SUPPRESSING MICRODATA TO PREVENT CLASSIFICATION BASED INFERENCE

by

AYÇA AZGIN HİNTOĞLU

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

SABANCI UNIVERSITY
August 2011

SUPPRESSING MICRODATA TO PREVENT CLASSIFICATION BASED INFERENCE

APPROVED BY

Assoc. Prof. Dr. Yücel Saygın (Thesis Supervisor)	....
Assoc. Prof. Dr. Erkay Savaş	....
Assoc. Prof. Dr. Albert Levi	....
Assoc. Prof. Dr. Mehmet Keskinöz	....
Asst. Prof. Dr. Mehmet Ercan Nergiz	....

DATE OF APPROVAL: 09.08.2011

to my family

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor, Dr. Yücel Saygın, for his constant encouragement, mentoring, sustained support, and help. I have benefited greatly from his extensive knowledge in privacy preserving data disclosure and data mining research. He was always accessible for frequent discussions with me no matter how busy his schedule was. His many insightful comments and research ideas have inspired me to explore new directions in my research. My research endeavors would not have been successful without him.

My sincere appreciation goes to Assoc. Prof. Dr. Erkay Savaş, Assoc. Prof. Dr. Albert Levi, Assoc. Prof. Dr. Mehmet Keskinöz, and Asst. Prof. Dr. Mehmet Ercan Nergiz. I would like to thank them for serving on my supervising committee, for reading my thesis and making constructive suggestions.

Finally, my special thanks are due to my husband, Hakan, for his love and support throughout the years. I am also indebted to my parents and brother for their unconditional love, support and encouragement.

SUPPRESSING MICRODATA TO PREVENT CLASSIFICATION BASED INFERENCE

Ayça Azgın Hintoğlu
Electronics Engineering and Computer Science

Ph.D. Thesis, 2011

Thesis Supervisor: Assoc. Prof. Dr. Yücel Saygın

Keywords: Privacy, Data Disclosure Protection, Data Suppression, Data Perturbation, Data Generalization, Data Mining

Abstract

The revolution of Internet together with the progression in computer technology makes it easy for institutions to collect unprecedented amount of personal data. This pervasive data collection rally coupled with the increasing necessity of sharing of it raised a lot of concerns about privacy. Widespread usage of data mining techniques, enabling institutions to extract previously unknown and strategically useful information from huge collections of data sets, and thus gain competitive advantages, has also contributed to the fears about privacy.

One method to ensure privacy during disclosure is to selectively hide or generalize the confidential information. However, with data mining techniques it is now possible for an adversary to predict hidden or generalized confidential information using the rest of the disclosed data set. We concentrate on one such possible threat, classification, which is a data mining technique widely used for prediction purposes, and propose algorithms that modify a given microdata set either by inserting unknown values (i.e. deletion) or by generalizing the original values to prevent both probabilistic and decision tree classification based inference.

To evaluate the proposed algorithms we experiment with real-life data sets. Results show that proposed algorithms successfully suppress microdata and prevent both probabilistic and decision tree classification based inference. The hybrid versions of the algorithms, which aim to suppress a confidential data value against both classification models, block the inference channels with substantially less side effects. Similarly, the enhanced versions of the algorithms, which aim to suppress multiple confidential data values, reduce the side effects by nearly 50%.

VERİYİ BASTIRMAK SURETİYLE SINIFLANDIRMA TABANLI ÇIKARIMIN ENGELLENMESİ

Ayça Azgın Hintoğlu
Elektronik Mühendisliği ve Bilgisayar Bilimi
Doktora Tezi, 2011

Tez Danışmanı: Doc. Dr. Yücel Saygın

Anahtar Sözcükler: Mahremiyet, Verinin İfşa Edilirken Korunması, Veri Bastırma, Veri Karıştırma, Veri Genelleme, Veri Madenciliği

Özet

İnternet devrimi ve bilgisayar teknolojisinin ilerlemesi ile birlikte, kurumların daha önce benzeri görülmemiş miktarda kişisel veri toplaması mümkün olmuştur. Yaygınlaşan veri toplama aktiviteleri, artan veri paylaşma ihtiyacı ile birleştiğinde veri mahremiyeti ile ilgili endişeleri tetiklemiştir. Ayrıca kurumların oldukça büyük veri setlerinden önceden bilinmeyen ancak stratejik olarak faydalı bilgileri bulmasını sağlayan veri madenciliği tekniklerinin yaygınlaşması da mahremiyetle ilgili endişeleri arttırmıştır.

Veri paylaşımı esnasında mahremiyeti sağlamanın bir yolu gizlenmesi gereken veri alanlarının tek tek saklanması ya da genellenmesidir. Ancak, veri madenciliği teknikleri ile kötü niyetli kullanıcıların verinin geri kalanını kullanarak, saklanmış ya da genellenmiş veri alanlarını tahmin etmesi mümkün olmaktadır. Bu tez kapsamında popüler tahminsel veri madenciliği tekniklerinden biri olan sınıflandırmaya odaklanılarak, verilen bir veri setini gerek veri alanlarını silerek gerekse genelleyerek güncelleyen, olasılıksal ve karar ağacı kökenli sınıflandırma tekniklerine dayalı çıkarımları önleyen algoritmalar önerilmektedir.

Önerilen algoritmaların performansları gerçek veri setleri kullanılarak test edilmiştir. Test sonuçları, önerilen algoritmaların veri setlerini başarıyla baskıladığını ve hem olasılıksal hem de karar ağacı kökenli sınıflandırma tekniklerine dayalı çıkarımları engellediğini göstermiştir. Algoritmaların aynı anda hem olasılıksal hem de karar ağacı kökenli sınıflandırma tekniklerine dayalı çıkarımları önleyen melez sürümleri, gizli verileri çok daha az yan etki ile saklamayı başarmıştır. Benzer şekilde, algoritmaların birden fazla gizli veri alanını saklamayı hedefleyen gelişmiş sürümlerinin, yan etkileri %50 civarında azalttığı gözlenmiştir.

Contents

List of Figures

List of Tables

List of Abbreviations

CSP Cell suppression problem

DECP The suppression algorithm which aims to decrease probability of actual confidential value below the probability of the random next best guess using the deletion modification technique

DECP-G The suppression algorithm which aims to decrease probability of actual confidential value below the probability of the random next best guess using the generalization modification technique

DROPP The suppression algorithm which aims to drop probability of actual confidential value below the probability of the random next best guess using the deletion modification technique

DROPP-G The suppression algorithm which aims to drop probability of actual confidential value below the probability of the random next best guess using the generalization modification technique

e-DECP The enhanced suppression algorithm which aims to decrease probability of actual confidential value below the probability of the random next best guess using the deletion modification technique

e-DECP-G The enhanced suppression algorithm which aims to decrease probability of actual confidential value below the probability of the random next best guess using the generalization modification technique

e-DROPP The enhanced suppression algorithm which aims to drop probability of actual confidential value below the probability of the random next best guess using the deletion modification technique

e-DROPP-G The enhanced suppression algorithm which aims to drop probability of actual confidential value below the probability of the random next best guess using the generalization modification technique

HID3 The suppression algorithm which aims to avoid decision tree classification based inference using the deletion modification technique

HID3-G The suppression algorithm which aims to avoid decision tree classification based inference using the generalization modification technique

INCP The suppression algorithm which aims to increase probability of class attribute values in next best guess set above the probability of actual confidential value using the deletion modification technique

INCP-G The suppression algorithm which aims to increase probability of class attribute values in next best guess set above the probability of actual confidential value using the generalization modification technique

MSP Microdata suppression problem

NBG Next best guess

NRD Naive row deletion

RNBG Random next best guess

Chapter 1

INTRODUCTION

This chapter introduces the main issue addressed and the necessary background for the thesis. A brief description of our approaches and the outline of the structure of the thesis are provided.

1.1 Background and Motivation

In tandem with the advances in networking and storage technologies, the private sector as well as the public sector has increased their efforts to gather, manipulate, and commodify information on a large scale. Non-governmental organizations collect large amounts of personal information about their customers or members for many reasons including better customer relationship management and high-level decision making. Public safety, on the other hand, is the major motivation for large-scale personal information collection efforts initiated by governmental organizations. This pervasive data harvesting efforts coupled with the increasing need to share the data with other institutions or with public raised concerns about privacy[3]. Privacy is the ability of an individual to prevent information about himself becoming known to other people without his approval [4]. More specifically, it is the right of individuals to have the control over the data they provide. This includes controlling (1) how the data is going to be used, (2) who is going to use it, and (3) for what purpose.

Table 1.1: Academic Health Medical Records

Name	**Zipcode**	**Gender**	**Age**	**Indigestion**	**Chest Pain**	**Palpitation**	**Diagnosis**
Alice	90302	Female	29	Y	N	Y	Dyspepsia
Bob	90410	Male	22	N	Y	Y	Angina Pectoris
John	90301	Male	27	Y	N	N	Dyspepsia
Lisa	90310	Female	43	Y	N	N	Gastritis
Chris	90301	Male	52	N	Y	Y	Gastritis
Leo	90410	Male	47	Y	Y	Y	Angina Pectoris
Prue	90305	Female	30	N	N	Y	Angina Pectoris
Joe	90402	Male	36	N	Y	Y	Angina Pectoris
Ross	90301	Male	52	Y	Y	Y	Gastritis

Widespread usage of powerful data analysis tools and data mining techniques, enabling institutions to extract previously unknown and strategically useful information from huge collections of data sets, and thus gain competitive advantages, has also contributed to the fears about privacy. Data mining techniques can be used for many reasons including but not limited to national security warning and national security decision making [1] for government agencies, and providing better business intelligence and customer relationship management for enterprises. On the other hand, they can also be used by adversaries to infer hidden confidential, i.e. sensitive, information about individuals from the disclosed data sets, and thus pose a great threat to privacy. The security and privacy threats due to use of data mining techniques was first pointed out by O'Leary [43] and was discussed further in a symposium on knowledge discovery in databases and personal privacy [44; 33; 45; 51]. Since then, privacy issues have become one of the most important aspects of database and data mining research.

Example 1. Consider an on-line federation of hospitals and research organizations collaborating with each other, named *HealthFed*. Each federated hospital collects medical records of their patients together with their privacy preferences, and interacts with research organizations within the federation to share this information. In particular, assume that the city clinic Academic Health and Academic Research Institute, both being part of the *HealthFed* federation, collaborate with each other for research purposes. More specifically, Academic Health shares patients' medical records with

Table 1.2: Academic Health Medical Records Shared with Academic Research Institute

Zipcode	**Gender**	**Age**	**Indigestion**	**Chest Pain**	**Palpitation**	**Diagnosis**
90302	Female	29	Y	N	Y	Dyspepsia
90410	Male	22	N	Y	Y	?
90301	Male	27	Y	N	N	Dyspepsia
90310	Female	43	Y	N	N	Gastritis
90301	Male	52	N	Y	Y	Gastritis
90410	Male	47	Y	Y	Y	Angina Pectoris
90305	Female	30	N	N	Y	Angina Pectoris
90402	Male	36	N	Y	Y	Angina Pectoris
90301	Male	52	Y	Y	Y	Gastritis

Academic Research Institute after ensuring the privacy preferences of each patient are satisfied. Table 1.1 shows a set of such patients who gave consent to Academic Health to disclose their medical records to third parties for research purposes provided that their Name attribute is removed before disclosure. However, Bob, knowing that it might still be possible to link his medical records with other data sources through potentially identifying attributes like gender, zipcode, and age, required not only his name but also his diagnosis information to be hidden before disclosure. Therefore, Academic Health removed not only the Name attribute but also the diagnosis information from Bob's medical records before sharing it, as shown in Table 1.2. Unfortunately, given these medical records, Academic Research Institute can easily find Bob's diagnosis to be Angina Pectoris using a predictive data mining technique called classification.

1.2 Our Approaches

In this work, we address this particular problem of privacy preserving microdata disclosure. We assume that each individual might have different preferences regarding to their privacy. Therefore, the confidential attributes might differ for each individual. In such a setting, one method to ensure privacy while disclosing a microdata set is to selectively hide (i.e. replace with a symbol denoting unknown) or generalize the confidential data

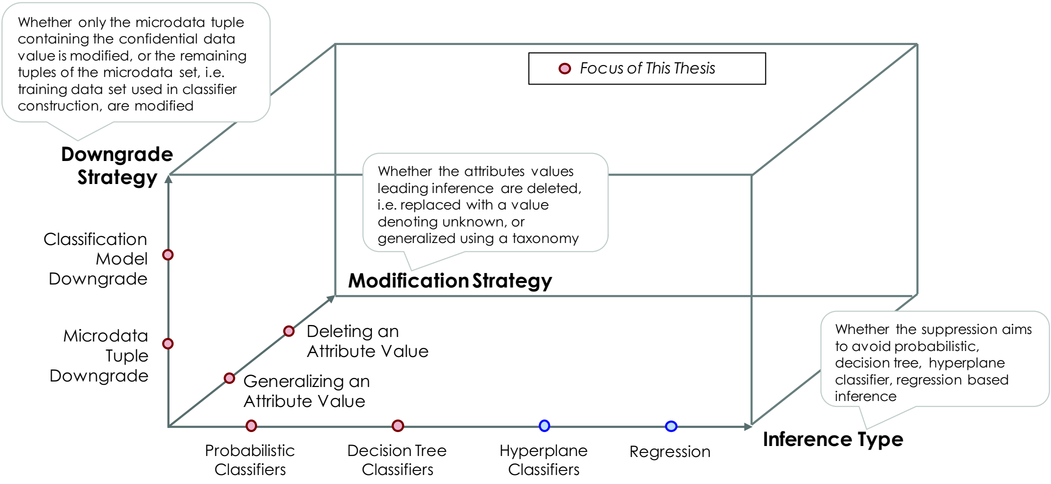

Whether only the microdata tuple containing the confidential data value is modified, or the remaining tuples of the microdata set, i.e. training data set used in classifier construction, are modified
Focus of This Thesis
Downgrade Strategy
Whether the attributes values leading inference are deleted, i.e. replaced with a value denoting unknown, or generalized using a taxonomy
Classification Model Downgrade
Modification Strategy
Microdata Tuple Downgrade
Deleting an Attribute Value
Generalizing an Attribute Value
Whether the suppression aims to avoid probabilistic, decision tree, hyperplane classifier, regression based inference
Inference Type
Probabilistic Classifiers
Decision Tree Classifiers
Hyperplane Classifiers
Regression

Figure 1.1: Main Focus Areas of This Thesis

values. This method ensures privacy from a micro-level perspective. But, this is not the case for the macro-level perspective, as with powerful data analysis tools and data mining techniques it is now possible for an adversary to predict hidden or generalized confidential information using the rest of the disclosed data set. We concentrate on one such possible threat, classification, which is a data mining technique widely used for prediction purposes, and propose algorithms that modify a given microdata set to prevent both probabilistic and decision tree classification based inference. We select Naïve Bayesian and ID3 as typical representatives of probabilistic and decision tree classifiers respectively, and develop our algorithms accordingly. Apart from avoiding different inference types, the algorithms proposed either employ different modification strategies or different downgrade strategies as pointed out in Figure 1.1.

More specifically, we design and implement the following algorithms which employ different modification and downgrade strategies, and aim to avoid either probabilistic or decision tree classification based inference.

1. The DECP and INCP algorithms suppress a single confidential data value against Naïve Bayesian classifier. These two algorithms downgrade the Naïve Bayesian classifier by identifying a set of data values from the rest of the data set that might

cause confidential information to be inferred, and deleting them (i.e. replacing them with a special symbol indicating unknown).

2. The DROPP and HID3 algorithms suppress a single confidential data value against Naïve Bayesian and ID3 classifiers respectively. These two algorithms downgrade the microdata tuple containing the confidential data value. They identify a set of data values from the microdata tuple itself that might cause confidential information to be inferred, and delete them.

3. The DECP-G and INCP-G algorithms suppress a single confidential data value against Naïve Bayesian classifiers. These two algorithms downgrade the Naïve Bayesian classifiers by identifying a set of data values from the rest of the data set that might cause confidential information to be inferred, and generalizing them.

4. The DROPP-G and HID3-G algorithms suppress a single confidential data value against Naïve Bayesian and ID3 classifiers respectively. These two algorithms downgrade the microdata tuple containing the confidential data value. They identify a set of data values from the microdata tuple itself that might cause confidential information to be inferred, and generalize them.

5. The e-DECP and e-DROPP algorithms suppress multiple confidential data values against Naïve Bayesian classifiers using the deletion modification strategy.

6. The e-DECP-G and e-DROPP-G algorithms suppress multiple confidential data values against Naïve Bayesian classifiers using the generalization modification strategy.

1.3 Outline of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2, a brief introduction to learning classifiers from a given training data set is given. We then formally define the problem of suppressing microdata to prevent classification based inference, and give a brief survey of the related work on privacy preserving data disclosure. In Chapter 3, we present the microdata suppression algorithms that prevent classification based inference using the deletion modification strategy. We describe in detail how we downgrade (i) the Naive Bayesian Classifier, and (ii) the microdata tuple to prevent inference of a confidential data value. In Chapter 4, we present the microdata suppression algorithms that prevent classification based inference using the generalization modification strategy. We describe in detail how we downgrade (i) the Naive Bayesian Classifier, and (ii) the microdata tuple to prevent inference of a confidential data value. Then, we present the performance evaluation results of the proposed algorithms in Chapter 5. Finally, we conclude the thesis in Chapter 6 where a summary and conclusions from this study are given. Some interesting future research problems are also addressed in Chapter 6.

Chapter 2

PRELIMINARY CONCEPTS AND RELATED WORK

This chapter provides preliminary definitions and problem formulations for this thesis. First, we describe the microdata suppression problem. Then, we define the modification and downgrade strategies for microdata suppression. Next, we provide the metrics that are used evaluate the proposed algorithms. Finally, we examine related work in literature on protecting privacy while disclosing microdata.

2.1 Problem Formulation

Let $\Lambda = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ be the set of attributes with associated domains[1] V_{α_1}, V_{α_2} ,..., V_{α_n}, and extended domains[2] eV_{α_1}, eV_{α_2}, ..., eV_{α_n} respectively. Let $D = \{d_1, d_2, ..., d_m\}$ be the microdata set where each tuple $d_i \in eV_{\alpha_1} \times eV_{\alpha_2} \times ... \times eV_{\alpha_n}$ is an ordered list of values.

For each attribute $\alpha_j \in \Lambda$, there is a mapping $\alpha_j[d_i] : eV_{\alpha_1} \times eV_{\alpha_2} \times ... \times eV_{\alpha_n} \rightarrow eV_{\alpha_j}$ from $eV_{\alpha_1} \times eV_{\alpha_2} \times ... \times eV_{\alpha_n}$ into the extended domain eV_{α_j}. The mapping $\alpha_j[d_i]$

[1] The domain of an attribute is represented by a finite set of discrete values excluding the unknown (i.e. null) value denoted by ν.

[2] The extended domain of an attribute is represented by a finite set of discrete values including the unknown (i.e. null) value denoted by ν such that $eV_{\alpha_j} = V_{\alpha_j} \cup \{\nu\}$.

represents the value of attribute α_j of microdata tuple d_i.

Similarly, for each microdata set D, there is a mapping $D[constraint] : D \rightarrow S \in 2^D$ from $D = \{d_1, d_2, ..., d_m\}$ into $S \in 2^D$. The mapping $D[constraint]$ represents the set of all tuples satisfying the *constraint*, expressed in conjunctive normal form, on attribute values. Examples of valid constraint expressions include the following:

- $\alpha_1[d] = val_1$,
- $\neg\ \alpha_1[d] = val_1$,
- $\alpha_i[d] = val_i \wedge \neg\ \alpha_j[d] = val_j$, and
- $\alpha_1[d] = val_1 \wedge \alpha_2[d] = val_2 \wedge ... \wedge \alpha_n[d] = val_n$.

Definition 2.1. Classifiers(Σ). *Σ denotes the set of all classifiers that aims to predict the value of a single attribute, i.e. the target attribute*[3] *α_τ, in terms of the predictor attributes*[4].

Each classifier $\varsigma \in \Sigma$ is defined in the context of a training data set and a target attribute. For example, a classifier of type *Naïve Bayesian* (i.e. NB) built using the data set D with $\alpha_\tau \in \Lambda$ as the target attribute is denoted as ${\varsigma_{nb}}^{D,\alpha_\tau}$. If the type of the classifier, the training data set or the target attribute is unknown or not relevant in a given context, then a special symbol $\perp$ is used instead of the respective symbol. For example, ${\varsigma_\perp}^{D,\alpha_\tau}$ denotes the set of all classifiers built using the data set D with $\alpha_\tau \in \Lambda$ as the target attribute. Following the training phase, each classifier $\varsigma \in \Sigma$ can be viewed as a function that takes a microdata tuple and predicts the most probable value of the target attribute based on other attributes' values. For example, if $\varsigma \in {\varsigma_\perp}^{\perp,\alpha_\tau}$ then $\varsigma : (eV_{\alpha_1} \times eV_{\alpha_2} \times \ldots \times eV_{\alpha_n}) \rightarrow V_{\alpha_\tau}$.

Definition 2.2. Naïve Bayesian Classifier(Σ_{nb}). *Let the j^{th} attribute value of tuple d_i, that is $\alpha_j[d_i]$, is unknown. According to* Bayes' theorem *the probability that $\alpha_j[d_i]$*

[3] Also called the class attribute or the dependent attribute

[4] Also called the independent attributes

has value $v \in V_{\alpha_j}$ *is equal to the posterior probability of* v *conditioned on* d_i *and is given by*

$$p(v|d_i) = \frac{p(v)p(d_i|v)}{p(d_i)} \tag{2.1}$$

where $p(v)$ *and* $p(d_i)$ *are the prior probabilities of* v *and* d_i *respectively, and* $p(d_i|v)$ *is the posterior probability of* d_i *conditioned on* v*. Naïve Bayesian classifier is a probabilistic classifier based on Bayes' theorem with the* class conditional independence *assumption, that is, the effect of an attribute value on another attribute (i.e. class attribute) is independent of the values of the remaining attributes. Due to* class conditional independence *assumption, we can rewrite the posterior probability* $p(d_i|v)$ *as follows:*

$$p(d_i|v) = \prod{}^{j-1}{}_{k=1} p(\alpha_k[d_i]|v) \prod{}^{n}{}_{k=j+1} p(\alpha_k[d_i]|v) \tag{2.2}$$

The Naïve Bayesian Classifier ${\varsigma_{nb}}^{D-d_i,\alpha_j}$ *built using* $D - d_i$ *as the training data set will predict the most probable value for* $\alpha_j[d_i]$ *as* $v_\pi \in V_{\alpha_j}$ *if and only if the following condition holds:*

$$p(v_\pi|d_i) > p(v|d_i) \mid \forall v \in V_{\alpha_j} - v_\pi \tag{2.3}$$

Since $p(d_i)$ *is same for all* $v \in V_{\alpha_j}$*, it can be ignored as shown below:*

$$p(v_\pi)p(d_i|v_{\mathit{\Pi}}) > p(v)p(d_i|v) \mid \forall v \in V_{\alpha_j} - v_\pi \tag{2.4}$$

Definition 2.3. ID3 Classifier (Σ_{id3}) *Let* $\alpha_j[d_i]$ *be unknown. The ID3 classifier* ${\varsigma_{id3}}^{D-d_i,\alpha_j}$ *built using* $D - d_i$ *as the training data set is a decision tree where each internal node represents a decision node, each branch represents an outcome of the decision and each leaf node represents a possible value* $v \in V_{\alpha_j}$ *for* $\alpha_j[d_i]$*. Such a classifier will predict the most probable value for* $\alpha_j[d_i]$ *as* $v_\pi \in V_{\alpha_j}$ *if and only if the test of the remaining attributes of* d_i *against the decision tree leads a path from the root node to a leaf node labeled with* v_π*.*

Definition 2.4. Suppressing a Confidential Data Value. *Let* D' *be the microdata set after applying a set of modifications to* D*. The confidential data value* $\alpha_j[d_i]$ *will be suppressed with respect to* D'*, if and only if there exist no classifiers that can correctly* *predict the confidential data value.*

$$\varsigma({d_i}') \neq \alpha_j[d_i] \ \ \forall \varsigma \in {\varsigma_\perp}^{D'-{d_i}',\alpha_j} \tag{2.5}$$

In this work, we relax the above statement such that there exists no Naïve Bayesian or ID3 classifier that can correctly predict the confidential data value.

$$\varsigma({d_i}') \neq \alpha_j[d_i] \ \ \forall \varsigma \in {\varsigma_{nb}}^{D'-{d_i}',\alpha_j} \cup {\varsigma_{id3}}^{D'-{d_i}',\alpha_j} \tag{2.6}$$

2.2 Modification Strategies for Microdata Suppression

There are two possible modification strategies that can be adopted to address the microdata suppression problem.

