http://doi.acm.org/10.1145/1217299.1217302`.

[3] Cynthia Dwork. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3-540-79227-9, 978-3-540-79227-7. URL `http://dl.acm.org/citation.cfm?id=1791834.1791836`.

[4] Murat Kantarcioglu. *Privacy-Preserving Distributed Data Mining And Processing On Horizontally Partitioned Data*. PhD thesis, Purdue University, 08 2005.

[5] Jaideep Vaidya. *Privacy Preserving Data Mining over Vertically Partitioned Data*. PhD thesis, Purdue University, 08 2004.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL `http://doi.acm.org/10.1145/1656274.1656278`.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-32731-8. doi: 10.1007/11681878_14. URL `http://dx.doi.org/10.1007/11681878_14`.

[8] Xiaokui Xiao and Yufei Tao. Output perturbation with query relaxation. *Proc. VLDB Endow.*, 1(1):857–869, August 2008. ISSN 2150-8097. doi: 10.14778/1453856.1453949. URL `http://dx.doi.org/10.14778/1453856.1453949`.

[9] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant Features and the Subset Selection Problem. In *International Conference on Machine Learning*, pages 121–129, 1994. URL http://citeseer.ist.psu.edu/john94irrelevant.html.

[10] Zhongzhe Xiao, Emmanuel Dell, Weibei Dou, and Liming Chen. Esfs: A new embedded feature selection method based on sfs.

[11] P.D. Allison. *Missing Data*. Number 136. no. in Missing Data. SAGE Publications, 2002. ISBN 9780761916727. URL https://books.google.com.tr/books?id=ZtYArHXjpB8C.

[12] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[13] M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

[14] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 901–909. VLDB Endowment, 2005. ISBN 1-59593-154-6. URL http://dl.acm.org/citation.cfm?id=1083592.1083696.

[15] Ninghui Li and Tiancheng Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *In Proc. of IEEE 23rd Int'l Conf. on Data Engineering (ICDE'07*, 2007.

[16] M. M. Zhang G. L. Zou. A new data perturbation method of reference control in statistical database. *Applied Mechanics and Materials*, 241-244:3134 – 3137, 2012. URL http://www.scientific.net/AMM.241-244.3134.

[17] Laura Zayatz Timothy Evans and John Slanta. Using noise for disclosure limitation of establishment tabular data. *Journal of Official Statistics*, 14(4):537 – 551, 1998. URL http://www.jos.nu/Articles/article.asp.

[18] Natalie Shlomo and Chris J. Skinner. Privacy protection from sampling and perturbation in survey microdata. *Journal of Privacy and Confidentiality*, 4, 2012. URL http://repository.cmu.edu/jpc/vol4/iss1/7.

[19] Mohammad Ali Kadampur and Durvasula V. L. N. Somayajulu. A noise addition scheme in decision tree for privacy preserving data mining. *CoRR*, abs/1001.3504, 2010. URL http://arxiv.org/abs/1001.3504.

[20] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3. URL http://dl.acm.org/citation.cfm?id=645526.657137.

[21] Jordi Soria-Comas and Josep Domingo-Ferrer. Optimal data-independent noise for differential privacy. *Information Sciences*, 250(0):200 – 214, 2013. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2013.07.004. URL http://www.sciencedirect.com/science/article/pii/S0020025513005094.

[22] D.G.Y. Lee. *Protecting Patient Data Confidentiality Using Differential Privacy: A Capstone Thesis*. Oregon Health & Science University, 2008. URL https://books.google.ie/books?id=m7bmPgAACAAJ.

[23] Nam Yeon Lee and Ohbyung Kwon. A privacy-aware feature selection method for solving the personalization-privacy paradox in mobile wellness healthcare services. *Expert Syst. Appl.*, 42(5):2764–2771, 2015. doi: 10.1016/j.eswa.2014.11.031. URL http://dx.doi.org/10.1016/j.eswa.2014.11.031.

[24] Aris Gkoulalas-Divanis, Grigorios Loukides, and Jimeng Sun. Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of Biomedical Informatics*, 50:4–19, 2014. doi: 10.1016/j.jbi.2014.06.002. URL http://dx.doi.org/10.1016/j.jbi.2014.06.002.

Appendix A

Graphical User Interface for Empirical Analysis

Figure A.1 is the graphical user interface of our program. Processing file represents name of the currently processed arff file. Specified path is same with path of the program jar file. If more than one file is wanted to be executed, jar of the program should be placed in the same path and this means more than one file can be executed at the same time. While program is running, we can see percentages of remaining time to finish currently work. The arff files that finish execute are placed in the classified files box as shown in Figure A.2.

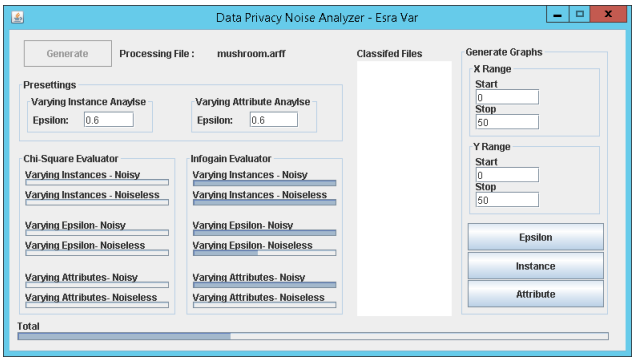


Figure A.1: Program is progressing .arff files

If we want to see the results obtained from a data set, firstly we should select by clicking the file in classified files box and select which parameters results we want to see by clicking buttons epsilon, instance and attribute. We can change X range and Y range from interface of the program and also if we want to repeat generation by changing epsilon value, we can update from pre-settings. After update generate button will be active, with clicking this button all analysis run again with desired epsilon values.

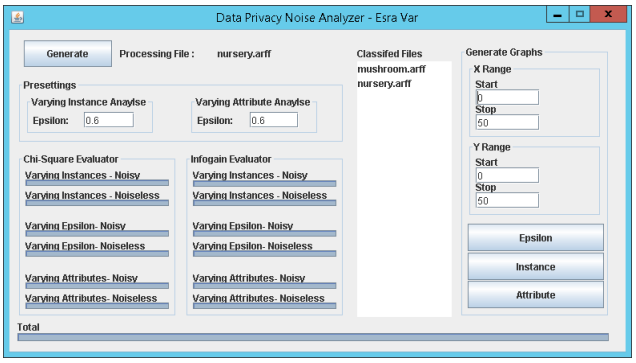


Figure A.2: Program completed the process

Source code of the software developed as part of this thesis can be reached at the following URL address from SourceForge.Net: `https://sourceforge.net/projects/dataprivacynoiseanalyser/`

Vitae

Curriculum Vitae