Modification Strategy 1. Deleting an Attribute Value. This modification scheme, also referred to as hiding, involves replacement of attribute values, including the confidential data value(s), with a special symbol denoting the unknown (i.e. null) value ν. Replacing attribute values with ν results in uncertainty in the microdata set. For example, in the simplest case of a binary attribute, an unknown value can be either 0 or 1. Assuming that the value was 0 will contribute to the resulting classification model in a contradicting way compared to the assumption that it was 1. By carefully hiding instances of certain attributes, we can decrease the precision of the classification models which can then be used to predict the confidential data values.

Definition 2.5. Taxonomy. *A taxonomy T_i for an attribute α_i is a tree structured concept hierarchy in the form of a partially ordered set $(eV_{\alpha_i}, \prec)$, where eV_{α_i} is the extended domain that enumerates all possible attribute values of α_i, and $\prec$ is the partial order that specifies is-a relationships among attribute values in eV_{α_i}. Figure 2.1 shows the taxonomy for the disease attribute.*

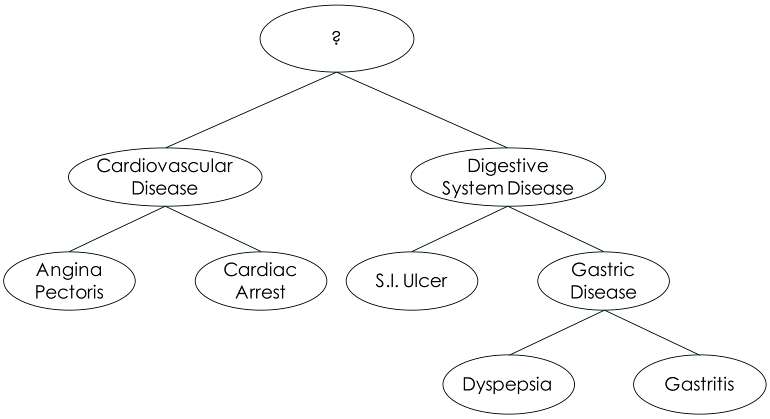


Figure 2.1: Disease Taxonomy

Modification Strategy 2. Generalizing an Attribute Value. Let us assume that $T = \{T_1, T_2, ..., T_n\}$ represents the ordered set of taxonomies associated with attributes $\Lambda = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ respectively. This modification scheme involves generalization of attribute values, including the confidential data value(s), using the set of taxonomies T.

2.3 Downgrade Strategies for Microdata Suppression

There are two possible downgrade strategies that can be adopted to address the microdata suppression problem.

Downgrade Strategy 1. Classification Model Downgrade. Let D be the original microdata set with the confidential data value $\alpha_j[d_i]$. Classification model downgrade aims to transform the original microdata set D to D' which satisfies the following constraints:

i. The confidential data value $\alpha_j[d_i]$ is either deleted or generalized,

ii. The tuple containing the confidential data value is not modified. The only exception to this is the confidential data value which is either deleted or generalized,

$$\forall \alpha \in \Lambda - \alpha_j, \alpha[d_i'] = \alpha[d_i]$$

iii. The remaining tuples of the microdata set are modified if and only if there exists at least one classifier that can correctly predict the actual confidential data value,

$$D - d_i \neq D' - d_i', \text{ iff } \exists \varsigma \in \varsigma_{\perp}{}^{D-d_i,\alpha_j} \varsigma(d_i) = \alpha_j[d_i]$$

iv. There exists no classifiers that can correctly predict the actual confidential data value using the modified microdata set.

$$\forall \varsigma \in \varsigma_{\perp}{}^{D'-d_i',\alpha_j} \varsigma(d_i') \neq \alpha_j[d_i]$$

This scheme aims at downgrading all classification models $\varsigma \in \varsigma_{\perp}{}^{D'-d_i',\alpha_j}$ by modifying the tuples $d \in D - d_i$.

Downgrade Strategy 2. Microdata Tuple Downgrade. Let D be the original microdata set with the confidential data value $\alpha_j[d_i]$. Microdata tuple downgrade aims to transform the original microdata set D to D' which satisfies the following constraints:

i. The confidential data value $\alpha_j[d_i]$ is either deleted or generalized,

ii. Remaining attribute values of the tuple containing the confidential data value are modified if and only if there exists at least one classifier that can correctly predict the actual confidential data value,

$$\exists \alpha \in \Lambda - \alpha_j, \alpha[d_i'] \neq \alpha[d_i], \text{ iff } \exists \varsigma \in \varsigma_{\perp}{}^{D-d_i,\alpha_j} \varsigma(d_i) = \alpha_j[d_i],$$

iii. The remaining tuples of the microdata set, which constitute the training data set for classifier construction, are not modified,

$$D - d_i = D' - d_i'$$

iv. The classifiers built using the modified microdata set are the same as the ones built using the original data set,

$$\varsigma_{\perp}{}^{D'-d_i',\alpha_j} = \varsigma_{\perp}{}^{D-d_i,\alpha_j}$$

v. There exists no classifiers that can correctly predict the actual confidential data value using the modified microdata tuple.

$$\forall\varsigma \in \varsigma_{\perp}{}^{D-d_i,\alpha_j},\ \varsigma(d_i') \neq \alpha_j[d_i]$$

Unlike the classification model downgrade, this scheme downgrades only the microdata tuple containing the confidential data value d_i, such that the classifiers $\varsigma \in \varsigma_{\perp}{}^{D-d_i,\alpha_j}$ cannot correctly predict the confidential data value $\alpha_j[d_i]$.

2.4 Evaluation Measures

The two important issues in microdata suppression are; (1) minimization of information loss enabling further use of the modified microdata set, and (2) maximization of uncertainty enabling protection of confidential data values from classification based inference. In the following, seven metrics for measuring information loss and uncertainty incurred by the suppression process are introduced respectively.

2.4.1 Information Loss Metrics

Within the scope of this thesis, three different metrics are used to measure the information loss: the *Direct Distance*, *Sum of Kullback Leibler Distances* and *Average Change in Mutual Information.*

The *Direct Distance*, the simplest of all information loss metrics, basically counts the number of attribute values hidden during the suppression process.

Definition 2.6. Direct Distance. *Let D and D' be the original and modified microdata sets respectively. The direct distance between D and D' can be defined as the number of non-matching attribute values.*

$$DD(D, D') = \sum_{i=1}{}^{m} \sum_{j=1}{}^{n} dist_{ij} \tag{2.7}$$

where

$$dist_{ij} = \begin{cases} 0 & \textit{if } \alpha_j[d_i] = \alpha_j[d_i'] \\ 1 & \textit{otherwise} \end{cases}$$

The second information loss metric, utilized within the scope of this thesis, is the *Sum of Kullback Leibler Distances*. This metric measures the information loss in terms of the distance between the first order probability distributions of the original and the modified microdata sets.

Definition 2.7. Kullback Leibler Distance. *Let D and D' be the original and modified microdata sets respectively. Let $\alpha \in \Lambda$ be an attribute with probability distribution p_α in D and $p_\alpha{}'$ in D'. The Kullback Leibler distance between D and D' in terms of attribute α can be defined as the distance between the first order probability distributions of α in D and D'.*

$$KLD(D, D') = D(p_\alpha || p_\alpha{}') = \sum_{v \in V_\alpha} p_\alpha(v) log \frac{p_\alpha(v)}{p_\alpha{}'(v)} \tag{2.8}$$

Definition 2.8. Sum of Kullback Leibler Distances. *Let D and D' be the original and modified microdata sets respectively. The sum of Kullback Leibler distances between D and D' over all attributes $\alpha \in \Lambda$ can be defined as follows.*

$$SKLD(D, D') = \sum_{\alpha \in \Lambda} D(p_\alpha || p_\alpha{}') \tag{2.9}$$

The last information loss metric, utilized within the scope of this thesis, is the *Average Change in Mutual Information*. This metric measures the information loss by finding the average change in joint probability distributions of all attributes.

Definition 2.9. Mutual Information. *Let $\alpha_k \in \Lambda$ and $\alpha_l \in \Lambda$ be two attributes of the microdata set D with probability distributions p_{α_k} and p_{α_l} respectively, and joint probability distribution p_{α_k,α_l}. The mutual information between α_k and α_l in D, measuring their mutual dependence, can be defined as follows.*

$$\begin{aligned} I_D(\alpha_k, \alpha_l) &= D(p_{\alpha_k,\alpha_l} || p_{\alpha_k} p_{\alpha_l}) \\ &= \sum{}_{v_k \in V_{\alpha_k}} \sum{}_{v_l \in V_{\alpha_l}} p_{\alpha_k,\alpha_l}(v_k, v_l) log \frac{p_{\alpha_k,\alpha_l}(v_k, v_l)}{p_{\alpha_k} p_{\alpha_l}} \end{aligned} \tag{2.10}$$

Definition 2.10. Average Change in Mutual Information. *Let D and D' be the original and modified microdata sets respectively. The average change in mutual information over all attributes $\alpha \in \Lambda$ can be defined as follows.*

$$ACMI(D, D') = \frac{2 \sum{}_{i=1}{}^{n} \sum{}_{j=i}{}^{n} \frac{I_D(\alpha_i;\alpha_j)}{I_{D'}(\alpha_i;\alpha_j)}}{n(n-1)} \tag{2.11}$$

2.4.2 Uncertainty Metrics

The *Sum of Conditional Entropies* is used to measure the uncertainty introduced into the modified microdata set.

Definition 2.11. Conditional Entropy. *Let D and D' be the original and modified microdata sets respectively, and $\alpha \in \Lambda$ be an attribute. Let ${X_\alpha}^D$ on eV_α be a random variable with instances $\alpha[d_1], \alpha[d_2], ..., \alpha[d_m]$ and probability distribution p_α. Let ${X_\alpha}^{D'}$ on eV_α be a random variable with instances $\alpha[d_1'], \alpha[d_2'], ...,$ $\alpha[d_m']$ and probability distribution p_α'. The conditional entropy of ${X_\alpha}^D$ given ${X_\alpha}^{D'}$ can be defined as follows.*

$$H({X_\alpha}^D | {X_\alpha}^{D'}) = -\sum{}_{v \in V_\alpha} \sum{}_{v' \in eV_\alpha} p(v, v') log(p(v|v')) \tag{2.12}$$

Definition 2.12. Sum of Conditional Entropies. *Let D and D' be the original and modified microdata sets respectively. The sum of conditional entropies of D given*

D' can be defined as follows.

$$SCE(D, D') = \sum\nolimits_{\alpha \in \Lambda} H({X_\alpha}^D | {X_\alpha}^{D'}) \tag{2.13}$$

The detailed descriptions of the information theoretic metrics introduced in this section can be found in [11].

2.5 Related Work

The problem of protecting privacy while disclosing public-use data sets were previously investigated in the context of statistical databases (SDBs) as the *statistical disclosure limitation* problem (also referred to as the *inference problem*)[5].The statistics literature, motivated by the need to publish statistical data sets with one or more contingency tables containing aggregate statistics, focused on identifying and protecting sensitive cells which may lead to derivation of aggregate confidential information. An extensive survey of statistical database security can be found in [5] and more recent work on disclosure control in statistical databases can be found in [17; 22]. According to [5] the methods proposed for securing SDBs from inference attacks can be mainly classified into four categories:*conceptual*, *query restriction*, *data perturbation* and *output perturbation.* Conceptual approaches include techniques that detect and remove inference channels during the database design mainly at the conceptual data model level. Query restriction approaches provide protection by restricting the query set size, controlling the overlap among successive queries, or making query results of small size unavailable to users of the database. On the other hand, data perturbation approaches introduce noise in the data by transforming the original database into a perturbed one. These approaches either replace the whole data set with a new one coming from the same probability distribution or perturb some of the attribute values once and for all. Finally, the output perturbation approaches perturb the answer to queries while leaving the data in the database unchanged.

One popular disclosure protection approach is *cell suppression* [12; 49]. Cell suppression consists of two sub-approaches:*primary suppression* and *secondary (i.e. complementary) suppression*. The basic idea of primary cell suppression is to find all sensitive cells that might cause confidential information to be disclosed from the released statistical data set and replace them by a symbol indicating the suppression. Yet primary suppression itself is not enough to protect the sensitive cells due to inference channels existing in the data set. In order to reach the desired protection for sensitive cells, other cells, i.e. marginal totals, containing nonconfidential information that might lead to inference of suppressed sensitive cells also needs to be suppressed; this is called secondary (complementary) cell suppression. Moreover, while finding a set of complementary suppressions protecting all sensitive cells, the information loss associated with the suppressed entries have to be minimized. This combinatorial optimization problem is known as the Cell Suppression Problem (CSP) in statistics literature. Since CSP belongs to the class of NP-hard problems [25; 31; 32], many heuristic methods have been proposed including but not limited to [12; 23; 24; 32; 49] (see [61] for more references) to address the problem. CSP problem is similar to the Microdata Suppression Problem (MSP) that this work tries to address. Nevertheless, the methodologies used to address these problems are quite different due to the difference in the types of data sets they are trying to protect. In statistical data sets, inference results from the marginal totals given along with the data itself. On the other hand, in microdata sets inference results from the statistical correlations between attributes like income and education.

Another popular disclosure protection approach that belongs to data perturbation family is *microaggregation* [20; 26; 34; 36; 41; 50; 52; 59]. Different from cell suppression, microaggregation aims at protecting numeric microdata by clustering individual records into small aggregates and replacing actual values of individual records by group means prior to publication. As Ferrer et al. pointed out in his work [20], microaggregation assumes that confidentiality rules in use allow publication of microdata sets if the individual records correspond to groups of k or more individuals. While an efficient polynomial algorithm exists for optimal univariate microaggregation [26], microaggregation of multivariate data guaranteeing minimum information loss is known to be

NP-hard [41]. Hence, several heuristic methods have been proposed [20; 50; 34] to address this problem. Recently, new heuristics employing genetic algorithms have been proposed to further lower the information loss [36; 52]. Moreover, microaggregation has been extended to handle categorical data by means of employing different clustering algorithms [59]. Different from MSP problem, microaggregation assumes all respondents contributed to the microdata set have the same privacy preferences. It is meaningful to use microaggregation in such a setting where sensitive attributes are the same for all respondents. Nevertheless, if respondents' privacy preferences differ then it will result unnecessary attribute values to be generalized meaning more information loss.

The security and privacy issues arising from the inference problem, which results in *private-sensitive* data to be inferred from *public-insensitive* data, has also been invastigated by multilevel secure databases research [30; 39; 46; 53; 54] and general purpose databases research [9; 13; 14; 28]. Methods proposed within the database context mainly focus on detection and removal of meta-data (i.e. database constraints like functional and multi-valued dependencies) based inferences either during database design [14; 27; 39; 54] or during query time [15; 53; 57]. However, they do not take into account the statistical correlations among database attributes which can be discovered by various data mining techniques and hence result in imprecise inferences like the rule 'A implies B' with 25% confidence.

There are also other approaches investigating the privacy issues arising during microdata disclosure within the scope of anonymization problem. *K-anonymity*, [47; 48; 55], being one of those approaches, aims at preserving the anonymity during the data dissemination process using generalizations and suppressions on potentially identifying portions of the data set. While k-anonymity protects against identity disclosure, it is insufficient to prevent attribute disclosure. To address this limitation of k-anonymity, Machanavajjhala et al. [37] recently introduced a new notion of privacy, called ℓ-diversity, which requires that the distribution of a sensitive attribute in each equivalence class has at least well-represented values. One problem with ℓ-diversity is that it is limited in its assumption of adversarial knowledge. It is possible for an adversary to gain information about a sensitive attribute as long as s/he has information about the global distribution of this attribute. Li et al. [35] addresses this particular problem by t-closeness which formalizes the idea of global background knowledge by requiring that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table (i.e., the distance between the two distributions should be no more than a threshold t). This effectively limits the amount of individual-specific information an observer can learn. Other approaches addressing the anonymization problem include [18; 19; 29; 42; 37; 35]. In his work [29], Iyengar uses suppression and generalization approaches to satisfy privacy constraints. Moreover, he examines the tradeoff between privacy and information loss within different data usage contexts and proposes a genetic algorithm to find the optimal anonymization. On the other hand, in [42] Øhrn et al. uses boolean reasoning, and in [18; 19] Ferrer et al. uses microaggregation to address the anonymization problem. Besides the fact that these approaches successfully preserve privacy through anonymization, none of them addresses the inference threat to privacy due to data mining approaches. Therefore, they do not directly apply to MSP. Moreover, similar to the microaggregation, anonymization approaches assume that each respondent who contributes to the microdata set has the same privacy preferences, i.e. wants to be anonymous, which is not realistic.

Another approach, proposed by Wang et al. [60], addresses the threats caused by data mining abilities, using a template-based approach. The proposed approach aims to (1) preserve the information for a given classification analysis, and (2) limit the usefulness of unwanted sensitive inferences, i.e. classification rules, that may be derived from the data. More specifically, it focuses on suppressing sensitive rules, instead of sensitive data values.

The work closest to ours is proposed by Chang et al.[10]. In his work, Chang proposes a new paradigm for dealing with the inference problem, which combines the application of decision tree analysis with the concept of *parsimonious downgrading*. He shows how classification models can be used to predict suppressed confidential data values and concludes that some feedback mechanism is needed to protect suppressed data values against classification models.

Chapter 3

SUPPRESSING MICRODATA TO PREVENT CLASSIFICATION BASED INFERENCE USING DELETION

As pointed in Section 2, hiding a confidential data value alone may not be enough to protect it, in case the whole data set is going to be disclosed. This results from the fact that an adversary can build a classification model using the rest of the data set as the training data set and s/he could use it to predict the actual confidential data value. In order to avoid such attacks, we propose four algorithms suppressing only one confidential data value at a time, against two popular classifier types: probabilistic and decision tree classifiers, as shown in Figure 3. We have selected Naïve Bayesian and ID3 as typical representatives of probabilistic and decision tree classifiers respectively, and developed our heuristics accordingly. Moreover, we propose enhancement to two of the proposed algorithms to suppress multiple confidential data values with fewer side effects.

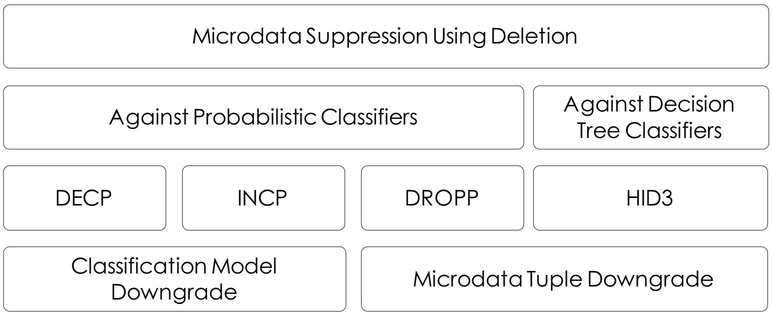

Microdata Suppression Using Deletion
Against Probabilistic Classifiers
Against Decision Tree Classifiers
DECP
INCP
DROPP
HID3
Classification Model Downgrade
Microdata Tuple Downgrade

Figure 3.1: Taxonomy of Microdata Suppression Algorithms using Deletion Modification Technique

3.1 Suppression Against Probabilistic Classification Models

In the following, we present three algorithms for preventing probabilistic classification based inference. The proposed algorithms aim to suppress a confidential data value, such that it is no longer among the Top-1 Probable value set.

Definition 3.1. Top-k Probable. *Let $\alpha_j[d_i]$ be confidential, thus be replaced by ν. The Naïve Bayesian Classifier $\varsigma_{nb}{}^{D-d_i,\alpha_j}$ built using $D-d_i$ as the training data set will predict the Top-k Probable value set for $\alpha_j[d_i]$ as $\Omega_k{}^{\alpha_j[d_i]} \subseteq V_{\alpha_j}$. The Top-k Probable value set satisfies the following constraints.*
i. It's size is equal to k.

$$\left|\Omega_k{}^{\alpha_j[d_i]}\right| = k \tag{3.1}$$

ii. The probability of $\alpha_j[d_i]$ being equal to the least probable value in the Top-k Probable value set is greater than the probability of $\alpha_j[d_i]$ being equal to the most probable value among the remaining attribute values.

$$p(\omega|d_i) > p(v|d_i) \mid \forall v \in V_{\alpha_j} - \Omega_k{}^{\alpha_j[d_i]} \wedge \omega \in \Omega_k{}^{\alpha_j[d_i]} \tag{3.2}$$

The proposed suppression algorithms aim at either reducing $p(\alpha_j[d_i]|d_i)$ below that of a randomly selected attribute, called the *Random Next Best Guess*, among Top-k Probable value set or increasing the probability of a set of selected attributes, called the *Next Best Guess Set*, above $p(\alpha_j[d_i]|d_i)$.

Definition 3.2. Random Next Best Guess. *The random next best guess, $v_{rnbg} \in V_{\alpha_j}$, is a randomly selected value from V_{α_j} satisfying the following conditions.*
i. It is different from $\alpha_j[d_i]$.

$$v_{rnbg} \neq \alpha_j[d_i] \tag{3.3}$$

ii. It is among the Top-k Probable value set.

$$v_{rnbg} \in {\Omega_k}^{\alpha_j[d_i]} \tag{3.4}$$

iii. The probability of ${\alpha_j}^{th}$ attribute of d_i being equal to v_{rnbg} is smaller than that of confidential data value $\alpha_j[d_i]$ and greater than zero.

$$p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i) > 0 \tag{3.5}$$

Definition 3.3. Next Best Guess Set. *The next best guess set, $S_{nbg} \subseteq {\Omega_k}^{\alpha_j[d_i]}$, for microdata tuple d_i is the set of all attribute values $v \in {\Omega_k}^{\alpha_j[d_i]} - \alpha_j[d_i]$ satisfying the following condition.*

$$S_{nbg} = \left\{v|v \in {\Omega_k}^{\alpha_j[d_i]} - \alpha_j[d_i] \wedge p(v|d_i) \geq p(v_{rnbg}|d_i)\right\} \tag{3.6}$$

3.1.1 DECP Algorithm

The DECP algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$ so that it cannot be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D'-d_i,\alpha_j}$. It accomplishes its goal by decreasing the probability $p(\alpha_j[d_i]|d_i)$ below that of the random next best guess v_{rnbg}.

Definition 3.4. Maximum Impact Attribute. *The attribute with maximum impact on $p(\alpha_j[d_i]|d_i)$, denoted by ${\alpha_{MI}}^{\alpha_j[d_i]}$, is the one that satisfies the following conditions.*

$$
\begin{aligned}
{\alpha_{MI}}^{\alpha_j[d_i]} = \arg\min{}_{\alpha \in \mathit{\Lambda}} \left(|D[\alpha_j[d] = \alpha_j[d_i] \wedge \alpha[d] = \alpha[d_i]] - d_i|\right) \\
\wedge\, |D[\alpha_j[d] = \alpha_j[d_i] \wedge \alpha[d] = \alpha[d_i]] - d_i| > 1 \qquad (3.7)
\end{aligned}
$$

Definition 3.5. Maximum Impact Data Values. *The maximum impact data values are the instances of ${\alpha_{MI}}^{\alpha_j[d_i]}$ in tuples $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]]$ excluding d_i.*

In each iteration, the DECP algorithm identifies the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ and modifies the tuples d, such that $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] - d_i$, by replacing ${\alpha_{MI}}^{\alpha_j[d_i]}[d]$ with ν until the goal is achieved, that is until $p(\alpha_j[d_i]|d_i)$ becomes less than $p(v_{rnbg}|d_i)$. Each such replacement results in the maximum possible reduction in $p(\alpha_j[d_i]|d_i)$, thus requiring less number of modifications.

Theorem 3.1. *Let ${\alpha_{MI}}^{\alpha_j[d_i]}$ be the maximum impact attribute satisfying Equation* (3.7)*. Then, every replacement of a maximum impact data value with ν causes the maximum decrease in $p(\alpha_j[d_i]|d_i)$, thus resulting in fewer data values to be modified.*

Proof: Let us first find the effect of replacing a maximum impact data value with ν on $p(\alpha_j[d_i])\ p(d_i|\alpha_j[d_i])$. Remember that, since $p(d_i)$ is same for all $v \in V_{\alpha_j}$, it can be ignored when calculating $p(\alpha_j[d_i]|d_i)$.

$$
\begin{aligned}
p(\alpha_j[d_i]|d_i) &= \frac{p(\alpha_j[d_i])p(d_i|\alpha_j[d_i])}{p(d_i)} \\
&\cong p(\alpha_j[d_i])p(d_i|\alpha_j[d_i]) \\
&\cong p(\alpha_j[d_i])p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) \\
&\qquad \times \prod{}_{\alpha \in \mathit{\Lambda} - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i])
\end{aligned}
$$

Let us assume that;

- $F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}$ be the size of the microdata set $D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d]$

$= {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] - d_i$, and

- $F_{\alpha_j[d_i]}$ be the size of the microdata set $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$.

Single replacement of a maximum impact data value causes $p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i])$ to decrease from $\frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{\alpha_j[d_i]}}$ to $\frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i]}}$. This, in turn decreases $p(d_i|\alpha_j[d_i])$ by $\frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}}$ as shown below.

$$\begin{aligned}
p'(d_i|\alpha_j[d_i]) &= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p'(\alpha[d_i]|\alpha_j[d_i]) \times p'({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \times \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i]}} \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \times \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i]}} \\
&\quad \times \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}} \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \times p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) \\
&\quad \times \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}} \\
&= p(d_i|\alpha_j[d_i]) \times \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}}
\end{aligned}$$

Now let us assume that there is another attribute α_k which decreases $p(\alpha_j[d_i]|d_i)$ more than that of ${\alpha_{MI}}^{\alpha_j[d_i]}$. This implies the following.

$$\begin{aligned}
\frac{F_{\alpha_j[d_i],\alpha_k}-1}{F_{\alpha_j[d_i],\alpha_k}} &< \frac{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1}{F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}} \\
(F_{\alpha_j[d_i],\alpha_k}-1)F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}} &< (F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}-1)F_{\alpha_j[d_i],\alpha_k} \\
F_{\alpha_j[d_i],\alpha_k} &< F_{\alpha_j[d_i],{\alpha_{MI}}^{\alpha_j[d_i]}}
\end{aligned}$$

which contradicts the definition of *Maximum Impact Attribute*. So, we can conclude that every replacement of a maximum impact data value with ν causes the highest decrease in $p(\alpha_j[d_i]|d_i)$ which in turn implies that the number of data values that should be modified is minimal□

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm verifies the need for suppression. It finds $p(v|d_i)$ for all $v \in V_{\alpha_j}$ and checks the truth value of the following assertion:

$$p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i] \tag{3.8}$$

If Assertion (3.8) is true, it picks a random next best guess v_{rnbg} from V_{α_j}. Next, in each iteration it finds the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ and replaces the maximum impact data values by ν as long as $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$. After proccessing all maximum impact attributes, it re-checks the truth value of Assertion (3.8). If Assertion (3.8) is still true, it reverts all changes and deletes the tuple d_i from the microdata set. An overview of the algorithm is provided in Figure 3.1.1.

If $|V_{\alpha_j}| = 2$ is true, then suppressing the confidential data value might result in an adversary guessing it correctly with 100% confidence. Therefore, the decision to suppress a confidential data value is randomized for the case where $|V_{\alpha_j}| = 2$. This results in an adversary guessing the actual confidential data value with 50% confidence which is the maximum uncertainty that can be achieved under such circumstances.

Lemma 3.1. *Let $\alpha_j[d_i]$ be the confidential data value, n be the number of attributes and N be the number of tuples in $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$. Then, the upper bound for the number of data values that can be modified by the DECP algorithm is equal to $(n-1)(N-1)$.*

Proof: The DECP algorithm modifies the maximum impact data values from the tuples $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] - d_i$. As $D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] \subseteq D[\alpha_j[d] = \alpha_j[d_i]]$, the number of tuples that can be modified for each maximum impact attribute is bounded by $N-1$. At each iteration, the DECP algorithm picks a different maximum impact attribute and replaces the instances of this attribute with ν. Since, there are $n-1$ different alternatives for a maximum impact attribute, we can conclude that the DECP algorithm can replace at most $(n-1)(N-1)$ data values with ν for suppressing a confidential data value □

```
INPUT:        $D$, the microdata set
              $d_i$, the tuple containing the confidential data value
              $\alpha_j$, the attribute containing the confidential data value
              $k$, degree of suppression
OUTPUT:       $D'$, the new data set
BEGIN
              Find probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j}$
              If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ {
                If $|V_{\alpha_j}| = 2$
                  Randomly decide whether or not to continue suppression
                Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d_i]}$
                While $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$ and candidates for maximum impact attribute exist {
                  Find the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$
                  Find the maximum impact data values
                  Count $= |D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]]|$
                  While $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$ and $Count > 1$ {
                    Replace the next data value in maximum impact data values with $\nu$
                    $p(\alpha_j[d_i]|d_i)* = \frac{Count-1}{Count}$
                    $Count = Count - 1$
                  }
                }
                If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ {
                  Revert all changes
                  Delete microdata tuple $d_i$
                }
                Else
                  Replace $\alpha_j[d_i]$ with $\nu$
              }
END
ATTRIBUTE     $FindMaximumImpactAttributeForDECP(D,d_i,\alpha_j)$ {
                $count = |D|$
                For each attribute $\alpha \in \Lambda - \alpha_j${
                  $count_\alpha$ = number of all tuples $d \in D - d_i$ satisfying the constraint
                            $\alpha_j[d] = \alpha_j[d_i] \wedge \alpha[d] = \alpha[d_i]$
                  If $count_\alpha < count$ and $count_\alpha > 1${
                    $count = count_\alpha$
                    ${\alpha_{MI}}^{\alpha_j[d_i]} = \alpha$
                  }
                }
                return ${\alpha_{MI}}^{\alpha_j[d_i]}$
              }
```

Figure 3.2: Pseudocode of DECP Algorithm

Example 2. Now, let us illustrate how the DECP algorithm suppresses Bob's confidential diagnosis.

Step 1. Initially, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j} = \{dyspepsia, angina\ pectoris, gastritis\}$. The Na- ïve Bayesian classification model constructed using the medical records of Table 1.2 is shown in Table 3.1. According to the model $p(dyspepsia|d_2) = 0$, $p(angina\ pectoris|d_2) = \frac{1}{6}$, and $p(gastritis|d_2) = \frac{1}{18}$.

Step 2. The probability $p(angina\ pectoris|d_2)$ is greater than both $p(dyspepsia|d_2)$ and

Table 3.1: Naïve Bayesian Classification Model Constructed Using the Medical Records Shown in Table 1.2

Diagnosis	p(**Diagnosis**)	p(**Symptom**|**Diagnosis**)			p(**Diagnosis**$\|d_2$)
		Indigestion	**Chest Pain**	**Palpitation**	
Dyspepsia	2/8	0	0	1/2	0
Gastritis	3/8	1/3	2/3	2/3	1/18
Angina Pectoris	3/8	2/3	2/3	1	1/6

$p(gastritis|d_2)$. As Bob's diagnosis can be correctly predicted, the suppression process starts.

Step 3. Let's assume that gastritis is selected as the random next best guess. From this point on the DECP algorithm will try to decrease $p(angina\ pectoris|d_2)$ below $p(gastritis|d_2)$.

Step 4. To select the maximum impact attribute, the counts for each symptom attribute is found as follows;

- $count_I = |D[diagnosis[d] = diagnosis[d_2] \wedge indigestion[d] = indigestion[d_2]]| = 2$
- $count_{CP} = |D[diagnosis[d] = diagnosis[d_2] \wedge chestpain[d] = chestpain[d_2]]| = 2$
- $count_P = |D[diagnosis[d] = diagnosis[d_2] \wedge palpitation[d] = palpitation[d_2]]| = 3$

Both indigestion and chest pain attributes have the minimum count. Therefore, they are the candidates for the maximum impact attribute. Let's assume that indigestion is selected as the maximum impact attribute.

Step 5. All tuples d satisfying the constraint $indigestion\ [d] = N \wedge diagnosis[d] = angina\ pectoris$ are found. Tuples 7 and 8 satisfy the mentioned constraint.

Step 6. The indigestion attribute is hidden from tuple 7. With this replacement $p(angina\ pectoris|d_2)$ decreases by $\frac{1}{2}$ to $\frac{1}{12}$. As $p(angina\ pectoris|d_2)$ is still greater than $p(gastritis|d_2)$, the suppression process continues with the next maximum impact attribute which is chest pain.

Table 3.2: Academic Health Medical Records After DECP Execution

Zipcode	**Gender**	**Age**	**Indigestion**	**Chest Pain**	**Palpitation**	**Diagnosis**
90302	Female	29	Y	N	Y	Dyspepsia
90410	Male	22	N	Y	Y	?
90301	Male	27	Y	N	N	Dyspepsia
90310	Female	43	Y	N	N	Gastritis
90301	Male	52	N	Y	Y	Gastritis
90410	Male	47	Y	?	Y	Angina Pectoris
90305	Female	30	?	N	Y	Angina Pectoris
90402	Male	36	N	Y	Y	Angina Pectoris
90301	Male	52	Y	Y	Y	Gastritis

Step 7. All tuples d satisfying the constraint $chestpain[d] = Y \wedge diagnosis[d] = angina\ pectoris$ are found. Tuples 6 and 8 satisfy the mentioned constraint.

Step 8. The chest pain attribute is hidden from tuple 6. With this replacement $p(angina\ pectoris|d_2)$ decreases by $\frac{1}{2}$ to $\frac{1}{24}$. As $p(angina\ pectoris|d_2)$ is smaller than $p(gastritis|d_2)$, the suppression process stops. The resulting microdata set can be seen in Table 3.2.

3.1.2 INCP Algorithm

The INCP algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that it cannot be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D'-d_i,\alpha_j}$. It accomplishes its goal, as its name implies, by increasing the probabilities $p(v|d_i)$ for all v in the next best guess set,S_{nbg}, above $p(\alpha_j[d_i] \mid d_i)$.

For each $v \in S_{nbg}$, the INCP algorithm identifies the tuples $d \in D[\alpha_j[d] = v]$ having no common attribute value with d_i and modifies them by replacing $\alpha_j[d]$ with ν in order to increase $p(v|d_i)$.

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm verifies the need for suppression. It finds $p(v|d_i)$ for all $v \in V_{\alpha_j}$ and checks the truth value of Assertion (3.8).If Assertion (3.8) is true, it picks a random next

```
INPUT:    $D$, the microdata set
          $d_i$, the tuple containing the confidential data value
          $\alpha_j$, the attribute containing the confidential data value
          $k$, degree of suppression
OUTPUT:   $D'$, the new data set
BEGIN
          Find probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j}$
          If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ {
            If $|V_{\alpha_j}| = 2$
              Randomly decide whether or not to continue suppression
            Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d_i]}$
            $S_{nbg}$ = All attribute values $v \in V_{\alpha_j}$ satisfying $p(v|d_i) \geq p(v_{rnbg}|d_i)$
            For each $v \in S_{nbg}$ {
              While $p(\alpha_j[d_i]|d_i) > p(v|d_i)$ and $D[\alpha_j[d] = v] \neq empty$ {
                $T$ = next tuple in $D[\alpha_j[d] = v]$
                If $T \cap d_i = empty$ {
                  Replace $\alpha_j[T]$ with $\nu$
                  Recalculate probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j}$
                }
              }
            }
            If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$
              Run algorithm DECP
            Else
              Replace $\alpha_j[d_i]$ with $\nu$
          }
END
```

Figure 3.3: Pseudocode of INCP Algorithm

best guess v_{rnbg} from V_{α_j} and forms S_{nbg} by finding the attribute values $v \in V_{\alpha_j}$ satisfying $p(v|d_i) \geq p(v_{rnbg}|d_i)$. Next, for each $v \in S_{nbg}$, the algorithm finds the tuples $d \in D[\neg\ \alpha_1[d] = \alpha_1[d_i] \wedge \ldots \wedge \neg\ \alpha_{j-1}[d] = \alpha_{j-1}[d_i] \wedge \alpha_j[d] = v \wedge \neg\ \alpha_{j+1}[d] = \alpha_{j+1}[d_i] \wedge \ldots \wedge \neg\ \alpha_n[d] = \alpha_n[d_i]]$ and modifies them by replacing $\alpha_j[d]$ with ν until the goal is achieved, that is until $p(v|d_i)$ becomes less than or equal to $p(\alpha_j[d_i]|d_i)$. After proccessing all attribute values $v \in S_{nbg}$, it re-checks the truth value of Assertion (3.8). If Assertion (3.8) is still true, then DECP algorithm is executed to complete the algorithm. An overview of the algorithm is provided in Figure 3.1.2.

Lemma 3.2. *Let $\alpha_j[d_i]$ be the confidential data value, m be the number of tuples in D and N be the number of tuples in $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$. Assuming that there are enough number of tuples that can be used for the suppression process (i.e. no need for executing DECP), the upper bound for the number of data values that can be modified by the INCP algorithm is equal to $m - N - 1 - |S_{nbg}|$.*

Proof: The INCP algorithm modifies the tuples $d \in D[\neg\ \alpha_1[d] = \alpha_1[d_i] \wedge \ldots \wedge$

$\neg\ \alpha_{j-1}[d] = \alpha_{j-1}[d_i] \wedge \alpha_j[d] = v \wedge \neg\ \alpha_{j+1}[d] = \alpha_{j+1}[d_i] \wedge \ldots \wedge \neg\ \alpha_n[d] = \alpha_n[d_i]]$ for each $v \in S_{nbg}$. In the worst case, S_{nbg} contains all possible values of attribute α_j except $\alpha_j[d_i]$. This implies $\sum_{v \in S_{nbg}} |D[\alpha_j[d] = v]| = m - N - 1$. Moreover, due to the definition of next best guess set and random next best guess the probability $p(v|d_i)$ for each $v \in S_{nbg}$ must be greater than zero. This implies that, in the worst case there exists at least one tuple which has the same data values with d_i (except α_j) for each $v \in S_{nbg}$. So, we can conclude that the INCP algorithm can replace at most $m - N - 1 - |S_{nbg}|$ data values with ν for suppressing a confidential data value □

Example 3. Now, let us illustrate how the INCP algorithm suppresses Bob's confidential diagnosis.

Step 1. Initially, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j} = \{dyspepsia, angina\ pectoris, gastritis\}$. The Naïve Bayesian classification model constructed using the medical records of Table 1.2 is shown in Table 3.1. According to the model the probabilities are $p(dyspepsia|d_2) = 0$, $p(angina\ pectoris|d_2) = \frac{1}{6}$, and $p(gastritis|d_2) = \frac{1}{18}$.

Step 2. The probability $p(angina\ pectoris|d_2)$ is greater than both $p(dyspepsia|d_2)$ and $p(gastritis|d_2)$. As Bob's diagnosis can be correctly predicted, the suppression process starts.

Step 3. Let's assume that gastritis is selected as the random next best guess. From this point on, the INCP algorithm will try to increase $p(gastritis|d_2)$ above $p(anginapectoris|d_2)$.

Step 4. All tuples d which has no common symptoms with Bob among $D[diagnosis[d] = gastritis]$ is found. Tuple 4 satisfies the mentioned constraint.

Step 5. The diagnosis attribute is hidden from tuple 4. After this replacement, $p(gastritis|d_2)$ increases to $\frac{1}{7}$, and $p(angina\ pectoris|d_2)$ increases to $\frac{4}{21}$. As $p(angina\ pectoris|d_2)$ is still greater than $p(gastritis|d_2)$, the suppression process continues.

Step 6. Since there are no more tuples which has no common symptoms with Bob

Table 3.3: Academic Health Medical Records After INCP Execution

Zipcode	**Gender**	**Age**	**Indigestion**	**Chest Pain**	**Palpitation**	**Diagnosis**
90302	Female	29	Y	N	Y	Dyspepsia
90410	Male	22	N	Y	Y	?
90301	Male	27	Y	N	N	Dyspepsia
90310	Female	43	Y	N	N	?
90301	Male	52	N	Y	Y	Gastritis
90410	Male	47	Y	Y	Y	Angina Pectoris
90305	Female	30	?	N	Y	Angina Pectoris
90402	Male	36	N	Y	Y	Angina Pectoris
90301	Male	52	Y	Y	Y	Gastritis

among $D[diagnosis[d] = gastritis]$, the suppression process continues with the DECP execution.

Step 7. To select the maximum impact attribute, the counts for each symptom attribute is found as follows;

- $count_I = |D[diagnosis[d] = diagnosis[d_2] \wedge indigestion[d] = indigestion[d_2]]| = 2$
- $count_{CP} = |D[diagnosis[d] = diagnosis[d_2] \wedge chestpain[d] = chestpain[d_2]]| = 2$
- $count_P = |D[diagnosis[d] = diagnosis[d_2] \wedge palpitation[d] = palpitation[d_2]]| = 3$

Both indigestion and chest pain attributes have the minimum count. Therefore, they are the candidates for the maximum impact attribute. Let's assume that indigestion is selected as the maximum impact attribute.

Step 8. All tuples d satisfying the constraint $indigestion\ [d] = N \wedge diagnosis[d] = angina\ pectoris$ are found. Tuples 7 and 8 satisfy the mentioned constraint.

Step 9. The indigestion attribute is hidden from tuple 7. With this replacement $p(angina\ pectoris|d_2)$ decreases by $\frac{1}{2}$ to $\frac{2}{21}$. As $p(angina\ pectoris|d_2)$ is smaller than $p(gastritis|d_2)$, the suppression process stops. The resulting microdata can be seen in Table 3.3.

3.1.3 DROPP Algorithm

The DROPP algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that it cannot be correctly predicted by the classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$. It aims at dropping the probability $p(\alpha_j[d_i]|d_i)$ below that of the random next best guess v_{rnbg}, so that it cannot be correctly predicted by the classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$. Unlike DECP and INCP algorithms, it achieves its goal by downgrading the tuple d_i, instead of downgrading classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$.

The algorithm employs the following modified definition of *Maximum Impact Attribute.*

Definition 3.6. Maximum Impact Attribute. *The attribute with maximum impact on $p(\alpha_j[d_i]|d_i)$, denoted by ${\alpha_{MI}}^{\alpha_j[d_i]}$, is the one that satisfies the following conditions.*

$$
{\alpha_{MI}}^{\alpha_j[d_i]} = \arg\max{}_{\alpha\in A}\left(\frac{|D[\alpha_j[d]=\alpha_j[d_i]\wedge\alpha[d]=\alpha[d_i]]-d_i|}{|D[\alpha_j[d]=v_{rnbg}\wedge\alpha[d]=\alpha[d_i]]|}\right)
$$
$$
\wedge\ p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) > p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|v_{rnbg}) \qquad (3.9)
$$

Definition 3.7. Maximum Impact Data Value. *The maximum impact data value is the instance of maximum impact data attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ in tuple d_i.*

It must be noted that, the maximum impact data values have a higher probability of occurrence in tuples $d\in D[\alpha_j[d]=\alpha_j[d_i]]-d_i$ than that of tuples $d\in D[\alpha_j[d]=v_{rnbg}]$. Therefore, they are the key to decrease $p(\alpha_j[d_i]|d_i)$ below $p(v_{rnbg}|d_i)$.

In each iteration, the DROPP algorithm identifies ${\alpha_{MI}}^{\alpha_j[d_i]}$ and modifies the tuple d_i by replacing ${\alpha_{MI}}^{\alpha_j[d_i]}[d_i]$ with ν until the goal is achieved, that is until $p(\alpha_j[d_i]|d_i)$ becomes less than $p(v_{rnbg}|d_i)$. Each such replacement results in the maximum possible reduction in $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$, thus requiring less number of modifications.

Theorem 3.2. *Let ${\alpha_{MI}}^{\alpha_j[d_i]}$ be the maximum impact attribute satisfying Equation (3.9). Then, every replacement of a maximum impact data value with ν causes the maximum decrease in $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$, thus resulting in fewer data values to be modified.*

Proof: Let us first find the effect of replacing a maximum impact data value with ν on $p(\alpha_j[d_i]|d_i)$ and $p(v_{rnbg}|d_i)$. Remember that, since $p(d_i)$ is same for all $v \in V_{\alpha_j}$, it can be ignored when calculating $p(\alpha_j[d_i]|d_i)$.

$$\begin{aligned} p(\alpha_j[d_i]|d_i) &= \frac{p(\alpha_j[d_i])p(d_i|\alpha_j[d_i])}{p(d_i)} \\ &\cong p(\alpha_j[d_i])p(d_i|\alpha_j[d_i]) \\ &\cong p(\alpha_j[d_i])p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) \\ &\quad \times \prod_{\alpha \in \Lambda - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \end{aligned}$$

Similarly,

$$\begin{aligned} p(v_{rnbg}|d_i) &= \frac{p(v_{rnbg})p(d_i|v_{rnbg})}{p(d_i)} \\ &\cong p(v_{rnbg})p(d_i|v_{rnbg}) \\ &\cong p(v_{rnbg})p({\alpha_{MI}}^{v_{rnbg}}[d_i]|v_{rnbg}) \\ &\quad \times \prod_{\alpha \in \Lambda - \left\{\alpha_j, {\alpha_{MI}}^{v_{rnbg}}\right\}} p(\alpha[d_i]|v_{rnbg}) \end{aligned}$$

Let the size of the microdata set $D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] - d_i$ be $F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}$ and the size of the microdata set $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$ be $F_{\alpha_j[d_i]}$. Let the size of the microdata set $D[\alpha_j[d] = v_{rnbg} \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]]$ be $F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}$ and the size of the microdata set $D[\alpha_j[d] = v_{rnbg}]$ be $F_{v_{rnbg}}$. Replacement of the maximum impact data value causes $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ to decrease by $\frac{F_{v_{rnbg}}}{F_{\alpha_j[d_i]}} \times \frac{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}}$ as shown below.

$$\begin{aligned} p'(\alpha_j[d_i]|d_i) &\cong p'(\alpha_j[d_i])p'(d_i|\alpha_j[d_i]) \\ &\cong p(\alpha_j[d_i]) \prod_{\alpha \in \Lambda - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \\ &\cong p(\alpha_j[d_i]|d_i) \frac{F_{\alpha_j[d_i]}}{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}} \end{aligned}$$

$$
\begin{aligned}
p'(v_{rnbg}|d_i) &\cong p'(v_{rnbg})p'(d_i|v_{rnbg}) \\
&\cong p(v_{rnbg}) \prod\nolimits_{\alpha \in \mathit{\Lambda} - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|v_{rnbg}) \\
&\cong p(v_{rnbg}|d_i) \frac{F_{v_{rnbg}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}} \\
\frac{p'(\alpha_j[d_i]|d_i)}{p'(v_{rnbg}|d_i)} &= \frac{p(\alpha_j[d_i]|d_i) \frac{F_{\alpha_j[d_i]}}{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}}}{p(v_{rnbg}|d_i) \frac{F_{v_{rnbg}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}}} \\
&= \frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)} \frac{F_{\alpha_j[d_i]}}{F_{v_{rnbg}}} \frac{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}} \\
\frac{\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}}{\frac{p'(\alpha_j[d_i]|d_i)}{p'(v_{rnbg}|d_i)}} &= \frac{F_{v_{rnbg}}}{F_{\alpha_j[d_i]}} \frac{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}}
\end{aligned}
$$

Now let us assume that there is another attribute α_k which decreases $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ more than that of ${\alpha_{MI}}^{\alpha_j[d_i]}$. This implies the following:

$$
\begin{aligned}
\frac{F_{v_{rnbg}}}{F_{\alpha_j[d_i]}} \frac{F_{\alpha_j[d_i], \alpha_k}}{F_{v_{rnbg}, \alpha_k}} &> \frac{F_{v_{rnbg}}}{F_{\alpha_j[d_i]}} \frac{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}} \\
\frac{F_{\alpha_j[d_i], \alpha_k}}{F_{v_{rnbg}, \alpha_k}} &> \frac{F_{\alpha_j[d_i], {\alpha_{MI}}^{\alpha_j[d_i]}}}{F_{v_{rnbg}, {\alpha_{MI}}^{\alpha_j[d_i]}}}
\end{aligned}
$$

However, this contradicts the definition of *Maximum Impact Attribute*. So, we can conclude that every replacement of a maximum impact data value with ν causes the highest decrease in $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ which in turn implies that the number of data values that should be modified is minimal□

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm verifies the need for suppression. It finds $p(v|d_i)$ for all $v \in V_{\alpha_j}$ and checks the truth value of Assertion (3.8). If Assertion (3.8) is true, it picks a random next best guess v_{rnbg} from V_{α_j}. Next, in each iteration it finds the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ and replaces the maximum impact data value ${\alpha_{MI}}^{\alpha_j[d_i]}[d_i]$ by ν. After each iteration, it re-checks the truth value of Assertion (3.8) to decide whether to continue execution. If Assertion (3.8) is still true after all possible maximum impact attributes are processed, it reverts all changes and deletes the tuple d_i from the microdata set.

```
INPUT:      $D$, the microdata set
            $d_i$, the tuple containing the confidential data value
            $\alpha_j$, the attribute containing the confidential data value
            $k$, degree of suppression
OUTPUT:     $D'$, the new data set
BEGIN
            Find probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j}$
            If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ {
              If $|V_{\alpha_j}| = 2$
                Randomly decide whether or not to continue suppression
              Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d_i]}$
              While $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$ and candidates for maximum impact attribute exist {
                Find the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$
                Replace the maximum impact data value ${\alpha_{MI}}^{\alpha_j[d_i]}[d_i]$ with $\nu$
                Recalculate probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j}$
              }
              If $p(\alpha_j[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ {
                Revert all changes
                Delete microdata tuple $d_i$
              }
              Else
                Replace $\alpha_j[d_i]$ with $\nu$
            }
END
ATTRIBUTE   $FindMaximumImpactAttributeForDROPP(D, d_i, \alpha_j, v_{rnbg})$ {
              $ratio = 0$
              $count_{\alpha_j[d_i]}$ = number of all tuples $d \in D - d_i$ satisfying the constraint $\alpha_j[d] = \alpha_j[d_i]$
              $count_{v_{rnbg}}$ = number of all tuples $d \in D$ satisfying the constraint $\alpha_j[d] = v_{rnbg}$
              For each attribute $\alpha \in \Lambda - \alpha_j$ {
                ${count_\alpha}^{\alpha_j[d_i]}$ = number of all tuples $d \in D - d_i$ satisfying the constraint
                                  $\alpha_j[d] = \alpha_j[d_i] \wedge \alpha[d] = \alpha[d_i]$
                ${count_\alpha}^{v_{rnbg}}$ = number of all tuples $d \in D$ satisfying the constraint $\alpha_j[d] = v_{rnbg} \wedge \alpha[d] = \alpha[d_i]$
                $p(\alpha[d_i]|\alpha_j[d_i]) = \frac{{count_\alpha}^{\alpha_j[d_i]}}{count_{\alpha_j[d_i]}}$, $p(\alpha[d_i]|v_{rnbg}) = \frac{{count_\alpha}^{v_{rnbg}}}{count_{v_{rnbg}}}$
                If ${count_\alpha}^{\alpha_j[d_i]}/{count_\alpha}^{v_{rnbg}} > ratio$ and $p(\alpha[d_i]|\alpha_j[d_i]) > p(\alpha[d_i]|v_{rnbg})$ {
                  $ratio = {count_\alpha}^{\alpha_j[d_i]}/{count_\alpha}^{v_{rnbg}}$
                  ${\alpha_{MI}}^{\alpha_j[d_i]} = \alpha$
                }
              }
              return ${\alpha_{MI}}^{\alpha_j[d_i]}$
            }
```

Figure 3.4: Pseudocode of DROPP Algorithm

An overview of the algorithm is provided in Figure 3.4.

Lemma 3.3. *Let $\alpha_j[d_i]$ be the confidential data value and n be the number of attributes. Then, the upper bound for the number of data values that can be modified by the DROPP algorithm is equal to $n - 1$.*

Proof: The DROPP algorithm modifies only tuple d_i which has $n - 1$ data values excluding the confidential data value. So, we can conclude that the DROPP algorithm can replace at most $n - 1$ data values with ν for suppressing a confidential data value □

Example 4. Now, let us illustrate how the DROPP algorithm suppresses Bob's confidential diagnosis.

Step 1. Initially, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{\alpha_j} = \{dyspepsia, angina\ pectoris, gastritis\}$. The Naïve Bayesian classification model constructed using the medical records of Table 1.2 is shown in Table 3.1. According to the model the probabilities are $p(dyspepsia|d_2) = 0$, $p(angina\ pectoris|d_2) = \frac{1}{6}$, and $p(gastritis|d_2) = \frac{1}{18}$.

Step 2. The probability $p(angina\ pectoris|d_2)$ is greater than both $p(dyspepsia|d_2)$ and $p(gastritis|d_2)$. As Bob's diagnosis can be correctly predicted, the suppression process starts.

Step 3. Let's assume that gastritis is selected as the random next best guess. From this point on, the DROPP algorithm will try to drop $p(angina\ pectoris|d_2)$ below $p(gastritis|d_2)$.

Step 4. To select the maximum impact attribute, the following counts and ratios are found;

- ${count_I}^{AP} = |D[diagnosis[d] = angina\ pectoris \wedge indigestion[d] = indigestion[d_2]]| = 2$
- ${count_I}^{G} = |D[diagnosis[d] = gastritis \wedge\ indigestion[d] = indigestion[d_2]]| = 1$
- ${count_{CP}}^{AP} = |D[diagnosis[d] = angina\ pectoris \wedge\ chest\ pain[d] = chest\ pain[d_2]]| = 2$
- ${count_{CP}}^{G} = |D[diagnosis[d] = gastritis \wedge\ chest\ pain[d] = chest\ pain[d_2]]| = 2$
- ${count_P}^{AP} = |D[diagnosis[d] = angina\ pectoris \wedge\ palpitation[d] = palpitation[d_2]]| = 3$
- ${count_P}^{G} = |D[diagnosis[d] = gastritis \wedge\ palpitation[d] = palpitation[d_2]]| = 2$
- $ratio_I = 2$
- $ratio_{CP} = 1$
- $ratio_P = 3/2$

Table 3.4: Academic Health Medical Records After DROPP Execution

Zipcode	Gender	Age	Indigestion	Chest Pain	Palpitation	Diagnosis
90302	Female	29	Y	N	Y	Dyspepsia
90410	Male	22	?	Y	?	?
90301	Male	27	Y	N	N	Dyspepsia
90310	Female	43	Y	N	N	Gastritis
90301	Male	52	N	Y	Y	Gastritis
90410	Male	47	Y	Y	Y	Angina Pectoris
90305	Female	30	N	N	Y	Angina Pectoris
90402	Male	36	N	Y	Y	Angina Pectoris
90301	Male	52	Y	Y	Y	Gastritis

Indigestion has the maximum ratio. Therefore, it is selected as the maximum impact attribute.

Step 5. The indigestion attribute is hidden from tuple 2. With this replacement $p(angina\ pectoris|d_2)$ increases to $\frac{1}{4}$, and $p(gastritis|d_2)$ increases to $\frac{1}{6}$. As $p(angina\ pectoris|d_2)$ is still greater than $p(gastritis|d_2)$, the suppression process continues with the next maximum impact attribute which is palpitation.

Step 6. The palpitation attribute is hidden from tuple 2. With this replacement $p(angina\ pectoris|d_2)$ remains the same, but $p(gastritis|d_2)$ increases to $\frac{1}{4}$. As $p(angina\ pectoris|d_2)$ is equal to $p(gastritis|d_2)$, the suppression process stops. The resulting microdata set can be seen in Table 3.4.

3.2 Suppression Against Decision Tree Classification Models

In the following, we present the HID3 algorithm for preventing decision tree classification based inference using deletion. Although we have used ID3 in our experiments, the proposed algorithm can be used to suppress a confidential data value from any decision tree algorithm.

3.2.1 HID3 Algorithm

The HID3 algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that the ID3 classifier ${\varsigma_{id3}}^{D-d_i,\alpha_j}$ cannot correctly predict its actual value. Similar to the DROPP algorithm, it achieves its goal by downgrading the microdata tuple d_i containing the confidential data value.

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm builds the decision tree using $D - d_i$ and verifies the need for suppression. If ${\varsigma_{id3}}^{D-d_i,\alpha_j}$ can correctly predict the confidential data value, it calls the recursive *ID3Hide* function. Then, the *ID3Hide* function checks whether the root node is a leaf or not. If it is a leaf and its value is different from the confidential data value $\alpha_j[d_i]$ it returns *true*, which in turn terminates the recursive function successfully. Or else, it returns *false*. If the root node is not a leaf, then it finds the most probable value $v_\pi \in V_{\alpha_j}$ for $\alpha_j[d_i]$, and checks whether v_π is equal to $\alpha_j[d_i]$ or not. If the most probable value v_π is not equal to the actual confidential data value $\alpha_j[d_i]$ it returns *true*. Otherwise, it further explores the child nodes of the root in order to suppress $\alpha_j[d_i]$. Let the decision attribute of the root node be α_{root}, the most common child of the root (i.e. the child with highest training population) be $child_{MC}$ and the child containing $\alpha_{root}[d_i]$ be $child_{Match}$. If $\alpha_{root}[d_i] = \nu$ or $child_{Match} = child_{MC}$ it tries to suppress the confidential data value using $child_{MC}$. Or else, it uses $child_{Match}$ for suppression. After exploring all possible sub-branches, if the algorithm fails to suppress the confidential data value, it reverts all changes and deletes the tuple d_i from the microdata set. An overview of the algorithm is provided in Figure 3.5.

If $|V_{\alpha_j}| = 2$ is true, then suppressing the confidential data value might result in an adversary guessing it correctly with 100% confidence. Therefore, the decision to suppress a confidential data value is randomized for the case where $|V_{\alpha_j}| = 2$. This results in an adversary guessing the actual confidential data value with 50% confidence which is the maximum uncertainty that can be achieved under such circumstances.

Lemma 3.4. *Let $\alpha_j[d_i]$ be the confidential data value and n be the number of attributes.*

```
INPUT:     $D$, the microdata set
           $d_i$, the tuple containing the confidential data value
           $\alpha_j$, the attribute containing the confidential data value
OUTPUT:    $D'$, the new data set
BEGIN
           Replace the confidential data value $\alpha_j[d_i]$ with $\nu$
           Build the decision tree using ID3
           If $v_\pi = \alpha_j[d_i]$ {
             If $|V_{\alpha_j}| = 2$
               Randomly decide whether or not to continue suppression
             If $ID3Hide(root$ of the decision tree$) == false$
               Delete microdata tuple $d_i$
           }
END
BOOL       $ID3Hide(root)$ {
           If $root$ is a leaf return $root.value \neq \alpha_j[d_i]$
           If $v_\pi \neq \alpha_j[d_i]$ return $true$
           $\alpha_{root}$ = decision attribute of the $root$
           $child_{MC}$ = most common child of the $root$
           $child_{Match}$ = child containing the value $\alpha_{root}[d_i]$
           If $\alpha_{root}[d_i] = \nu$ return $ID3Hide(child_{MC})$
           Else If $child_{MC} = child_{Match}$ {
             Replace $\alpha_{root}[d_i]$ with $\nu$
             If $ID3Hide(child_{MC})$ return $true$
             Else {
               Revert changes to $\alpha_{root}[d_i]$
               return $false$
             }
           }
           Else If $ID3Hide(child_{Match})$ return $true$
           Else {
             Replace $\alpha_{root}[d_i]$ with $\nu$
             If $ID3Hide(child_{MC})$ return $true$
             Else {
               Revert changes to $\alpha_{root}[d_i]$
               return $false$
             }
           }
```

Figure 3.5: Pseudocode of HID3 Algorithm

Then, the upper bound for the number of data values that can be modified by the HID3 algorithm is equal to $n - 1$.

Proof: The HID3 algorithm modifies only tuple d_i which has $n-1$ data values excluding the confidential data value. So, we can conclude that the HID3 algorithm can replace at most $n - 1$ data values with ν for suppressing a confidential data value □

Example 5. For this specific example, let us assume Bob *does not have any chest pain* and illustrate how the HID3 algorithm suppresses his confidential diagnosis.

Step 1. Initially, the ID3 classification model shown in Figure 5 is constructed based on the medical records shown in Table 1.2.

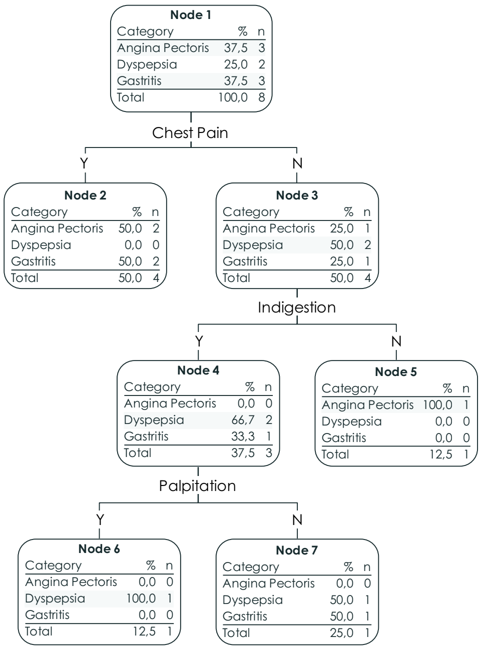


Figure 3.6: Decision Tree Constructed Using the Medical Records Shown in Table 1.2

Step 2. Starting from the *root=node 1*, the ID3Hide function checks whether it is possible to correctly predict Bob's diagnosis. Since Bob's diagnosis can be correctly predicted using the path $chest\ pain = N \wedge indigestion = N$, the suppression process starts.

Step 3. Using the whole microdata the ID3Hide function checks whether the majority of the tuples have $chest\ pain = N$. Since, the number of tuples having $chest\ pain = N$ is equal to the number of tuples having $chest\ pain = Y$, the function calls itself with *root=node 3*.

Step 4. Starting from the subtree *root=node 3*, the ID3Hide function checks whether it is possible to correctly predict Bob's diagnosis. Since Bob's diagnosis can be correctly predicted using the path $indigestion = N$, the suppression process continues.

Step 5. Using only the tuples with $chest\ pain = N$, the ID3Hide function checks whether the majority of the tuples have $indigestion = N$. Since, the number of tuples having $indigestion = N$ is smaller than the number of tuples having $indigestion = Y$,

Table 3.5: Academic Health Medical Records After HID3 Execution

Zipcode	**Gender**	**Age**	**Indigestion**	**Chest Pain**	**Palpitation**	**Diagnosis**
90302	Female	29	Y	N	Y	Dyspepsia
90410	Male	22	?	Y	Y	?
90301	Male	27	Y	N	N	Dyspepsia
90310	Female	43	Y	N	N	Gastritis
90301	Male	52	N	Y	Y	Gastritis
90410	Male	47	Y	Y	Y	Angina Pectoris
90305	Female	30	N	N	Y	Angina Pectoris
90402	Male	36	N	Y	Y	Angina Pectoris
90301	Male	52	Y	Y	Y	Gastritis

the function calls itself with *root=node 5*.

Step 6. As node 5 is a leaf, the ID3Hide function checks whether the most probable value, i.e. angina pectoris, and the confidential diagnosis are equal or not. As they are equal, the function returns from the recursive call signaling an unsuccessful run.

Step 7. As the recursive call to ID3Hide was unsuccessful, the current node's attribute, i.e. the indigestion attribute, is hidden from Bob's tuple. Next, the function calls itself with *root=node 4*, as tuples with $indigestion = Y$ constitute the majority among tuples with $chest\ pain = N$.

Step 8. Starting from the subtree *root=node 4*, the ID3Hide function checks whether it is possible to correctly predict Bob's diagnosis. Since Bob's diagnosis cannot be correctly predicted using the path $palpitation = Y$, the suppression process stops. The resulting microdata set can be seen in Table 3.5.

3.3 Suppression of Multiple Confidential Data Values

In the following, we present the enhanced versions of DECP and DROPP algorithms for preventing probabilistic classification based inference. The proposed algorithms aim to reduce to side-effects while suppressing multiple confidential data values.

3.3.1 e-DECP Algorithm

The enhanced DECP algorithm aims at suppressing multiple confidential data values so that none of them can be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D',\alpha_j}$. The proposed algorithm reduces the side-effects of the original DECP algorithm when (1) *all confidential data values belong to a single attribute*, and (2) all confidential data values have the same value.

The algorithm works as follows: Let α_j be the confidential attribute, $S \subset D$ be the set of tuples for which α_j, satisfying the constraint $\alpha_j[d] = conf\_value$ for all $d \in S$,is confidential. As the first step, the algorithm replaces all confidential data values with ν. Then, it identifies the candidate maximum impact data values, and initializes their primary and secondary impacts. The primary impact is the number of tuples which will be affected (i.e. the probabilities will be affected) if an instance of the maximum impact data value is replaced with ν. The secondary impact, on the other hand is the number of tuples that support both the confidential data value (i.e. $\alpha_j = conf\_value$) and maximum impact data value. Next, for each tuple $d \in S$, the need for suppression is verified by finding $p(v|d)$ for all $v \in V_{\alpha_j}$ and checking the truth value of the following assertion:

$$p(\alpha_j[d]|d) > p(v|d)|\forall v \in V_{\alpha_j} - \alpha_j[d] \tag{3.10}$$

If Assertion (3.10) is true for a tuple $d \in S$, it picks a random next best guess ${v_{rnbg}}^d$, from V_{α_j}. Next, the candidate maximum impact data values are sorted. Different from the original DECP, which uses only the secondary impact to determine which maximum impact data value to use, e-DECP also uses the primary impact in order to guarantee suppression of maximum number of confidential data values with a single iteration. With maximum impact values sorted, the rest of the execution is quite similar to the original DECP which involves replacement of maximum impact data value instances, re-calculation of probabilities and re-checking of Assertion (3.10). An overview of the

```
INPUT:    $D$, the microdata set
          $\alpha_j$, the attribute containing the confidential data values
          $S$, the set of tuples for which $\alpha_j$ is confidential
          $conf\_value$, the value of confidential attribute $\alpha_j \in S$
OUTPUT:   $D'$, the new data set
BEGIN
          For each tuple $d \in S$
            Replace the confidential data value of $d$ with $\nu$
          For each attribute $\alpha \in \Lambda - \alpha_j$ {
            For each possible value of $v_\alpha \in V_\alpha$ {
              Create the maximum impact data value candidate $MIV[\alpha][v_\alpha]$
              Set $MIV[\alpha][v_\alpha].primary\_impact$ to 0
              Set $MIV[\alpha][v_\alpha].secondary\_impact$ to $|D[\alpha_j[d] = conf\_value \wedge \alpha[d] = v_\alpha]|$
            }
          For each tuple $d \in S$ {
            Find probabilities $p(v|d)$ for all $v \in V_{\alpha_j}$
            If Not $(p(\alpha_j[d]|d) > p(v|d)|\forall v \in V_{\alpha_j} - \alpha_j[d])$ Remove $d$ from $S$
            Else If $|V_{\alpha_j}| = 2$
              Randomly decide whether to suppress the confidential data value and remove $d$ from $S$
                 if decision = 'not suppress'
            If $d \in S$ {
              Pick a random next best guess ${v_{rnbg}}^d$ among ${\Omega_k}^{\alpha_j[d]}$
              For each attribute $\alpha \in \Lambda - \alpha_j$
                Increment $MIV[\alpha][\alpha[d]].primary\_impact$ by 1
            }
          }
          Sort $MIV$ first by $primary\_impact$ in descending order, then by $secondary\_impact$ in ascending order
          For each maximum impact value $miv \in MIV$ {
            While $|S| > 0$ and $miv.secondary\_impact > 1$ {
              Replace the next instance of $miv$ with $\nu$
              $miv.secondary\_impact - -$
              For each tuple $d \in S$ {
                Update $p(\alpha_j[d]|d)$
                If $p(\alpha_j[d]|d) \leq p({v_{rnbg}}^d|d)$ Remove $d$ from $S$
              }
            }
            If $|S| = 0$ break
          }
END
```

Figure 3.7: Pseudocode of e-DECP Algorithm

algorithm is provided in Figure 3.7.

3.3.2 e-DROPP Algorithm

The enhanced DROPP algorithm aims at suppressing multiple confidential data values so that none of them can be correctly predicted by the corresponding classification models ${\varsigma_{nb}}^{D,\alpha}$. The proposed algorithm reduces the side-effects of the original DROPP algorithm when *all confidential data values belong to a single tuple*.

The algorithm works as follows: Let d_i be the tuple containing all confidential data values, and S be the set of attributes containing a confidential data value in d_i. As the first step, the algorithm verifies the need for suppression for each confidential data value. More specifically, for each $\alpha \in S$, it finds $p(v|d_i)$ where $v \in V_\alpha$ and checks the truth value of the following assertion:

$$p(\alpha[d_i]|d_i) > p(v|d_i)|\forall v \in V_\alpha - \alpha[d_i] \tag{3.11}$$

If Assertion (3.11) is true, it picks a random next best guess ${v_{rnbg}}^{v_\alpha}$ from V_α. Next, it identifies the candidate maximum impact data values, and initializes their impacts on each confidential value. To identify the maximum impact data value in each iteration, the impacts of candidates are averaged and sorted. With maximum impact values sorted, the rest of the execution is quite similar to the original DROPP which involves replacement of maximum impact data value instances from d_i, re-calculation of probabilities and re-checking of Assertion (3.11). An overview of the algorithm is provided in Figure 3.8.

3.4 Discussion on the Effectiveness of Proposed Algorithms

The motivation of the suppression algorithms presented in this section is to make a given set of confidential data values non-discoverable, while minimizing the effect on the usefulness of the data for purposes other than predicting the confidential data values. But how can we make sure that an adversary would not be able to predict the suppressed confidential data values? Certainly this might be a problem if randomization is not employed in various stages of the algorithms. Let us assume that an adversary knows not only D', the transformed microdata set, but also V_{α_j}, the domain of the confidential data value $\alpha_j[d_i]$, and analyze how randomization avoids prediction of the actual confidential data value. First, let us assume that a modified version of DECP is used in order to suppress the confidential data value $\alpha_j[d_i]$. This version of DECP

```
INPUT:     $D$, the microdata set
           $d_i$, the tuple containing the confidential data values
           $S$, the set of attributes containing a confidential data value in $d_i$
OUTPUT:    $D'$, the new data set
BEGIN
           For each attribute $\alpha \in S$ {
             Replace the confidential data value in $\alpha[d_i]$ with $\nu$
             Find probabilities $p(v|d_i)$ for all $v \in V_\alpha$
             If Not $p(\alpha[d_i]|d_i) > p(v|d_i)|\forall v \in V_{\alpha_j} - \alpha_j[d_i]$ Remove $\alpha$ from $S$
             Else If $|V_\alpha| = 2$
               Randomly decide whether to suppress the confidential data value and remove $\alpha$ from $S$
                 if decision = 'not suppress'
             Pick a random next best guess ${v_{rnbg}}^{\alpha}$ among ${\Omega_k}^{\alpha[d_i]}$
             For each non confidential attribute $\alpha'$
               If $\alpha'[d_i]! = \nu$ {
                 Create the maximum impact data value candidate $MIV[\alpha][\alpha']$
                 Set $MIV[\alpha'][\alpha].\alpha_{MI}$ to $\alpha'$
                 Set $MIV[\alpha'][\alpha].impact$ to $\frac{|D[\alpha[d]=\alpha[d_i] \wedge \alpha'[d]=\alpha'[d_i]]-d_i|}{|D[\alpha[d]=v_{rnbg} \wedge \alpha'[d]=\alpha'[d_i]]|}$
               }
           }
           For each non confidential attribute $\alpha'$
             Find average impact $MIV[\alpha'].average\_impact$
           Sort maximum impact attributes by $average\_impact$ in descending order
           For each maximum impact value $miv \in MIV$ {
             Replace the maximum impact data value $miv.\alpha_{MI}[d_i]$ with $\nu$
             For each confidential attribute $\alpha \in S$ {
               Update the probabilities
               If $p(\alpha[d_i]|d_i) \leq p({v_{rnbg}}^{\alpha}|d_i)$ Remove $\alpha$ from $S$
             }
             If $|S| = 0$ break
           }
END
```

Figure 3.8: Pseudocode of e-DROPP Algorithm

aims at decreasing $p(\alpha_j[d_i]|d_i)$ below that of the next best guess v_{nbg} instead of v_{rnbg}.

Definition 4.1. Next Best Guess. *The next best guess, $v_{nbg} \in V_{\alpha_j}$, is a randomly selected value from V_{α_j} satisfying the following conditions.*
i. It is different from $\alpha_j[d_i]$,

$$v_{nbg} \neq \alpha_j[d_i] \tag{3.12}$$

ii. It is among the top-2 probable set,

$$v_{nbg} \in {\Omega_2}^{\alpha_j[d_i]} \tag{3.13}$$

iii. The probability of ${\alpha_j}^{th}$ attribute of d_i being equal to v_{nbg} is smaller than that of confidential data value $\alpha_j[d_i]$ and greater than zero.

$$p(\alpha_j[d_i]|d_i) > p(v_{nbg}|d_i) > 0 \tag{3.14}$$

This leads to a change in the ordering of the top-2 probable set ${\Omega_2}^{\alpha_j[d_i]} = \{\alpha_j[d_i], v_{nbg}\}$. Knowing this fact, an adversary can predict the actual confidential data value to be the one with the second highest probability in ${\Omega_2}^{\alpha_j[d_i]}$ with a confidence equal to the success rate of the algorithm. That is to say, if the success rate of the algorithm is 100%, then the adversary can predict the actual confidential data value with 100% confidence. This problem exists not only in DECP but also in INCP and DROPP algorithms. Therefore, the random next best guess is employed during suppression in order to reduce the confidence of an adversary predicting the actual confidential value as shown below.

$$Confidence = \frac{SuccessRate}{k} \tag{3.15}$$

The second issue, that is inherent in all suppression algorithms, occurs when $|V_{\alpha_j}| = 2$. Let us assume that the decision to suppress the confidential data value $\alpha_j[d_i]$ is not randomized when $|V_{\alpha_j}| = 2$. In this case, the algorithms will try to suppress the confidential data value with the maximum possible success rate. Knowing this fact, an adversary can predict the actual confidential data value to be the one with the second highest probability in V_{α_j} with a confidence equal to the success rate of the algorithm. In order to avoid such attacks, we randomly decide to suppress a confidential data value for microdata sets with $|V_{\alpha_j}| = 2$.

Another issue is the effectiveness of the suppression algorithms against different classification models. Remember that two of the proposed algorithms, the DECP and INCP algorithms, aim at downgrading the classification model by modifying $D - d_i$. In the first method, the probability of resemblance of the tuple containing the confidential data value to other tuples $d \in D$ satisfying $\alpha_j[d] = \alpha_j[d_i]$ is reduced. And, in the latter method the probability of resemblance of the tuple containing the confidential data value to the tuples $d \in D$ satisfying $\alpha_j[d] \neq \alpha_j[d_i]$ is increased. On the other hand, the DROPP and HID3 algorithms aim at downgrading the microdata tuple containing the confidential data value. Both methods find the attributes that enable correct prediction of the actual confidential value and hide them from the tuple containing the confidential data value. As a result, the probability of similarity of the tuple containing the confidential data value to the other tuples $d \in D$ satisfying $\alpha_j[d] = \alpha_j[d_i]$ is reduced. Since all classification methods tend to find the target attribute value of a tuple based on its resemblance to other tuples in the training data set, the proposed suppression algorithms are expected to achieve their goal even when used with other classification methods. In order to verify this, we measured the effectiveness of each algorithm against both Naïve Bayesian, ID3 and SVM classification. The results can be found in Chapter 5.

The final issue that needs to be discussed is the side effects of the proposed algorithms which is related to the number of attribute values hidden excluding the confidential data value. Remember that, for each suppression algorithm we derived an upper bound for the number of attribute values that will be modified in the previous section. According to these derivations we can conclude the following;

i. The upper bound for the number of data values that can be modified by the INCP algorithm depends on m, the number of tuples in D,

ii. The upper bound for the number of data values that can be modified by the DROPP and HID3 algorithms depends on n, the number of attributes in D,

iii. The upper bound for the number of data values that can be modified by the DECP algorithm depends on $n*m$, the number of attributes in D times the number of tuples in D,

Now, let us assume that $m >> n$. In this case, the worst case performance of the DROPP and HID3 algorithms should be much better than the worst case performance of the DECP and INCP algorithms with respect to the side effects. However, for data sets satisfying $n >> m$, e.g. gene expression data, the worst case performance of the INCP algorithm will outperform the DECP, DROPP and HID3 algorithms with respect to the side effects. Note that the DECP algorithm will perform slightly worse than the other algorithms, as in both cases either m or n loses its significance with respect to the other term.

Chapter 4

SUPPRESSING MICRODATA TO PREVENT CLASSIFICATION BASED INFERENCE USING GENERALIZATION

Besides deletion, as presented in Section 3, we have proposed four algorithms to suppress a confidential data value using the generalization modification strategy. The proposed algorithms aim at suppressing only one confidential data value at a time, against two popular classifier types: probabilistic and decision tree classifiers, as shown in Figure 4.1. We select Naïve Bayesian and ID3 as typical representatives of probabilistic and decision tree classifiers respectively, and developed our heuristics accordingly. Moreover, we propose enhancement to two of the proposed algorithms to suppress multiple confidential data values with fewer side effects.

4.1 Suppression Against Probabilistic Classification Models

In the following, we first explain how we calculate the class conditional frequencies in presence of taxonomies, then we present three suppression algorithms preventing probabilistic classification based inference using generalization. The proposed algorithms

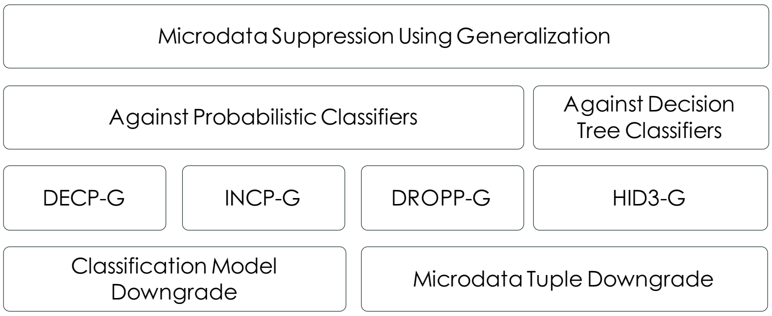

Microdata Suppression Using Generalization
Against Probabilistic Classifiers
Against Decision Tree Classifiers
DECP-G
INCP-G
DROPP-G
HID3-G
Classification Model Downgrade
Microdata Tuple Downgrade

Figure 4.1: Taxonomy of Microdata Suppression Algorithms using Generalization Modification Technique

aim to suppress a confidential data value, such that it is no longer among the Top-1 Probable value set.

Definition 4.1. Top-k Probable. *Let $\alpha_j[d_i]$ be confidential, and thus generalized by λ levels using the taxomomy T_{α_j}. The Naïve Bayesian Classifier built using $D[\alpha_j[d] = Generalize(\alpha_j[d_i], \lambda, T_{\alpha_j}) \vee Descendant(\alpha_j[d], Generalize(\alpha_j[d_i], \lambda, T_{\alpha_j})) - d_i$ as the training data set will predict the Top-k Probable value set for $\alpha_j[d_i]$ as ${\Omega_k}^{\alpha_j[d_i]} \subseteq V_{\alpha_j}$. The Top-k Probable value set satisfies the following constraints.*

i. It's size is equal to k.

$$\left|{\Omega_k}^{\alpha_j[d_i]}\right| = k \tag{4.1}$$

ii. Every $\omega \in {\Omega_k}^{\alpha_j[d_i]}$ should be a descendant of generalized $\alpha_j[d_i]$.

$$Descendant(\omega, Generalize(\alpha_j[d_i], \lambda, T_{\alpha_j})) \tag{4.2}$$

iii. The probability of $\alpha_j[d_i]$ being equal to the least probable value in the Top-k Probable value set is greater than the probability of $\alpha_j[d_i]$ being equal to the most probable value among the remaining attribute values.

$$p(\omega|d_i) > p(v|d_i) \mid \forall v \in V_{\alpha_j} - {\Omega_k}^{\alpha_j[d_i]} \wedge Descendant(v, Generalize(\alpha_j[d_i], \lambda, T_{\alpha_j}))$$
$$\wedge\ \omega \in {\Omega_k}^{\alpha_j[d_i]} \tag{4.3}$$

The proposed suppression algorithms aim at either reducing $p(\alpha_j[d_i]|d_i)$ below that of an attribute selected randomly from the Top-k Probable value set, called the *Random Next Best Guess*, or increasing the probability of a set of attributes selected from the Top-k Probable value set, called the *Next Best Guess Set*, above $p(\alpha_j[d_i]|d_i)$.

Definition 4.2. Random Next Best Guess(RNBG). *The random next best guess, $v_{rnbg} \in V_{\alpha_j}$, is a randomly selected value from V_{α_j} satisfying the following conditions.*
i. It is different from $\alpha_j[d_i]$.

$$v_{rnbg} \neq \alpha_j[d_i] \tag{4.4}$$

ii. It is among the Top-k Probable value set.

$$v_{rnbg} \in {\Omega_k}^{\alpha_j[d_i]} \tag{4.5}$$

iii. The probability of ${\alpha_j}^{th}$ attribute of d_i being equal to v_{rnbg} is smaller than that of confidential data value $\alpha_j[d_i]$ and greater than zero.

$$p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i) > 0 \tag{4.6}$$

Definition 4.3. Next Best Guess (NBG) Set. *The next best guess set, $S_{nbg} \subseteq {\Omega_k}^{\alpha_j[d_i]}$, for microdata tuple d_i is the set of all attribute values $v \in {\Omega_k}^{\alpha_j[d_i]} - \alpha_j[d_i]$ satisfying the following condition.*

$$S_{nbg} = \left\{ v | v \in {\Omega_k}^{\alpha_j[d_i]} - \alpha_j[d_i] \wedge p(v|d_i) \geq p(v_{rnbg}|d_i) \right\} \tag{4.7}$$

4.1.1 Calculation of Class Conditional Frequency Counts

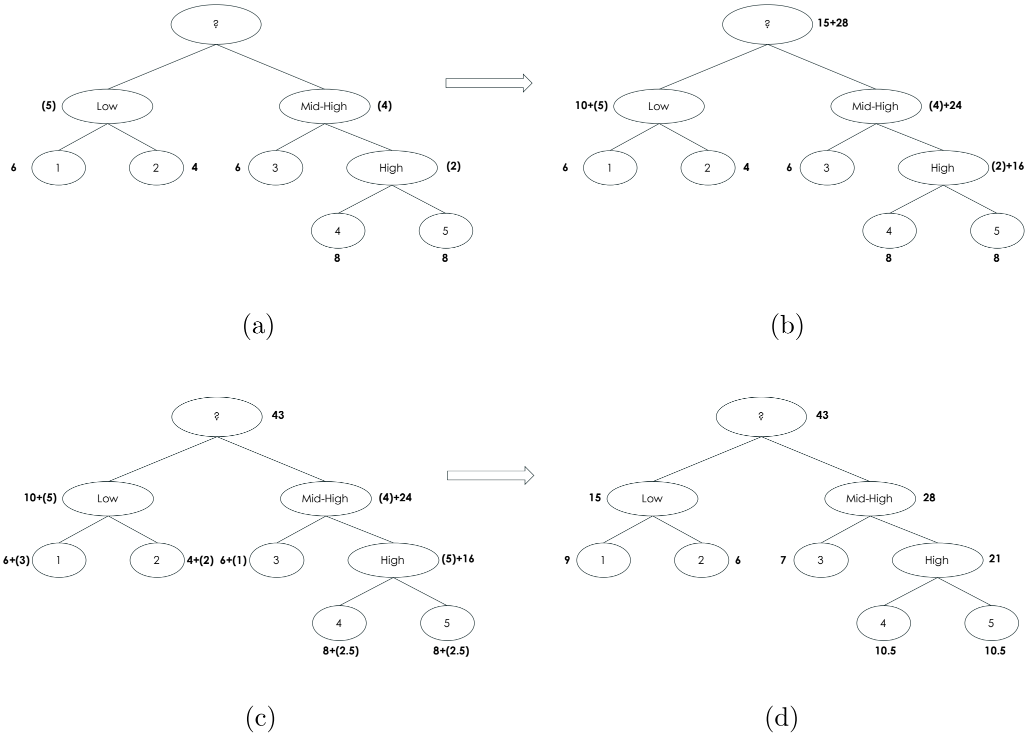
In order to calculate the class conditional frequency counts in presence of taxonomies, we adopt the expectation maximization algorithm proposed by Zhang et al. [62]. Given an attribute value taxonomy T_{α_i} for attribute $\alpha_i \in \Lambda$, the proposed algorithm constructs

```
INPUT:    $D$, the training data set
          $T_{\alpha_i}$, the taxonomy set attribute $\alpha_i \in \Lambda$
OUTPUT:   $CCFC(T_{\alpha_i})$, the class conditional frequency counts for $\alpha_i \in \Lambda$
BEGIN
          For each node $node \in T_{\alpha_i}$ {
            Create the corresponding node in $CCFC(T_{\alpha_i})$
            $CCFC(T_{\alpha_i}).node.count = |D[\alpha_i = node.value]|$
          }
          Starting from the leaves of $CCFC(T_{\alpha_i})$ aggregate $node.count$ upwards
          Starting from the root of $CCFC(T_{\alpha_i})$ propogate $node.count$ downwards according to the
              observed distribution among its children
          Return $CCFC(T_{\alpha_i})$
END
```

Figure 4.2: Calculation of Class Conditional Frequency Counts in Presence of a Taxonomy by Zhang et al.

a tree of class conditional frequency counts $CCFC(T\alpha_i)$ such that (i) there is a one-to-one correspondence between the nodes of the taxonomy T_{α_i} and the nodes of the corresponding $CCFC(T_{\alpha_i})$, (ii) the class conditional frequency count associated with a non leaf node,i.e. non-primitive value, of $CCFC(T_{\alpha_i})$ is equal to the sum of the class conditional frequency counts associated with its direct descendants. When all the tuples in the data set D are fully specified, i.e. all attribute values are primitive, construction of $CCFC(T_{\alpha_i})$ for each attribute $\alpha_i \in \Lambda$ is straightforward. First, the class conditional frequency counts associated with each of the primitive values of α_i is identified from the data set D. Then, identified class conditional frequency counts are used recursively to compute the class conditional frequency counts associated with non-primitive values of α_i. When some of the data are partially specified, i.e. some attribute values are already generalized, and therefore non-primitive, the two step approach illustrated in Figure 4.2 is used to construct $CCFC(T_{\alpha_i})$. First, an upward pass aggregating the class conditional frequency counts based on the specified attribute values in the data set is realized. Then, the counts associated with partially specified attribute values are propagated down through the tree, augmenting the counts at lower levels according to the distribution of values along the branches based on the subset of the data for which the corresponding values are fully specified.

Example 6. Let us illustrate estimation of class conditional frequency counts using a simple example. On the nausea symptom taxonomy shown in Figure 4.3-(a), we first



(a) (b)

(c) (d)

Figure 4.3: Estimation of class conditional frequency counts.(a) Initial counts associated with each attribute value showing the number of positively labeled instances. (b) Aggregation of counts upwards from each node to its ancestors. (c) Distribution of counts of a partially specified attribute value downwards among descendant nodes. (d) Updating the estimated frequency counts for all attribute values.

mark each attribute value with a count showing the total number of instances having that specific value as their nausea symptom. Then, we aggregate the counts upwards from each node to its ancestors. For example, in Figure 4.3-(b), the two counts 6 and 4 on primitive attribute values *1* and *2* add up to 10 as the count for *Low*. As we already have 5 instances which have their nausea symptom specified as *Low*, the two counts (5 and 10) aggregate towards the root. Next, we distribute the counts of a generalized attribute values downwards according to the distributions of values among their descendant nodes. For example, 5, the count of instances which have their nausea symptom specified as *Low*, is propagated down as 3 and 2 to descendant nodes *1* and *2* (See Figure 4.3-(c) for values in parentheses). Finally, we update the estimated frequency counts for all attribute values as shown in Figure 4.3-(d).

```
INPUT:      $D$, the microdata set
            $T$, the taxonomy set for each attribute
            $MaxLevel$, the maximum depth among the taxonomy sets
            $d_i$, the tuple containing the confidential data value
            $\alpha_j$, the attribute containing the confidential data value
            $k$, degree of suppression
OUTPUT:     $D'$, the new data set
BEGIN
            $v_{actual} = \alpha_j[d_i]$
            $\lambda_{\alpha_j} = 1$
            While $\lambda_{\alpha_j} <= FindMaxLevel(T, \alpha_j)$ {
              $\alpha_j[d_i] = Generalize(v_{actual}, \lambda_{\alpha_j}, T_{\alpha_j}, k)$
              If $success = DECP\text{-}G(D, T, v_{actual}, d_i, \alpha_j, \lambda_{\alpha_j}, MaxLevel)$ break
              Increment $\lambda_{\alpha_j}$ by 1
            }
            If not $success$ Delete tuple $d_i$
END
BOOL        $DECP\text{-}G(D,T,v_{actual},d_i,\alpha_j,\lambda_{\alpha_j},MaxLevel,k)$ {
              $V_{candidate}$ = all $v \in V_{\alpha_j}$ satisfying $Generalize(v, \lambda_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$
              Find probabilities $p(v|d_i)$ for all $v \in V_{candidate}$
              If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$ {
                If $|V_{candidate}| = 2$ {
                  Randomly decide whether or not to continue suppression
                  Return $true$
                }
                Select top-k from $V_{candidate}$ to form $\Omega_k{}^{\alpha_j[d_i]}$
                Pick a random next best guess $v_{rnbg}$ among $\Omega_k{}^{\alpha_j[d_i]}$
                $\lambda = 1$
                While $\lambda <= MaxLevel$ {
                  While $p(v_{actual}|d_i) > p(v_{rnbg}|d_i)$ and candidates for maximum impact attribute exist {
                    Find maximum impact attribute $\alpha_{MI}{}^{\alpha_j[d_i]}$ for current level $\lambda$
                    Find maximum impact data values
                    $F_1 = |D[\alpha_j[d] = v_{actual} \wedge \alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d] = Generalize(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T)]|$
                    $F_2 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d],$
                                    $Generalize(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda, T))]|$
                    $F_3 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d],$
                                    $Generalize(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T))]|$
                    $generalized_{MIDV} = Generalize(\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda, T)$
                    While $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$ and $(count_1 > 1$ or $\lambda > 1)$ {
                      Replace the next data value in maximum impact data values with $generalized_{MIDV}$
                      $p(\alpha_j[d_i]|d_i) *= \frac{F_1+F_3-1}{F_1+F_3} \times \frac{F_2}{F_2-1}$
                      Decrement $F_1$ and $F_2$ by 1
                    }
                  }
                  Increment $\lambda$ by 1
                }
              }
              If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$ Revert all changes and return $false$
              Return $true$
            }
ATTRIBUTE   $FindMaximumImpactAttributeForDECP\text{-}G(D,T,v_{actual},d_i,\alpha_j,\lambda)$ {
              $ratio = |D|$
              For each attribute $\alpha \in \Lambda - \alpha_j${
                $F_1 = |D[\alpha_j[d] = v_{actual} \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda - 1, T_\alpha)]|$
                $F_2 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha[d], Generalize(\alpha[d_i], \lambda, T_\alpha))]|$
                $F_3 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha[d], Generalize(\alpha[d_i], \lambda - 1, T_\alpha))]|$
                $ratio' = \frac{F_1+F3-1}{F_1+F_3} \times \frac{F_2}{F_2-1}$
                If $ratio' < ratio$ and $(F_1 > 1$ or $\lambda > 1)$ {
                  $ratio = \frac{F_1+F_3-1}{F_1+F_3} \times \frac{F_2}{F_2-1}$
                  $\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T} = \alpha$
                }
              }
              Return $\alpha_{MI}{}^{\alpha_j[d_i],\lambda,T}$
            }
```

Figure 4.4: Pseudocode of DECP-G Algorithm

4.1.2 DECP-G Algorithm

The DECP-G algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$ via generalization so that it cannot be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D'-d_i,\alpha_j}$. It accomplishes its goal by decreasing the probability $p(\alpha_j[d_i]|d_i)$ below that of the random next best guess v_{rnbg}.

Definition 4.4. Maximum Impact Attribute for DECP-G. *Let* F_1,F_2 *and* F_3 *denote the original frequency count of tuples satisfying the following constraints respectively* $\alpha[d] = Generalize(\alpha[d_i], \lambda - 1, T)$, $Descendant(\alpha[d], Generalize(\alpha[d_i], \lambda, T)) = True$, *and* $Descendant(\alpha[d], Generalize(\alpha[d_i], \lambda - 1, T)) = True$ *among the dataset* $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$. *The attribute with maximum impact on* $p(\alpha_j[d_i]|d_i)$, *when generalized by* λ *levels, denoted by* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$, *is the one that satisfies the following condition.*

$$
{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T} = \arg\min_{\alpha\in A}\left(\frac{F_1 + F_3 - 1}{F_1 + F_3} \times \frac{F_2}{F_2 - 1}\right)
$$

$$
(|D[\alpha_j[d] = \alpha_j[d_i] \wedge \alpha[d] = Generalize\,(\alpha[d_i], \lambda - 1, T_\alpha)] - d_i| > 1 \vee \lambda > 1) \tag{4.8}
$$

Definition 4.5. Maximum Impact Data Values. *The maximum impact data values are the instances of* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ *in tuples* $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T_\alpha]$ *excluding* d_i.

In each iteration, the DECP-G algorithm (i) identifies the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ for the current generalization level λ, and (ii) modifies the tuples $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T_\alpha)] - d_i$, by generalizing ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d]$ λ levels until the goal is achieved, that is until $p(\alpha_j[d_i]|d_i)$ becomes less than $p(v_{rnbg}|d_i)$. Each such generalization results in the maximum possible reduction in $p(\alpha_j[d_i]|d_i)$, thus requiring less number of generalizations.

Theorem 4.1. *Let* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ *be the maximum impact attribute for level* λ *satisfying the equation* (4.8). *Then, every generalization of a maximum impact data value from* *level $\lambda - 1$ to λ causes the maximum decrease in $p(\alpha_j[d_i]|d_i)$, thus resulting in fewer data values to be modified.*

Proof: Let us first find the effect of generalizing a maximum impact data value on $p(\alpha_j[d_i])\ p(d_i|\alpha_j[d_i])$. Remember that, since $p(d_i)$ is same for all $v \in V_{\alpha_j}$, it can be ignored when calculating $p(\alpha_j[d_i]|d_i)$.

$$\begin{aligned} p(\alpha_j[d_i]|d_i) &= \frac{p(\alpha_j[d_i])p(d_i|\alpha_j[d_i])}{p(d_i)} \\ &\cong p(\alpha_j[d_i])p(d_i|\alpha_j[d_i]) \\ &\cong p(\alpha_j[d_i])p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i]) \\ &\quad \times \prod\nolimits_{\alpha \in \mathit{\Lambda} - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i]}\right\}} p(\alpha[d_i]|\alpha_j[d_i]) \end{aligned}$$

Let us assume that;

- $\bar{F}_0$ be the updated frequency count of tuples satisfying $\alpha_j[d] = \alpha_j[d_i]$ among D excluding d_i,
- $\bar{F}_1$ be the updated frequency count of tuples satisfying the following constraint;

  $${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = \mathit{Generalize}({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda, T)$$

  among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i,
- F_1 be the original frequency count of tuples satisfying the following constraint;

  $${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = \mathit{Generalize}({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T)$$

  among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i,
- F_2 be the original frequency count of tuples satisfying the following constraint;

  $$\mathit{Descendant}({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d], \mathit{Generalize}({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T))$$

  among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i,

- F_3 be the original frequency count of tuples satisfying the following constraint;

$$Descendant({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d], Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda-1, T))$$

  among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i, and

- c be the difference between the updated frequency count $\bar{F}_1$ and $F_1 + F_3$.

Single generalization of a maximum impact data value from level $\lambda - 1$ to λ causes $p({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i]|\alpha_j[d_i])$ to decrease from $\frac{1+\frac{F_1}{F_3}+c\frac{F_1+F_3}{F_2F_3}}{\bar{F}_0}$ to $\frac{1+\frac{F_1-1}{F_3}+(c+1)\frac{F_1+F_3-1}{(F_2-1)F_3}}{\bar{F}_0}$. This, in turn decreases $p(d_i|\alpha_j[d_i])$ by $\frac{F_1+F_3-1}{F_1+_{F_3}} \times \frac{F_2}{F_2-1}$ as shown below.

$$\begin{aligned}
p'(d_i|\alpha_j[d_i]) &= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p'(\alpha[d_i]|\alpha_j[d_i])p'({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i]|\alpha_j[d_i]) \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|\alpha_j[d_i])\frac{1+\frac{F_1-1}{F_3}+(c+1)\frac{F_1+F_3-1}{(F_2-1)F_3}}{\bar{F}_0} \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|\alpha_j[d_i])\frac{F_1+F_3-1}{F_2-1}\frac{F_2+c}{F_3\bar{F}_0} \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|\alpha_j[d_i])\frac{F_1+F_3-1}{F_2-1}\frac{F_2+c}{F_3\bar{F}_0} \\
&\qquad \times\frac{1+\frac{F_1}{F_3}+c\frac{F_1+F_3}{F_2F_3}}{1+\frac{F_1}{F_3}+c\frac{F_1+F_3}{F_2F_3}} \\
&= \prod\nolimits_{\alpha\in\Lambda-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|\alpha_j[d_i])\frac{F_1+F_3-1}{F_2-1}\frac{F_2+c}{F_3} \\
&\qquad \times\frac{p({\alpha_{MI}}^{\alpha_j[d_i]}[d_i]|\alpha_j[d_i])}{1+\frac{F_1}{F_3}+c\frac{F_1+F_3}{F_2F_3}} \\
&= \frac{F_1+F_3-1}{F_2-1}\frac{F_2+c}{F_3}\frac{F_2}{F_1+F_3}\frac{F_3}{F_2+c}p(d_i|\alpha_j[d_i]) \\
&= \frac{F_1+F_3-1}{F_1+F_3}\frac{F_2}{F_2-1}p(d_i|\alpha_j[d_i])
\end{aligned}$$

Let us assume that there is another attribute α_k which decreases $p(\alpha_j[d_i]|d_i)$ more than that of ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$. This implies the following;

$$\frac{F_{1,\alpha_k}+F_{3,\alpha_k}-1}{F_{1,\alpha_k+F_{3,\alpha_k}}}\frac{F_{2,\alpha_k}}{F_{2,\alpha_k}-1} < \frac{F_{1,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}+F_{3,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}-1}{F_{1,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}+F_{3,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}}\frac{F_{2,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}}{F_{2,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}}-1}$$

However, this contradicts the definition of *Maximum Impact Attribute*. Therefore, we can conclude that every generalization of a maximum impact data value causes the highest decrease in $p(\alpha_j[d_i]|d_i)$ which in turn implies that the number of data values that should be modified is minimal□

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm backs up the confidential data value $\alpha_j[d_i]$ as v_{actual}, and generalizes it to the next level. Then, it verifies the need for suppression. It first finds the candidate set for the confidential data value, $V_{candidate}$, which includes all $v \in V_{\alpha_j}$ satisfying the constraint $Generalize(v, level_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$. Then, it finds $p(v|d_i)$ for all $v \in V_{candidate}$ and checks the truth value of the following assertion:

$$p(v_{actual}|d_i) > p(v|d_i)|\forall v \in V_{candidate} \tag{4.9}$$

If Assertion (4.11) is true, it picks a random next best guess v_{rnbg} from $V_{candidate}$ and sets the level of generalization to 1. Next, in each iteration it finds the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ for the current level λ and generalizes the maximum impact data values by λ levels as long as $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$. After processing all maximum impact attributes, it re-checks the truth value of Assertion (4.9). If Assertion (4.9) is still true, then it increments the current level of generalization by 1 and repeats the generalization process until there are no levels left for generalization. After processing all levels and maximum impact attributes, it re-checks the truth value of Assertion (4.9). If Assertion (4.9) is still true, then it reverts all changes, generalizes the confidential value to the next level, and repeats the above steps until either the confidential data value is successfully suppressed or there are no levels left to generalize the confidential data value. Finally, if the algorithm is not successful, i.e. the confidential data value cannot be suppressed, then tuple d_i is deleted from the microdata set. An overview of the algorithm is provided in Figure 4.4.

If $|V_{candidate}| = 2$ is true, then suppressing the confidential data value might result in an adversary guessing it correctly with 100% confidence. Therefore, the decision to suppress a confidential data value is randomized for the case where $|V_{candidate}| = 2$. This results in an adversary guessing the actual confidential data value with 50% confidence which is the maximum uncertainty that can be achieved under such circumstances.

Table 4.1: Academic Health Medical Records to be Shared with Academic Research Institute

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	5	2	Dyspepsia
4	5	4	Mid-High	4	Dyspepsia
5	2	3	3	5	Dyspepsia
3	3	3	4	2	Dyspepsia
2	2	5	3	3	Dyspepsia
4	4	3	2	4	Dyspepsia
3	4	3	3	3	Gastritis
5	2	4	4	2	Gastritis
3	3	4	4	3	Gastritis
3	2	4	5	5	Gastritis
4	5	2	3	5	Gastritis
4	3	3	2	4	Gastritis
3	2	3	2	3	S.I. Ulcer
5	2	2	3	4	S.I. Ulcer
4	3	2	2	2	S.I. Ulcer
3	4	5	3	4	S.I. Ulcer
4	3	3	4	5	S.I. Ulcer
4	4	3	3	4	**Dyspepsia**

Lemma 4.1. *Let $\alpha_j[d_i]$ be the confidential data value, n be the number of attributes and N be the number of tuples in $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$. Then, the upper bound for the number of data values that can be modified by the DECP-G algorithm is equal to $(n-1)(N-1)$.*

Proof: The DECP-G algorithm modifies the maximum impact data values from the tuples $d \in D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] - d_i$. As $D[\alpha_j[d] = \alpha_j[d_i] \wedge {\alpha_{MI}}^{\alpha_j[d_i]}[d] = {\alpha_{MI}}^{\alpha_j[d_i]}[d_i]] \subseteq D[\alpha_j[d] = \alpha_j[d_i]]$, the number of tuples that can be modified for each maximum impact attribute is bounded by $N-1$. At each iteration, the DECP-G algorithm picks a different maximum impact attribute and generalizes the instances of this attribute. Since, there are $n-1$ different alternatives for a maximum impact attribute, we can conclude that the DECP-G algorithm can change at most $(n-1)(N-1)$ data values for suppressing a confidential data value $\square$

Example 7. Table 4.1 shows a set of patient records to be disclosed to third parties for

Table 4.2: Academic Health Medical Records to be Shared with Academic Research Institute

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	5	2	Dyspepsia
4	5	4	Mid-High	4	Dyspepsia
5	2	3	3	5	Dyspepsia
3	3	3	4	2	Dyspepsia
2	2	5	3	3	Dyspepsia
4	4	3	2	4	Dyspepsia
3	4	3	3	3	Gastritis
5	2	4	4	2	Gastritis
3	3	4	4	3	Gastritis
3	2	4	5	5	Gastritis
4	5	2	3	5	Gastritis
4	3	3	2	4	Gastritis
3	2	3	2	3	S.I. Ulcer
5	2	2	3	4	S.I. Ulcer
4	3	2	2	2	S.I. Ulcer
3	4	5	3	4	S.I. Ulcer
4	3	3	4	5	S.I. Ulcer
4	4	3	3	4	**Gastric Disease**

research purposes. The diagnosis of the last patient, i.e. d_{18}, is confidential. Therefore, it is generalized one level up as shown in Table 4.2. Now, let us illustrate how the DECP-G algorithm suppresses the confidential diagnosis.

Step 1. Initially, the confidential diagnosis is generalized one level up as shown in Table 4.2.

Step 2. Next, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{candidate} = \{dyspepsia, gastritis\}$. The Naïve Bayesian classification model constructed using the medical records of Table 4.2 is shown in Table 4.3. According to the model the probabilities are $p(dyspepsia|d_{18}) = 0,0058$, and $p(gastritis|d_{18}) = 0,0005$.

Step 3. The probability $p(dyspepsia|d_{18})$ is greater than $p(gastritis|d_{18})$. As the confidential diagnosis can be correctly predicted, the suppression process starts.

Step 4. Since the size of $V_{candidate}$ is equal to 2, it is randomly decided whether to

Table 4.3: Naïve Bayesian Classification Model Constructed During the Run of DECP-G Algorithm

			p(**Symptom**|**Diagnosis**)					
Step	**Diagnosis**	p(**Diagnosis**)	**I**	**CP**	**P**	**N**	**B**	p(**Diagnosis**|d_{18})
2	Dyspepsia	6/12	3/6	2/6	3/6	2,5/6	2/6	0,00579
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
8	Dyspepsia	6/12	3/6	2/6	3/6	5/18	2/6	0,00386
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
10	Dyspepsia	6/12	3/6	1,5/6	3/6	5/18	2/6	0,00289
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
12	Dyspepsia	6/12	3/6	1,5/6	3/6	5/18	1,5/6	0,00217
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
14	Dyspepsia	6/12	3/6	1,5/6	2,5/6	5/18	1,5/6	0,00181
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
15	Dyspepsia	6/12	3/6	1,5/6	5/18	5/18	1,5/6	0,00121
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
17	Dyspepsia	6/12	4/9	1,5/6	5/18	5/18	1,5/6	0,00107
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
18	Dyspepsia	6/12	2/6	1,5/6	5/18	5/18	1,5/6	0,00080
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
21	Dyspepsia	6/12	2/6	2/9	5/18	5/18	1,5/6	0,00071
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
23	Dyspepsia	6/12	2/6	2/9	5/18	5/18	2/9	0,00064
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
25	Dyspepsia	6/12	7,5/24	2/9	5/18	5/18	2/9	0,00060
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
26	Dyspepsia	6/12	5/18	2/9	5/18	5/18	2/9	0,00053
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
28	Dyspepsia	6/12	5/18	2/9	1,6/6	5/18	2/9	0,00051
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051

continue with the suppression or not. Let us assume that it has been decided to continue with the suppression process.

Step 5. Let us assume that gastritis is selected as the random next best guess. From this point on the DECP-G algorithm will decrease $p(dyspepsia|d_{18})$ below $p(gastritis|d_{18})$.

Step 6. To select the maximum impact attribute, the ratios for each symptom attribute is found as shown in Table 4.4. The nausea symptom has the minimum ratio. Therefore, it is selected as the maximum impact attribute.

Step 7. All tuples d satisfying the constraint $nausea\ [d] = 3 \wedge diagnosis[d] = dyspepsia$

Table 4.4: Ratios Calculated to Determine the Maximum Impact Attribute for DECP-G

Step	Result	Indigestion	Chest Pain	Palpitation	Nausea	Burning
6	F_1	3	2	3	2	2
	F_2	4	3	5	4	3
	F_3	0	0	0	0	0
6	*Ratio*	8/9	3/4	10/12	4/6	3/4
19	F_1	2	1	2	2	1
	F_2	5	4	6	6	4
	F_3	2	2	3	3	2
19	*Ratio*	15/16	8/9	24/25	24/25	8/9

is found. Tuples 3 and 5 satisfy the mentioned constraint.

Step 8. The nause attribute of tuple 3 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{4}{6}$ to $0,00386$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is chest pain.

Step 9. All tuples d satisfying the constraint $chestpain[d] = 4 \wedge diagnosis[d] = dyspepsia$ is found. Tuples 2 and 6 satisfy the mentioned constraint.

Step 10. The chest pain attribute of tuple 2 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{3}{4}$ to $0,00289$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is burning.

Step 11. All tuples d satisfying the constraint $burning[d] = 4 \wedge diagnosis[d] = dyspepsia$ is found. Tuples 2 and 6 satisfy the mentioned constraint.

Step 12. The burning attribute of tuple 2 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{3}{4}$ to $0,00217$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is palpitation.

Step 13. All tuples d satisfying the constraint $palpitation[d] = 3 \wedge diagnosis[d] = dyspepsia$ is found. Tuples 3,4 and 6 satisfy the mentioned constraint.

Step 14. The palpitation attribute of tuple 3 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{10}{12}$ to $0,00181$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact value.

Step 15. The palpitation attribute of tuple 4 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{4}{6}$ to $0,00121$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is indigestion.

Step 16. All tuples d satisfying the constraint $indigestion[d] = 4 \wedge diagnosis[d] = dyspepsia$ is found. Tuples 1,2 and 6 satisfy the mentioned constraint.

Step 17. The indigestion attribute of tuple 1 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{8}{9}$ to $0,00107$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact value.

Step 18. The indigestion attribute of tuple 2 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{3}{4}$ to $0,00080$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next level $\lambda = 2$.

Step 19. To select the maximum impact attribute, the ratios for each symptom attribute is found as shown in Table 4.4. The chest pain symptom has the minimum ratio. Therefore, it is selected as the maximum impact attribute.

Step 20. All tuples d satisfying the constraint $chestpain[d] = H \wedge diagnosis[d] = dyspepsia$ is found. Tuple 1 satisfies the mentioned constraint.

Step 21. The chest pain attribute of tuple 1 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{8}{9}$ to $0,00071$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is burning.

Table 4.5: Academic Health Medical Records Shared with Academic Research Institute after Execution of DECP-G

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
Mid-High	**Mid-High**	2	5	2	Dyspepsia
Mid-High	5	4	Mid-High	**Mid-High**	Dyspepsia
5	2	?	**Mid-High**	5	Dyspepsia
3	3	**Mid-High**	4	2	Dyspepsia
2	2	5	3	3	Dyspepsia
4	4	3	2	4	Dyspepsia
3	4	3	3	3	Gastritis
5	2	4	4	2	Gastritis
3	3	4	4	3	Gastritis
3	2	4	5	5	Gastritis
4	5	2	3	5	Gastritis
4	3	3	2	4	Gastritis
3	2	3	2	3	S.I. Ulcer
5	2	2	3	4	S.I. Ulcer
4	3	2	2	2	S.I. Ulcer
3	4	5	3	4	S.I. Ulcer
4	3	3	4	5	S.I. Ulcer
4	4	3	3	4	**Gastric Disease**

Step 22. All tuples d satisfying the constraint $burning[d] = H \wedge diagnosis[d] = dyspepsia$ is found. Tuple 2 satisfies the mentioned constraint.

Step 23. The burning attribute of tuple 2 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{8}{9}$ to $0,00064$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is indigestion.

Step 24. All tuples d satisfying the constraint $indigestion[d] = H \wedge diagnosis[d] = dyspepsia$ is found. Tuples 1 and 2 satisfy the mentioned constraint.

Step 25. The indigestion attribute of tuple 1 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{15}{16}$ to $0,00060$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact value.

Step 26. The indigestion attribute of tuple 2 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{8}{9}$ to $0,00053$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is palpitation.

Step 27. All tuples d satisfying the constraint $palpitation[d] = MH \wedge diagnosis[d] = dyspepsia$ is found. Tuples 3 and 4 satisfy the mentioned constraint.

Step 28. The palpitation attribute of tuple 3 is generalized one level up. With this replacement $p(dyspepsia|d_{18})$ decreases by $\frac{24}{25}$ to $0,00051$. As $p(dyspepsia|d_{18})$ is equal to $p(gastritis|d_{18})$, the suppression process stops.

The resulting microdata set can be seen in Table 4.5.

4.1.3 INCP-G Algorithm

The INCP-G algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that it cannot be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D'-d_i,\alpha_j}$. It accomplishes its goal, as its name implies, by increasing the probabilities $p(v|d_i)$ for all v in the next best guess set,S_{nbg}, above $p(\alpha_j[d_i] \mid d_i)$.

For each $v \in S_{nbg}$, the INCP-G algorithm identifies the tuples $d \in D[\alpha_j[d] = v]$ having no common attribute value with d_i and modifies them by generalizing $\alpha_j[d]$ in order to increase $p(v|d_i)$.

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm verifies the need for suppression. It finds $p(v|d_i)$ for all $v \in V_{candidate}$ and checks the truth value of Assertion (4.9). If Assertion (4.9) is true, it picks a random next best guess v_{rnbg} from $V_{candidate}$, forms S_{nbg} by finding the attribute values $v \in V_{candidate}$ satisfying $p(v|d_i) \geq p(v_{rnbg}|d_i)$, and sets the level of generalization to 1. Next, for each $v \in S_{nbg}$, the algorithm finds the tuples $d \in D[\neg\ \alpha_1[d] = \alpha_1[d_i] \wedge \ldots \wedge \neg\ \alpha_{j-1}[d] = \alpha_{j-1}[d_i] \wedge \alpha_j[d] = v \wedge \neg\ \alpha_{j+1}[d] = \alpha_{j+1}[d_i] \wedge \ldots \wedge \neg\ \alpha_n[d] = \alpha_n[d_i]]$ and modifies them by generalizing $\alpha_j[d]$ by λ levels until the goal is achieved, that is until $p(v|d_i)$ becomes less than or equal to $p(\alpha_j[d_i]|d_i)$. After processing all levels, it re-

```
INPUT:    $D$, the microdata set
          $T$, the taxonomy set for each attribute
          $MaxLevel$, the maximum depth among the taxonomy sets
          $d_i$, the tuple containing the confidential data value
          $\alpha_j$, the attribute containing the confidential data value
          $k$, degree of suppression
OUTPUT:   $D'$, the new data set
BEGIN
          $v_{actual} = \alpha_j[d_i]$
          $\lambda_{\alpha_j} = 1$
          While $\lambda_{\alpha_j} <= FindMaxLevel(T, \alpha_j)$
          {
            $\alpha_j[d_i] = Generalize(v_{actual}, \lambda_{\alpha_j}, T_{\alpha_j})$
            If $success$ =$INCP\text{-}G(D, T, v_{actual}, d_i, \alpha_j, \lambda_{\alpha_j}, MaxLevel, k)$
              break
            Increment $\lambda_{\alpha_j}$ by 1
          }
          If not $success$
            Run algorithm DECP-G
END
BOOL      $INCP\text{-}G(D,T,v_{actual},d_i,\alpha_j,\lambda_{\alpha_j},MaxLevel,k)$
          {
            $V_{candidate}$ = all $v \in V_{\alpha_j}$ satisfying $Generalize(v, \lambda_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$
            If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$
            {
              If $|V_{candidate}| = 2$
              {
                Randomly decide whether or not to continue suppression
                Return $true$
              }
              Select top-k from $V_{candidate}$ to form ${\Omega_k}^{\alpha_j[d_i]}$
              Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d_i]}$
              $S_{nbg}$ = All attribute values $v \in {\Omega_k}^{\alpha_j[d_i]}$ satisfying $p(v|d_i) \geq p(v_{rnbg}|d_i)$
              $\lambda = 1$
              While $\lambda <= MaxLevel$
              {
                For each $v \in S_{nbg}$
                {
                  While $p(\alpha_j[d_i]|d_i) > p(v|d_i)$ and $D[\alpha_j[d] = Generalize(v, \lambda - 1, T_{\alpha_j})] \neq empty$
                  {
                    $t$ = next tuple in $D[\alpha_j[d] = Generalize(v, \lambda - 1, T_{\alpha_j})]$
                    If $t \cap d_i = empty$
                    {
                      $\alpha_j[t] = Generalize(v, \lambda, T_{\alpha_j})$
                      Recalculate probabilities $p(v|d_i)$ for all $v \in V_{candidate}$
                    }
                  }
                }
                Increment $\lambda$ by 1
              }
            }
            If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$
            {
              Revert all changes
              Return $false$
            }
            Return $true$
          }
```

Figure 4.5: Pseudocode of INCP-G Algorithm

checks the truth value of Assertion (4.9). If Assertion (4.9) is still true, then it reverts all changes, generalizes the confidential data value to the next level, and repeats the above steps until either the confidential data value is successfully suppressed or there are no levels left to generalize the confidential data value. Finally, if the algorithm is not successful, i.e. the confidential data value cannot be suppressed, then DECP-G algorithm is executed to complete the suppression process. An overview of the algorithm is provided in Figure 4.1.3.

Lemma 4.2. *Let $\alpha_j[d_i]$ be the confidential data value, m be the number of tuples in D and N be the number of tuples in $D[\alpha_j[d] = \alpha_j[d_i]] - d_i$. Assuming that there are enough number of tuples that can be used for the suppression process (i.e. no need for executing DECP), the upper bound for the number of data values that can be modified by the INCP-G algorithm is equal to $m - N - 1 - |S_{nbg}|$.*

Proof: The INCP-G algorithm modifies the tuples $d \in D[\neg\ \alpha_1[d] = \alpha_1[d_i] \wedge \ldots \wedge \neg\ \alpha_{j-1}[d] = \alpha_{j-1}[d_i] \wedge \alpha_j[d] = v \wedge \neg\ \alpha_{j+1}[d] = \alpha_{j+1}[d_i] \wedge \ldots \wedge \neg\ \alpha_n[d] = \alpha_n[d_i]]$ for each $v \in S_{nbg}$. In the worst case, S_{nbg} contains all possible values of attribute α_j except $\alpha_j[d_i]$. This implies $\sum_{v \in S_{nbg}} |D[\alpha_j[d] = v]| = m - N - 1$. Moreover, due to the definition of next best guess set and random next best guess the probability $p(v|d_i)$ for each $v \in S_{nbg}$ must be greater than zero. This implies that, in the worst case there exists at least one tuple which has the same data values with d_i (except α_j) for each $v \in S_{nbg}$. So, we can conclude that the INCP algorithm can generalize at most $m - N - 1 - |S_{nbg}|$ data values for suppressing a confidential data value □

Example 8. Now, let us illustrate how the INCP-G algorithm suppresses the confidential diagnosis.

Step 1. Initially, the confidential diagnosis is generalized one level up as shown in Table 4.2.

Step 2. Next, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{candidate} = \{dyspepsia, gastritis\}$. The Naïve Bayesian classification model constructed using the medical records of Table 4.2 is shown in Ta-

Table 4.6: Naïve Bayesian Classification Model Constructed During the Run of INCP-G Algorithm

Step	**Diagnosis**	p(**Diagnosis**)	p(**Symptom**|**Diagnosis**)					p(**Diagnosis**|d_{18})
			I	**CP**	**P**	**N**	**B**	
2	Dyspepsia	$6/12$	$3/6$	$2/6$	$3/6$	$2,5/6$	$2/6$	$0,00579$
	Gastritis	$6/12$	$2/6$	$1/6$	$2/6$	$2/6$	$1/6$	$0,00051$
7	Dyspepsia	$6,\tilde{54}/12$	$3/6,\tilde{54}$	$2/6,\tilde{54}$	$3/6,\tilde{54}$	$2,5/6,\tilde{54}$	$2/6,\tilde{54}$	$0,00409$
	Gastritis	$5,\tilde{45}/12$	$2/5,\tilde{45}$	$1/5,\tilde{45}$	$2/5,\tilde{45}$	$2/5,\tilde{45}$	$1/5,\tilde{45}$	$0,00075$
8	Dyspepsia	$7,2/12$	$3/7,2$	$2/7,2$	$3/7,2$	$2,5/7,2$	$2/7,2$	$0,00279$
	Gastritis	$4,8/12$	$2/4,8$	$1/4,8$	$2/4,8$	$2/4,8$	$1/4,8$	$0,00126$
9	Dyspepsia	$8/12$	$3/8$	$2/8$	$3/8$	$2,5/8$	$2/8$	$0,00183$
	Gastritis	$4/12$	$2/4$	$1/4$	$2/4$	$2/4$	$1/4$	$0,00260$

ble 4.6. According to the model the probabilities are $p(dyspepsia|d_{18}) = 0,0058$, and $p(gastritis|d_{18}) = 0,0005$.

Step 3. The probability $p(dyspepsia|d_{18})$ is greater than $p(gastritis|d_{18})$. As the confidential diagnosis can be correctly predicted, the suppression process starts.

Step 4. Since the size of $V_{candidate}$ is equal to 2, it is randomly decided whether to continue with the suppression or not. Let us assume that it has been decided to continue with the suppression process.

Step 5. Let us assume that gastritis is selected as the random next best guess. From this point on the INCP-G algorithm will increase $p(gastritis|d_{18})$ above $p(dyspepsia|d_{18})$.

Step 6. All tuples which has no common symptoms with d_{18} among $D[diagnosis = gastritis]$ is found. Tuples 8,9 and 10 satisfy the mentioned constraint.

Step 7. The diagnosis attribute of tuple 8 is generalized to *Gastric Disease.* After this generalization, $p(gastritis|d_{18})$ increases to $0,00075$, and $p(dyspepsia|d_{18})$ decreases to $0,00409$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues.

Step 8. The diagnosis attribute of tuple 9 is generalized to *Gastric Disease.* After this generalization, $p(gastritis|d_{18})$ increases to $0,00126$, and $p(dyspepsia|d_{18})$ increases to $0,00279$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression

Table 4.7: Academic Health Medical Records Shared with Academic Research Institute after Execution of INCP-G

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	5	2	Dyspepsia
4	5	4	Mid-High	4	Dyspepsia
5	2	3	3	5	Dyspepsia
3	3	3	4	2	Dyspepsia
2	2	5	3	3	Dyspepsia
4	4	3	2	4	Dyspepsia
3	4	3	3	3	Gastritis
5	2	4	4	2	**Gastric Disease**
3	3	4	4	3	**Gastric Disease**
3	2	4	5	5	**Gastric Disease**
4	5	2	3	5	Gastritis
4	3	3	2	4	Gastritis
3	2	3	2	3	S.I. Ulcer
5	2	2	3	4	S.I. Ulcer
4	3	2	2	2	S.I. Ulcer
3	4	5	3	4	S.I. Ulcer
4	3	3	4	5	S.I. Ulcer
4	4	3	3	4	**Gastric Disease**

process continues.

Step 9. The diagnosis attribute of tuple 10 is generalized to *Gastric Disease*. After this generalization, $p(gastritis|d_{18})$ increases to $0,00260$, and $p(dyspepsia|d_{18})$ increases to $0,00183$. As $p(dyspepsia|d_{18})$ is smaller than $p(gastritis|d_{18})$, the suppression process stops.

The resulting microdata set can be seen in Table 4.7.

4.1.4 DROPP-G Algorithm

The DROPP-G algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that it cannot be correctly predicted by the classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$. It aims at dropping the probability $p(\alpha_j[d_i]|d_i)$ below that of the random next best guess v_{rnbg}, so that it cannot be correctly predicted by the classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$. Unlike

```
INPUT:      $D$, the microdata set
            $T$, the taxonomy set for each attribute
            $MaxLevel$, the maximum depth among the taxonomy sets
            $d_i$, the tuple containing the confidential data value
            $\alpha_j$, the attribute containing the confidential data value
            $k$, degree of suppression
OUTPUT:     $D'$, the new data set
BEGIN
            $v_{actual} = \alpha_j[d_i]$
            $\lambda_{\alpha_j} = 1$
            While $\lambda_{\alpha_j} <= FindMaxLevel(T, \alpha_j)$ {
              $\alpha_j[d_i] = Generalize(v_{actual}, \lambda_{\alpha_j}, T_{\alpha_j})$
              If $success = DROPP\text{-}G(D, T, v_{actual}, d_i, \alpha_j, \lambda_{\alpha_j}, MaxLevel, k)$
                break
              Increment $\lambda_{\alpha_j}$ by 1
            }
            If not $success$
              Delete tuple $d_i$
END
BOOL        $DROPP\text{-}G(D,T,v_{actual},d_i,\alpha_j,\lambda_{\alpha_j},MaxLevel,k)$ {
              $V_{candidate}$ = all $v \in V_{\alpha_j}$ satisfying $Generalize(v, \lambda_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$
              Find probabilities $p(v|d_i)$ for all $v \in V_{candidate}$
              If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$ {
                If $|V_{candidate}| = 2$ {
                  Randomly decide whether or not to continue suppression
                  Return $true$
                }
                Select top-k from $V_{candidate}$ to form ${\Omega_k}^{\alpha_j[d_i]}$
                Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d_i]}$
                $\lambda = 1$
                While $\lambda <= MaxLevel$ {
                  While $p(v_{actual}|d_i) > p(v_{rnbg}|d_i)$ and candidates for maximum impact attribute exist {
                    Find maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ for current level
                    ${\alpha_{MI}}^{\alpha_j[d_i]}[d_i] = Generalize({\alpha_{MI}}^{\alpha_j[d_i]}[d_i], \lambda, T_{\alpha_{MI}})$
                    Recalculate probabilities $p(v|d_i)$ for all $v \in V_{candidate}$
                  }
                  Increment $\lambda$ by 1
                }
              }
              If $p(v_{actual}|d_i) > p(v|d_i) \mid \forall v \in V_{candidate}$ Revert all changes and return $false$
              Return $true$
            }
ATTRIBUTE   $FindMaximumImpactAttributeForDROPP\text{-}G(D,T,v_{actual},d_i,\alpha_j,v_{rnbg},\lambda)$ {
              $ratio = 0$
              For each attribute $\alpha \in \Lambda - \alpha_j$ {
                $\bar{F}_{\alpha,v_{rnbg},\lambda}$ = updated frequency count of tuples satisfying
                                 $D[\alpha_j[d] = v_{rnbg} \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda, T_\alpha)]$
                $\bar{F}_{\alpha,v_{rnbg},\lambda-1}$ = updated frequency count of tuples satisfying
                                 $D[\alpha_j[d] = v_{rnbg} \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda - 1, T_\alpha)]$
                $\bar{F}_{\alpha,v_{actual},\lambda}$ = updated frequency count of tuples satisfying
                                 $D[\alpha_j[d] = v_{actual} \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda, T_\alpha)]$
                $\bar{F}_{\alpha,v_{actual},\lambda-1}$ = updated frequency count of tuples satisfying
                                 $D[\alpha_j[d] = v_{actual} \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda - 1, T_\alpha)]$
                $ratio' = \frac{\bar{F}_{\alpha,v_{rnbg},\lambda}}{\bar{F}_{\alpha,v_{rnbg},\lambda-1}} \times \frac{\bar{F}_{\alpha,v_{actual},\lambda-1}}{\bar{F}_{\alpha,v_{actual},\lambda}}$
                If $ratio' > ratio$ {
                  $ratio = ratio'$
                  ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T} = \alpha$
                }
              }
              Return ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$
            }
```

Figure 4.6: Pseudocode of DROPP-G Algorithm

DECP-G and INCP-G algorithms, it achieves its goal by downgrading the tuple d_i, instead of downgrading classification model ${\varsigma_{nb}}^{D-d_i,\alpha_j}$.

The algorithm employs the following modified definition of *Maximum Impact Attribute.*

Definition 4.6. Maximum Impact Attribute for DROPP-G. *Let us assume that* $\bar{F}_{\alpha,v,\lambda}$ *denote the updated frequency count of tuples satisfying the following constraint* $\alpha_j[d] = v \wedge \alpha[d] = Generalize(\alpha[d_i], \lambda, T)$*. The attribute with maximum impact on* $p(\alpha_j[d_i]|d_i)$*, denoted by* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$*, is the one that satisfies the following condition.*

$$ {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T} = \arg\max{}_{\alpha\in\mathit{\Lambda}} \left(\frac{\bar{F}_{\alpha,v_{rnbg},\lambda}}{\bar{F}_{\alpha,v_{rnbg},\lambda-\mathit{1}}} \times \frac{\bar{F}_{\alpha,\alpha_j[d_i],\lambda-\mathit{1}}}{\bar{F}_{\alpha,\alpha_j[d_i],\lambda}} \right) \qquad (4.10) $$

Definition 4.7. Maximum Impact Data Value. *The maximum impact data value is the instance of maximum impact data attribute* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ *in tuple* d_i*.*

In each iteration, the DROPP-G algorithm identifies ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ and modifies the tuple d_i by generalizing ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i]$ until the goal is achieved, that is until $p(\alpha_j[d_i]|d_i)$ becomes less than $p(v_{rnbg}|d_i)$. Each such replacement results in the maximum possible reduction in $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$, thus requiring less number of modifications.

Theorem 4.2. *Let* ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$ *be the maximum impact attribute satisfying Equation* (4.10)*. Then, generalization of the maximum impact data value causes the maximum decrease in* $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$*, thus resulting in fewer data values to be modified.*

Proof: Let us first find the effect of generalizing a maximum impact data value on $p(\alpha_j[d_i]|d_i)$ and $p(v_{rnbg}|d_i)$. Remember that, since $p(d_i)$ is same for all $v \in V_{\alpha_j}$, it can be ignored when calculating $p(\alpha_j[d_i]|d_i)$.

$$
\begin{aligned}
p(\alpha_j[d_i]|d_i) &= \frac{p(\alpha_j[d_i])p(d_i|\alpha_j[d_i])}{p(d_i)} \\
&\cong p(\alpha_j[d_i])p(d_i|\alpha_j[d_i]) \\
&\cong p(\alpha_j[d_i])p({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i]|\alpha_j[d_i]) \\
&\quad \times \prod{}_{\alpha\in\mathit{\Lambda}-\left\{\alpha_j,{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|\alpha_j[d_i])
\end{aligned}
$$

Similarly,

$$\begin{aligned} p(v_{rnbg}|d_i) &= \frac{p(v_{rnbg})p(d_i|v_{rnbg})}{p(d_i)} \\ &\cong p(v_{rnbg})p(d_i|v_{rnbg}) \\ &\cong p(v_{rnbg})p({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i]|v_{rnbg}) \\ &\quad \times \prod\nolimits_{\alpha \in \mathit{\Lambda} - \left\{\alpha_j, {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}\right\}} p(\alpha[d_i]|v_{rnbg}) \end{aligned}$$

Let us assume that;

- $\bar{F}_{v_{rnbg}}$ be the updated frequency count of tuples satisfying the following constraint $D[\alpha_j[d] = v_{rnbg}]$ excluding d_i,

- $\bar{F}_{\alpha_j[d_i]}$ be the updated frequency count of tuples satisfying the following constraint $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i,

- $\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}, v_{rnbg}, \lambda}$ be the updated frequency count of tuples satisfying the following constraint;

  $$ {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda, T) $$

  among $D[\alpha_j[d] = v_{rnbg}]$ excluding d_i,

- $\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}, v_{rnbg}, \lambda-1}$ be the updated frequency count of tuples satisfying the following constraint;

  $$ {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda-1, T) $$

  among $D[\alpha_j[d] = v_{rnbg}]$ excluding d_i,

- $\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}, \alpha_j[d_i], \lambda}$ be the updated frequency count of tuples satisfying the following constraint;

  $$ {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = Generalize({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda, T) $$

  among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i, and

- $\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}$ be the updated frequency count of tuples satisfying the following constraint;

$$ {\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d] = \mathit{Generalize}({\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}[d_i], \lambda - 1, T) $$

among $D[\alpha_j[d] = \alpha_j[d_i]]$ excluding d_i.

Replacement of the maximum impact data value causes $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ to change by $\frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}} \times \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}}$ as shown below.

$$
\begin{aligned}
p'(\alpha_j[d_i]|d_i) &\cong p'(\alpha_j[d_i])p'(d_i|\alpha_j[d_i]) \\
&\cong p(\alpha_j[d_i]) \prod\nolimits_{\alpha\in\Lambda-\{\alpha_j,{\alpha_{MI'}}^{\alpha_j[d_i]}\}} p(\alpha[d_i]|\alpha_j[d_i]) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}{\bar{F}_{\alpha_j[d_i]}} \\
&\cong p(\alpha_j[d_i]) \prod\nolimits_{\alpha\in\Lambda-\{\alpha_j,{\alpha_{MI'}}^{\alpha_j[d_i]}\}} p(\alpha[d_i]|\alpha_j[d_i]) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}{\bar{F}_{\alpha_j[d_i]}} \\
&\quad \times \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}} \\
&\cong p(\alpha_j[d_i]|d_i) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}} \\
p'(v_{rnbg}|d_i) &\cong p'(v_{rnbg})p'(d_i|v_{rnbg}) \\
&\cong p(v_{rnbg}) \prod\nolimits_{\alpha\in\Lambda-\{\alpha_j,{\alpha_{MI'}}^{\alpha_j[d_i]}\}} p(\alpha[d_i]|v_{rnbg}) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}}{\bar{F}_{v_{rnbg}}} \\
&\cong p(v_{rnbg}) \prod\nolimits_{\alpha\in\Lambda-\{\alpha_j,{\alpha_{MI'}}^{\alpha_j[d_i]}\}} p(\alpha[d_i]|v_{rnbg}) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}}{\bar{F}_{v_{rnbg}}} \\
&\quad \times \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}} \\
&\cong p(v_{rnbg}|d_i) \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}}
\end{aligned}
$$

Now let us assume that there is another attribute α_k which decreases $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ more than that of ${\alpha_{MI}}^{\alpha_j[d_i],\lambda,T}$. This implies the following:

$$ \frac{\bar{F}_{\alpha_k,\alpha_j[d_i],\lambda}}{\bar{F}_{\alpha_k,\alpha_j[d_i],\lambda-1}} \frac{\bar{F}_{\alpha_k,v_{rnbg},\lambda-1}}{\bar{F}_{\alpha_k,v_{rnbg},\lambda}} < \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}} \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}} $$

$$\frac{\bar{F}_{\alpha_k,\alpha_j[d_i],\lambda-1}}{\bar{F}_{\alpha_k,\alpha_j[d_i],\lambda}}\frac{\bar{F}_{\alpha_k,v_{rnbg},\lambda}}{\bar{F}_{\alpha_k,v_{rnbg},\lambda-1}} > \frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda-1}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},\alpha_j[d_i],\lambda}}\frac{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda}}{\bar{F}_{{\alpha_{MI}}^{\alpha_j[d_i],\lambda,T},v_{rnbg},\lambda-1}}$$

However, this contradicts the definition of *Maximum Impact Attribute*. So, we can conclude that every generalization of the maximum impact data value causes the highest decrease in $\frac{p(\alpha_j[d_i]|d_i)}{p(v_{rnbg}|d_i)}$ which in turn implies that the number of data values that should be modified is minimal□

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm backs up the confidential data value $\alpha_j[d_i]$ as v_{actual}, and generalizes it to the next level. Then, it verifies the need for suppression. It first finds the candidate set for the confidential data value, $V_{candidate}$, which includes all $v \in V_{\alpha_j}$ satisfying the constraint $Generalize(v, level_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$. Then, it finds $p(v|d_i)$ for all $v \in V_{candidate}$ and checks the truth value of the following assertion:

$$p(v_{actual}|d_i) > p(v|d_i)|\forall v \in V_{candidate} \tag{4.11}$$

If Assertion (4.11) is true, it picks a random next best guess v_{rnbg} from $V_{candidate}$ and sets the level of generalization to 1. Next, in each iteration it finds the maximum impact attribute ${\alpha_{MI}}^{\alpha_j[d_i]}$ for the current level λ and generalizes the maximum impact data value by λ levels as long as $p(\alpha_j[d_i]|d_i) > p(v_{rnbg}|d_i)$. After each iteration, it re-checks the truth value of Assertion (4.11) to decide whether to continue execution or not. If Assertion (4.11) is still true after all possible maximum impact attributes are processed, it increments the current level of generalization by 1 and repeats the generalization process until there are no levels left for generalization. After processing all levels and maximum impact attributes, it re-checks the truth value of Assertion (4.11). If Assertion (4.11) is still true, then it reverts all changes, generalizes the confidential value to the next level, and repeats the above steps until either the confidential data value is successfully suppressed or there are no levels left to generalize the confidential data value. Finally, if the algorithm is not successful, i.e. the confidential data value cannot be suppressed, then tuple d_i is deleted from the microdata set. An overview of the algorithm is provided in Figure 4.6.

If $|V_{candidate}| = 2$ is true, then suppressing the confidential data value might result in an adversary guessing it correctly with 100% confidence. Therefore, the decision to suppress a confidential data value is randomized for the case where $|V_{candidate}| = 2$. This results in an adversary guessing the actual confidential data value with 50% confidence which is the maximum uncertainty that can be achieved under such circumstances.

Lemma 4.3. *Let $\alpha_j[d_i]$ be the confidential data value and n be the number of attributes. Then, the upper bound for the number of data values that can be modified by the DROPP-G algorithm is equal to $n - 1$.*

Proof: The DROPP-G algorithm modifies just the tuple d_i which has $n - 1$ data values excluding the confidential data value. So, we can conclude that the DROPP-G algorithm can generalize at most $n - 1$ data values for suppressing a confidential data value □

Example 9. Now, let us illustrate how the DROPP-G algorithm suppresses the confidential diagnosis.

Step 1. Initially, the confidential diagnosis is generalized one level up as shown in Table 4.2.

Step 2. Next, the Naïve Bayesian classification model is constructed to find the probabilities $p(v|d_i)$ for all $v \in V_{candidate} = \{dyspepsia, gastritis\}$. The Naïve Bayesian classification model constructed using the medical records of Table 4.2 is shown in Table 4.8. According to the model the probabilities are $p(dyspepsia|d_{18}) = 0,0058$, and $p(gastritis|d_{18}) = 0,0005$.

Step 3. The probability $p(dyspepsia|d_{18})$ is greater than $p(gastritis|d_{18})$. As the confidential diagnosis can be correctly predicted, the suppression process starts.

Step 4. Since the size of $V_{candidate}$ is equal to 2, it is randomly decided whether to

Table 4.8: Naïve Bayesian Classification Model Constructed During the Run of DROPP-G Algorithm

			p(**Symptom**|**Diagnosis**)					
Step	**Diagnosis**	p(**Diagnosis**)	**I**	**CP**	**P**	**N**	**B**	p(**Diagnosis**|d_{18})
2	Dyspepsia	6/12	3/6	2/6	3/6	2,5/6	2/6	0,00579
	Gastritis	6/12	2/6	1/6	2/6	2/6	1/6	0,00051
7	Dyspepsia	6/12	3/6	2/6	3/6	2,5/6	3/6	0,00868
	Gastritis	6/12	2/6	1/6	2/6	2/6	3/6	0,00154
8	Dyspepsia	6/12	3/6	2/6	5/6	2,5/6	3/6	0,01447
	Gastritis	6/12	2/6	1/6	5/6	2/6	3/6	0,00386
9	Dyspepsia	6/12	3/6	3/6	5/6	2,5/6	3/6	0,02170
	Gastritis	6/12	2/6	2/6	5/6	2/6	3/6	0,00772
10	Dyspepsia	6/12	3/6	3/6	5/6	5/6	3/6	0,04340
	Gastritis	6/12	2/6	2/6	5/6	5/6	3/6	0,01929
11	Dyspepsia	6/12	3/6	4/6	5/6	5/6	3/6	0,05787
	Gastritis	6/12	2/6	3/6	5/6	5/6	3/6	0,02894
13	Dyspepsia	6/12	5/6	4/6	5/6	5/6	3/6	0,07234
	Gastritis	6/12	6/6	3/6	5/6	5/6	3/6	0,05787
14	Dyspepsia	6/12	5/6	5/6	5/6	5/6	3/6	0,09645
	Gastritis	6/12	6/6	6/6	5/6	5/6	3/6	0,11574

continue with the suppression or not. Let us assume that it has been decided to continue with the suppression process.

Step 5. Let us assume that gastritis is selected as the random next best guess. From this point on the DROPP-G algorithm will drop $p(dyspepsia|d_{18})$ below $p(gastritis|d_{18})$.

Step 6. To select the maximum impact attribute, the ratios for each symptom attribute is found for $\lambda = 1$ as shown in Table 4.9. The burning symptom has the maximum ratio. Therefore, it is selected as the maximum impact attribute.

Step 7. The burning attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to 0,00868, and $p(gastritis|d_{18})$ increases to 0,00154. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is palpitation.

Step 8. The palpitation attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to 0,01447, and $p(gastritis|d_{18})$ increases to 0,00386. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression

Table 4.9: Ratios Calculated to Determine the Maximum Impact Attribute for DROPP-G

Step	**Result**	**Indigestion**	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**
6	$\bar{F}_{\alpha,Gastritis,1}$	3	2	5	5	3
	$\bar{F}_{\alpha,Gastritis,0}$	2	1	2	2	1
	$\bar{F}_{\alpha,Dyspepsia,1}$	4	3	5	5	3
	$\bar{F}_{\alpha,Dyspepsia,0}$	3	2	3	$2,5$	2
6	*Ratio*	9/8	4/3	15/10	25/20	6/3
12	$\bar{F}_{\alpha,Gastritis,2}$	6	4	6	6	5
	$\bar{F}_{\alpha,Gastritis,1}$	3	2	5	5	3
	$\bar{F}_{\alpha,Dyspepsia,2}$	5	4	6	6	4
	$\bar{F}_{\alpha,Dyspepsia,1}$	4	3	5	5	3
12	*Ratio*	24/15	12/8	1	1	15/12

process continues with the next maximum impact attribute which is chest pain.

Step 9. The chest pain attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to $0,02170$, and $p(gastritis|d_{18})$ increases to $0,00772$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is nausea.

Step 10. The nausea attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to $0,04340$, and $p(gastritis|d_{18})$ increases to $0,01929$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is indigestion.

Step 11. The indigestion attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to $0,05787$, and $p(gastritis|d_{18})$ increases to $0,02894$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next level $\lambda = 2$.

Step 12. To select the maximum impact attribute, the ratios for each symptom attribute is found for $\lambda = 2$ as shown in Table 4.9. The indigestion symptom has the maximum ratio. Therefore, it is selected as the maximum impact attribute.

Step 13. The indigestion attribute of tuple 18 is generalized one level up. With this

Table 4.10: Academic Health Medical Records Shared with Academic Research Institute after Execution of DROPP-G

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	5	2	Dyspepsia
4	5	4	Mid-High	4	Dyspepsia
5	2	3	3	5	Dyspepsia
3	3	3	4	2	Dyspepsia
2	2	5	3	3	Dyspepsia
4	4	3	2	4	Dyspepsia
3	4	3	3	3	Gastritis
5	2	4	4	2	Gastritis
3	3	4	4	3	Gastritis
3	2	4	5	5	Gastritis
4	5	2	3	5	Gastritis
4	3	3	2	4	Gastritis
3	2	3	2	3	S.I. Ulcer
5	2	2	3	4	S.I. Ulcer
4	3	2	2	2	S.I. Ulcer
3	4	5	3	4	S.I. Ulcer
4	3	3	4	5	S.I. Ulcer
Mid-High	**Mid-High**	**Mid-High**	**Mid-High**	**High**	**Gastric Disease**

generalization $p(dyspepsia|d_{18})$ increases to $0,07234$, and $p(gastritis|d_{18})$ increases to $0,05787$. As $p(dyspepsia|d_{18})$ is still greater than $p(gastritis|d_{18})$, the suppression process continues with the next maximum impact attribute which is chest pain.

Step 14. The chest pain attribute of tuple 18 is generalized one level up. With this generalization $p(dyspepsia|d_{18})$ increases to $0,09645$, and $p(gastritis|d_{18})$ increases to $0,11574$. As $p(dyspepsia|d_{18})$ is smaller than $p(gastritis|d_{18})$, the suppression process stops.

The resulting microdata set can be seen in Table 4.10.

4.2 Suppression Against Decision Tree Classification Models

In the following, we present the HID3-G algorithm for preventing decision tree classification based inference using generalization. Although we have used ID3 in our experiments, the proposed algorithm can be used to suppress a confidential data value from any decision tree algorithm.

4.2.1 HID3-G Algorithm

The HID3-G algorithm aims at suppressing the confidential data value $\alpha_j[d_i]$, so that the ID3 classifier ${\varsigma_{id3}}^{D-d_i,\alpha_j}$ cannot correctly predict its actual value. Similar to the DROPP-G algorithm, it achieves its goal by downgrading the microdata tuple d_i containing the confidential data value.

The algorithm works as follows: Let $\alpha_j[d_i]$ be confidential. As the first step, the algorithm backs up the confidential data value $\alpha_j[d_i]$ as v_{actual}, and generalizes it to the next level. Then, it verifies the need for suppression. It first finds the candidate set for the confidential data value, $V_{candidate}$, which includes all $v \in V_{\alpha_j}$ satisfying the constraint $Generalize(v, level_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$. Then, it builds the decision tree using the training data set $D[\alpha_j = v] - d_i$ where $v \in V_{candidate}$ and verifies the need for suppression. If ${\varsigma_{id3}}^{D[\alpha_j=v]-d_i|v\in V_{candidate},\alpha_j}$ can correctly predict the confidential data value it calls the recursive *ID3Hide* function. The *ID3Hide* function first checks whether the root node is a leaf or not. If it is a leaf, and its value is different from the actual confidential data value v_{actual} it returns *true*, which in turn terminates the recursive function successfully. Or else, it returns *false*. If the root node is not a leaf, then it finds the most probable value $v_\pi \in V_{candidate}$ for $\alpha_j[d_i]$, and checks whether v_π is equal to $v_a ctual$ or not. If the most probable value v_π is not equal to the actual confidential data value $\alpha_j[d_i]$, it returns *true*. Otherwise, it further explores the child nodes of the root in order to suppress $\alpha_j[d_i]$. Let the decision attribute of the root node be α_{root}, the most common child of the root (i.e. the child with highest training population) be $child_{MC}$ and the children matching $\alpha_{root}[d_i]$ ranked with respect to their matching training population (in descending order) be $children_{Match}$. If $\alpha_{root}[d_i] = \nu$ then it tries to suppress the confidential data value using the most probable child node $children_{Match}[0]$ which is actualy equal to $child_{MC}$. Or else, it identifies the current generalization level of $\alpha_{root}[d_i]$ and the maximum level of generalization for α_{root}. Until the current generalization level is greater than or equal to the maximum allowed level it repeats the following; It first checks whether $children_{Match}[0]$ is equal to $child_{MC}$ or not.If it is then it generalizes $\alpha_{root}[d_i]$ one level up, reidentifies the $children_{Match}$, and recursively tries to suppress using the child node $children_{Match}[0]$ which is equal to $child_{MC}$. Otherwise, it recursively tries to suppress using the child node $children_{Match}[0]$. If the supression does not succeed, then it generalizes $\alpha_{root}[d_i]$ one level up and reidentifies the $children_{Match}$. After exploring all possible sub-branches, if the algorithm fails to suppress the confidential data value, it reverts all changes and deletes the tuple d_i from the microdata set. An overview of the algorithm is provided in Figure 4.7.

If $|V_{\alpha_j}| = 2$ is true, then suppressing the confidential data value might result in an adversary guessing it correctly with 100% confidence. Therefore, the decision to suppress a confidential data value is randomized for the case where $|V_{\alpha_j}| = 2$. This results in an adversary guessing the actual confidential data value with 50% confidence which is the maximum uncertainty that can be achieved under such circumstances.

Lemma 4.4. *Let $\alpha_j[d_i]$ be the confidential data value and n be the number of attributes. Then, the upper bound for the number of data values that can be modified by the HID3-G algorithm is equal to $n - 1$.*

Proof: The HID3-G algorithm modifies just the tuple d_i which has $n - 1$ data values excluding the confidential data value. So, we can conclude that the HID3-G algorithm can generalize at most $n - 1$ data values for suppressing a confidential data value □

Example 10. For this specific example, let us assume that the decision tree built using

```
INPUT:    $D$, the microdata set
          $T$, the taxonomy set for each attribute
          $d_i$, the tuple containing the confidential data value
          $\alpha_j$, the attribute containing the confidential data value
OUTPUT:   $D'$, the new data set
BEGIN
          $v_{actual} = \alpha_j[d_i]$
          $\lambda_{\alpha_j} = 1$
          While $\lambda_{\alpha_j} <= FindMaxLevel(T, \alpha_j)$ {
            $\alpha_j[d_i] = Generalize(v_{actual}, \lambda_{\alpha_j}, T_{\alpha_j})$
            $V_{candidate}$ = all $v \in V_{\alpha_j}$ satisfying $Generalize(v, \lambda_{\alpha_j}, T_{\alpha_j}) = \alpha_j[d_i]$
            $root$=Build the decision tree using ID3($D[\alpha_j[d] = v] - d_i | v \in V_{candidate}$)
            $v_\pi$ =Classify($root, d_i$)
            If $v_\pi = v_{actual}$ {
              If $|V_{candidate}| = 2$ {
                Randomly decide whether or not to continue suppression
                Return $true$
              }
              If $success$ =$ID3Hide(root, T, v_{actual}, d_i)$
                break
            }
            Increment $\lambda_{\alpha_j}$ by 1
          }
          If not $success$
            Delete tuple $d_i$
END
BOOL      $ID3Hide(root, T, v_{actual}, d_i)$
{
          If $root$ is a leaf return $root.value \neq v_{actual}$
          $v_\pi$ =Classify($root, d_i$)
          If $v_\pi \neq v_{actual}$ return $true$
          $\alpha_{root}$ = decision attribute of the $root$
          $child_{MC}$ = most common child of the $root$
          $children_{Match}$ = children matching $\alpha_{root}[d_i]$ ranked wrt. training set size
          If $\alpha_{root}[d_i] = \nu$ return $ID3Hide(children_{Match}[0])$
          Else {
            $\lambda$ =Current generalization level of $\alpha_{root}$
            $MaxLevel$ =FindMaxLevel($T, \alpha_{root}$)
            Do {
              If $child_{MC}$ <> $children_{Match}[0]$
                If $ID3Hide(children_{Match}[0])$ return $true$
                Else {
                  Increment $\lambda$ by 1
                  Generalize($\alpha_{root}[d_i], \lambda, T_{\alpha_{root}}$)
                  $children_{Match}$ = children containing all possible values of $\alpha_{root}[d_i]$ ranked wrt. their training set size
                }
              Else {
                Increment $\lambda$ by 1
                Generalize($\alpha_{root}[d_i], \lambda, T_{\alpha_{root}}$)
                If $ID3Hide(children_{Match}[0])$ return $true$
              }
            }
            While $\lambda < MaxLevel$
            Revert changes to $\alpha_{root}[d_i]$
            return $false$
          }
```

Figure 4.7: Pseudocode of HID3-G Algorithm

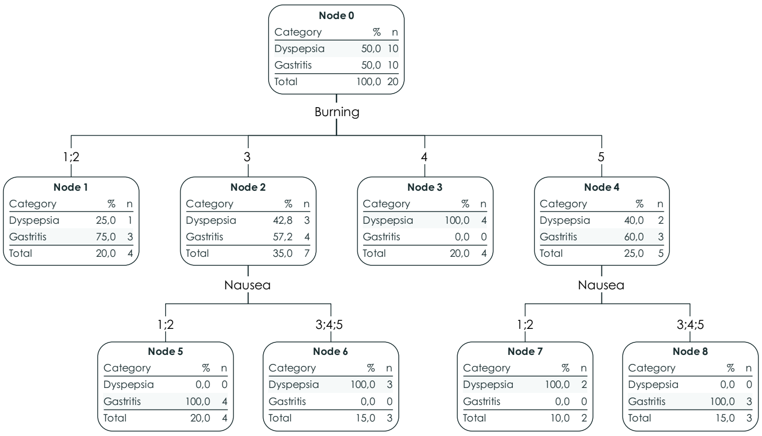


Figure 4.8: An Example Decision Tree

the training data set is as shown in 4.8, and the tuple with confidential diagnosis is as shown Table 4.11 Now, let us illustrate how the HID3-G algorithm suppresses the confidential diagnosis.

Step 1. Initially, the confidential diagnosis is generalized one level up as **Gastric Disease**.

Step 2. Next, the ID3 classification model is constructed using the tuples whose $diagnosis \in \{dyspepsia, gastritis\}$. Let us assume that the decision tree built using the training data set is as shown in 4.8. According to the model $v_{\pi} = dyspepsia$.

Step 3. As the confidential data value can be correctly predicted,the suppression process starts.

Step 4. Since the size of $V_{candidate} = \{dyspepsia, gastritis\}$ is equal to 2, it is randomly decided whether to continue with the suppression or not. Let us assume that it has been decided to continue with the suppression process.

Table 4.11: Tuple Whose Confidential Diagnosis To Be Suppressed By HID3-G

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	3	3	Dyspepsia

Step 5. Starting from the *root=node 0*, the ID3Hide function checks whether it is possible to correctly predict the confidential diagnosis or not. Since it can be correctly predicted using the path $burning = 3 \wedge nausea = 3$, the suppression process continues.

Step 6. The most common child node of the root and the matching child node are both found to be *node 2*.

Step 7. Since the matching child is equal to the most common child node of the root, the burning attribute is generalized one level up, and ID3Hide function is recursively called with *root=node 2*.

Step 8. Starting from the subtree *root=node 2*, the ID3Hide function checks whether it is possible to correctly predict the confidential diagnosis or not. Since the diagnosis can be correctly predicted using the path $nausea = 3$, the suppression process continues.

Step 9. The most common child node of the root and the matching child node are found to be *node 5* and *node 6* respectively.

Step 10. Since the matching child is not equal to the most common child node of the root, the ID3Hide function is recursively called with the matching child *root=node 6.*

Step 11. Since the *root=node 6* is a leaf node, and its value is equal to the actual confidential data value, *False* is returned from the recursive ID3Hide call.

Step 12. Since the recursive call to ID3Hide is not successful, nausea attribute is generalized one level up to be *Mid-High.*

Step 13. The matching child node is found to be *node 6.*

Step 14. Since the matching child is not equal to the most common child node of the root, the ID3Hide function is recursively called with the matching child *root=node 6.*

Step 15. As *root=node 6* is a leaf, the ID3Hide function checks whether the most probable value, i.e. dyspepsia, and the confidential diagnosis are equal or not. As they are equal, the function returns from the recursive call signaling an unsuccessful run.

Step 16. Since the recursive call to ID3Hide is not successful, the nausea attribute is generalized one level up to be *?*.

Step 17. The matching children nodes are found to be *node 5* and *node 6*.

Step 18. As, the first matching child is equal to the most common child node of the root, the ID3Hide function is recursively called with the most common child *root=node 5*.

Step 19. As *root=node 5* is a leaf, the ID3Hide function checks whether the most probable value, i.e. gastritis, and the confidential diagnosis are equal or not. Since they are not equal, the function returns from the recursive call signaling a successful run.

The resulting microdata tuple can be seen in Table 4.12.

Table 4.12: Tuple Whose Confidential Diagnosis Suppressed By HID3-G

Indigestion	**Chest Pain**	**Palpitation**	**Nausea**	**Burning**	**Diagnosis**
4	4	2	?	**Mid-High**	**Gastric Disease**

4.3 Suppression of Multiple Confidential Data Values

In the following, we present the enhanced versions of DECP-G and DROPP-G algorithms for preventing probabilistic classification based inference. The proposed algorithms aim to reduce to side-effects while suppressing multiple confidential data values.

4.3.1 e-DECP-G Algorithm

The enhanced DECP-G algorithm aims at suppressing multiple confidential data values, so that none of them can be correctly predicted by the downgraded classification model ${\varsigma_{nb}}^{D',\alpha_j}$. The proposed algorithm reduces the side-effects of the original DECP-G algorithm when (1) *all confidential data values belong to a single attribute*, and (2) all confidential data values have the same value.

The algorithm works as follows: Let α_j be the confidential attribute, $S \subset D$ be the set of tuples for which α_j, satisfying the constraint $\alpha_j[d] = conf\_value$ for all $d \in S$,is confidential. As the first step, the algorithm generalizes all confidential data values one level up. Then, it identifies the candidate maximum impact data values, and initializes their primary and secondary impacts along with counts. The primary impact is the number of tuples which will be affected (i.e. the probabilities will be affected) if an instance of the maximum impact attribute is generalized. The secondary impact, on the other hand is the ratio that is used to determine the maximum impact attribute. Next, for each tuple $d \in S$, the need for suppression is verified by finding $p(v|d)$ for all $v \in V_{candidate}$ and checking the truth value of the following assertion:

$$p(\alpha_j[d]|d) > p(v|d) | \forall v \in V_{candidate} - \alpha_j[d] \qquad (4.12)$$

If Assertion (4.12) is true for a tuple $d \in S$, it picks a random next best guess ${v_{rnbg}}^{d}$, from $V_{candidate}$. Next, the candidate maximum impact data values are sorted. Different from the DECP-G, which uses only the secondary impact to determine which maximum impact data value to use, e-DECP-G also uses the primary impact in order to guarantee suppression of maximum number of confidential data values with a single iteration. With maximum impact values sorted, the rest of the execution is quite similar to the original DECP-G which involves replacement of maximum impact data value instances, re-calculation of probabilities and re-checking of Assertion (4.12). An overview of the algorithm is provided in Figure 4.9.

4.3.2 e-DROPP-G Algorithm

The enhanced DROPP-G algorithm aims at suppressing multiple confidential data values at a time so that none of them can be correctly predicted by the corresponding

```
INPUT:    $D$, the microdata set
          $T$, the taxonomy set for each attribute
          $MaxLevel$, the maximum depth among the taxonomy sets
          $S$, the set of tuples for which $\alpha_j$ is confidential
          $\alpha_j$, the attribute containing the confidential data values
          $conf\_value$, the value of confidential attribute $\alpha_j \in S$
          $k$, degree of suppression
OUTPUT:   $D'$, the new data set
BEGIN
          $v_{actual} = conf\_value$
          $\lambda_{\alpha_j} = 1$
          While $\lambda_{\alpha_j} <= FindMaxLevel(T, \alpha_j)$ {
            For each tuple $d \in S$
              $\alpha_j[d] = Generalize(v_{actual}, \lambda_{\alpha_j}, T_{\alpha_j}, k)$
            $e\text{-}DECP\text{-}G(D, T, v_{actual}, S, \alpha_j, \lambda_{\alpha_j}, MaxLevel)$ break
            Increment $\lambda_{\alpha_j}$ by 1
          }
          For each tuple $d \in S$
            Delete tuple $d$
END
BOOL      $e\text{-}DECP\text{-}G(D,T,v_{actual},d_i,\alpha_j,\lambda_{\alpha_j},MaxLevel,k)$
          {
            $\lambda = 1$
            While $\lambda <= MaxLevel$ {
              For each attribute $\alpha \in \Lambda - \alpha_j$ {
                For each possible value of $v_\alpha \in V_\alpha$ {
                  Create the maximum impact data value candidate $MIV[\alpha][v_\alpha]$
                  Set $MIV[\alpha][v_\alpha].primary\_impact$ to 0
                  $F_1 = |D[\alpha_j[d] = v_{actual} \wedge \alpha[d] = Generalize(v_\alpha, \lambda - 1, T_\alpha)]|$
                  $F_2 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha[d], Generalize(v_\alpha, \lambda, T_\alpha))]|$
                  $F_3 = |D[\alpha_j[d] = v_{actual} \wedge Descendant(\alpha[d], Generalize(v_\alpha, \lambda - 1, T_\alpha)]|$
                  Set $MIV[\alpha][v_\alpha].secondary\_impact = \frac{F_1+F3-1}{F_1+F_3} \times \frac{F_2}{F_2-1}$
                  Set $MIV[\alpha][v_\alpha].count = F_1$
              }
              $V_{candidate}$ = all $v \in V_{\alpha_j}$ satisfying $Generalize(v, \lambda_{\alpha_j}, T_{\alpha_j}) = \alpha_j[S[0]]$
              For each tuple $d \in S$ {
                Find probabilities $p(v|d)$ for all $v \in V_{candidate}$
                If not $p(v_{actual}|d) > p(v|d) \mid \forall v \in V_{candidate}$ Remove $d$ from $S$
                Else If $|V_{candidate}| = 2$
                  Randomly decide whether to suppress the confidential data value and remove $d$ from $S$
                    if decision = ‘not suppress’
                If $d \in S$ {
                  Select top-k from $V_{candidate}$ to form ${\Omega_k}^{\alpha_j[d]}$
                  Pick a random next best guess $v_{rnbg}$ among ${\Omega_k}^{\alpha_j[d]}$
                  For each attribute $\alpha \in \Lambda - \alpha_j$
                    Increment $MIV[\alpha][\alpha[d]].primary\_impact$ by 1
                }
              }
              Sort $MIV$ first by $primary\_impact$ in descending order, then by $secondary\_impact$ in ascending order
              For each maximum impact value $miv \in MIV$ {
                While $|S| > 0$ and $(miv.count > 1$ or $\lambda > 1)$ {
                  Generalize the next instance of $miv$
                  $miv.count - -$
                  For each tuple $d \in S$ {
                    Update $p(\alpha_j[d]|d)$
                    If $p(\alpha_j[d]|d) \leq p({v_{rnbg}}^d|d)$ Remove $d$ from $S$
                  }
                }
                If $|S| = 0$ break
              }
            }
          }
```

Figure 4.9: Pseudocode of e-DECP-G Algorithm

classification models ${\varsigma_{nb}}^{D,\alpha}$. The proposed algorithm reduces the side-effects of the original DROPP-G algorithm when *all confidential data values belong to a single tuple*.

The algorithm works as follows: Let $[d_i]$ be the tuple containing multiple confidential data values, and S be the set of attributes containing a confidential data value in d_i. As the first step, the algorithm verifies the need for suppression for each confidential data value. More specifically, for each $\alpha \in S$, it finds $p(v|d_i)$ where $v \in V_{candidate}$ and checks the truth value of the following assertion:

$$p(\alpha[d_i]|d_i) > p(v|d_i)|\forall v \in V_{candidate} - \alpha[d_i] \tag{4.13}$$

If Assertion (4.13) is true, it picks a random next best guess ${v_{rnbg}}^{\alpha}$ from $V_{candidate}$. Next, it identifies the candidate maximum impact data values, and initializes their impacts on each confidential value. To identify the maximum impact data value in each iteration, the impacts of candidates are averaged and sorted. With maximum impact values sorted, the rest of the execution is quite similar to the original DROPP-G which involves generalization of maximum impact data value instances of d_i, re-calculation of probabilities and re-checking of Assertion (4.13). An overview of the algorithm is provided in Figure 4.10.

```
INPUT:     $D$, the microdata set
           $T$, the taxonomy set for each attribute
           $MaxLevel$, the maximum depth among the taxonomy sets
           $d_i$, the tuple containing the confidential data value
           $S$, the set of attributes containing a confidential data value in $d_i$
           $k$, degree of suppression
OUTPUT:    $D'$, the new data set
BEGIN
           For each attribute $\alpha \in S$
             ${v_{actual}}^{\alpha} = \alpha[d_i]$
           $\lambda_S = 1$
           While $\lambda_S <= MaxLevel$ {
             $\lambda = 1$
             While $\lambda <= MaxLevel$ {
               For each attribute $\alpha \in S$ {
                 $\alpha[d_i] = Generalize({v_{actual}}^{\alpha}, \lambda_S, T_{\alpha})$
                 ${V_{candidate}}^{\alpha} =$ all $v \in V_{\alpha}$ satisfying $Generalize(v, \lambda_S, T_{\alpha}) = \alpha[d_i]$
                 Find probabilities $p(v|d_i)$ for all $v \in {V_{candidate}}^{\alpha}$
                 If not $p({v_{actual}}^{\alpha}|d_i) > p(v|d_i)|\forall v \in {V_{candidate}}^{\alpha}$ Remove $\alpha$ from $S$
                 Else If $|{V_{candidate}}^{\alpha}| = 2$
                   Randomly decide whether to suppress the confidential data value and remove $\alpha$ from $S$
                     if decision = ‘not suppress’
                 Select top-k from ${V_{candidate}}^{\alpha}$ to form ${\Omega_k}^{\alpha[d_i]}$
                 Pick a random next best guess ${v_{rnbg}}^{\alpha}$ among ${\Omega_k}^{\alpha[d_i]}$
                 For each non confidential attribute $\alpha'$
                   If $\alpha'[d_i]! = \nu$ {
                     Create the maximum impact data value candidate $MIV[\alpha][\alpha']$
                     Set $MIV[\alpha'][\alpha].\alpha_{MI}$ to $\alpha'$
                     $\bar{F}_{\alpha', {v_{rnbg}}^{\alpha}, \lambda} =$ updated frequency count of tuples satisfying
                         $D[\alpha[d] = {v_{rnbg}}^{\alpha} \wedge \alpha'[d] = Generalize(\alpha'[d_i], \lambda, T_{\alpha'})]$
                     $\bar{F}_{\alpha', {v_{rnbg}}^{\alpha}, \lambda-1} =$ updated frequency count of tuples satisfying
                         $D[\alpha[d] = {v_{rnbg}}^{\alpha} \wedge \alpha'[d] = Generalize(\alpha'[d_i], \lambda - 1, T_{\alpha'})]$
                     $\bar{F}_{\alpha', {v_{actual}}^{\alpha}, \lambda} =$ updated frequency count of tuples satisfying
                         $D[\alpha[d] = {v_{actual}}^{\alpha} \wedge \alpha'[d] = Generalize(\alpha'[d_i], \lambda, T_{\alpha'})]$
                     $\bar{F}_{\alpha', {v_{actual}}^{\alpha}, \lambda-1} =$ updated frequency count of tuples satisfying
                         $D[\alpha[d] = {v_{actual}}^{\alpha} \wedge \alpha'[d] = Generalize(\alpha'[d_i], \lambda - 1, T_{\alpha'})]$
                     Set $MIV[\alpha'][\alpha].impact$ to $\frac{\bar{F}_{\alpha', {v_{rnbg}}^{\alpha}, \lambda}}{\bar{F}_{\alpha', {v_{rnbg}}^{\alpha}, \lambda-1}} \times \frac{\bar{F}_{\alpha', {v_{actual}}^{\alpha}, \lambda-1}}{\bar{F}_{\alpha', {v_{actual}}^{\alpha}, \lambda}}$
                   }
                 }
               }
               For each non confidential attribute $\alpha'$
                 Find average impact $MIV[\alpha'].average\_impact$
               Sort maximum impact attributes by $average\_impact$ in descending order
               For each maximum impact value $miv \in MIV$ {
                 Generalize the maximum impact data value $miv.\alpha_{MI}[d_i]$ by $\lambda$ levels
                 For each confidential attribute $\alpha \in S$ {
                   Update the probabilities
                   If $p(\alpha[d_i]|d_i) \leq p({v_{rnbg}}^{\alpha}|d_i)$ Remove $\alpha$ from $S$
                 }
                 If $|S| = 0$ break
               }
               If $|S| = 0$ break
               Increment $\lambda$ by 1
             }
             If $|S| = 0$ break
             Increment $\lambda_S$ by 1
           }
           If $|S| > 0$ Delete tuple $d_i$
END
```

Figure 4.10: Pseudocode of e-DROPP-G Algorithm

Chapter 5

EXPERIMENTAL RESULTS

This chapter presents the experimental results. The primary objective of the experiments is to compare the suppression algorithms in terms of their CPU time performance, rate of success, information loss, and uncertainty.

5.1 Data Sets and Implementation Details

In order to conduct the experiments we selected two data sets from the University of California at Irvine repository [2]; the Wisconsin Breast Cancer [38] and the Car Evaluation data set. Table 5.1 provides a description of the data sets including the number of instances, the number of attributes, and the number of unknowns.

We implemented the proposed algorithms using the C++ programming language. To evaluate the performance of the algorithms, we performed experiments on a 2.67 GHz Intel PC with 4GB of memory running the Windows 7 operating system. As the

Table 5.1: Data Sets Used In the Experiments

Data Set	No. of Instances	No. of Attributes	No. of Unknowns
Wisconsin Breast Cancer	699	10	16
Car Evaluation	1728	7	0

Table 5.2: Average Execution Times of Proposed Algorithms

	Average Execution Time (in ms)			
Data Set	DECP	INCP	DROPP	HID3
W. Breast Cancer	0,129	0,127	0,131	0,129
Car Evaluation	0,036	0,036	0,037	0,036
	DECP-G	INCP-G	DROPP-G	HID3-G
W. Breast Cancer	0,159	0,160	0,169	0,159

suppression algorithms contain random components, the experimental results presented are averages of five realizations unless stated otherwise. Moreover, in order to illustrate the power of the algorithms we choose to suppress confidential data values for which the domain size of the corresponding attribute is greater than 2.

5.2 Results and Analysis Of Algorithms

In this study, we first measured the average execution times required to suppress a confidential data value. In order to find the average execution times, we suppressed a data value from each instance of the data sets and averaged the CPU time results. The results, as depicted in Table 5.2, show that the suppression algorithms performed remarkably similar with respect to execution time.

Another performance criterion is the percent of successful suppressions. The suppression process is successful if and only if the confidential data value is suppressed without deleting the microdata tuple containing it. For each suppression algorithm, we first measured the percent of successful suppressions against the algorithm's primary[1] classification model. As illustrated in Table 5.3, the proposed algorithms successfully suppressed all confidential data values with respect to their primary classification model.

Next, we investigated the correctness of the following hypotheses:

[1] Naïve Bayesian classification model for DECP, INCP, DROPP, DECP-G, INCP-G, and DROPP-G algorithms, and ID3 classification model for HID3 and HID3-G algorithms.

Table 5.3: Success of Proposed Algorithms Against Different Classification Models

Data Set	Classification Model	Percent of Successful Suppressions			
		DECP	INCP	DROPP	HID3
W. Breast Cancer	Naïve Bayesian	100%	100%	100%	84%
	ID3	77%	75%	90%	100%
	SVM	41%	41%	89%	86%
Car Evaluation	Naïve Bayesian	100%	100%	100%	65%
	ID3	86%	84%	85%	100%
	SVM	65%	56%	88%	80%
		DECP-G	INCP-G	DROPP-G	HID3-G
W. Breast Cancer	Naïve Bayesian	100%	100%	100%	81%
	ID3	74%	73%	85%	100%
	SVM	40%	39%	87%	84%

Hypothesis 5.1. *Algorithms suppressing confidential data values against probabilistic classification models also block the decision tree classification based inference.*

Hypothesis 5.2. *Algorithms suppressing confidential data values against decision tree classification models also block the probabilistic classification based inference.*

Hypothesis 5.3. *Algorithms suppressing confidential data values against probabilistic or decision tree classification models also block inference based on more complex classification models(e.g. SVM).*

We measured the percent of successful suppressions achieved by each algorithm against (1) its secondary[2] classification model, and (2) SVM using the Joachim's SVM-Struct[3] [58], as it is a more powerful classification technique. As illustrated in Table 5.3, the proposed algorithms exceeded a success rate of 65% against their secondary classification models, thus proving the correctness of Hypotheses 5.1 and 5.2. For SVM, the success rates range from 39% to 89%. This proves that the proposed algorithms still protect confidential data values against more powerful classification techniques.

[2] ID3 classification model for DECP, INCP, DROPP, DECP-G, INCP-G, and DROPP-G algorithms, and Naïve Bayesian classification model for HID3 and HID3-G algorithms.

[3] Linear kernel has been used while training.

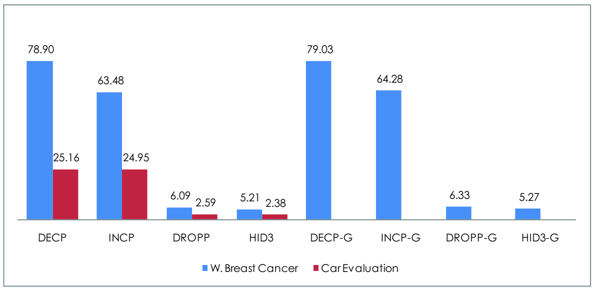
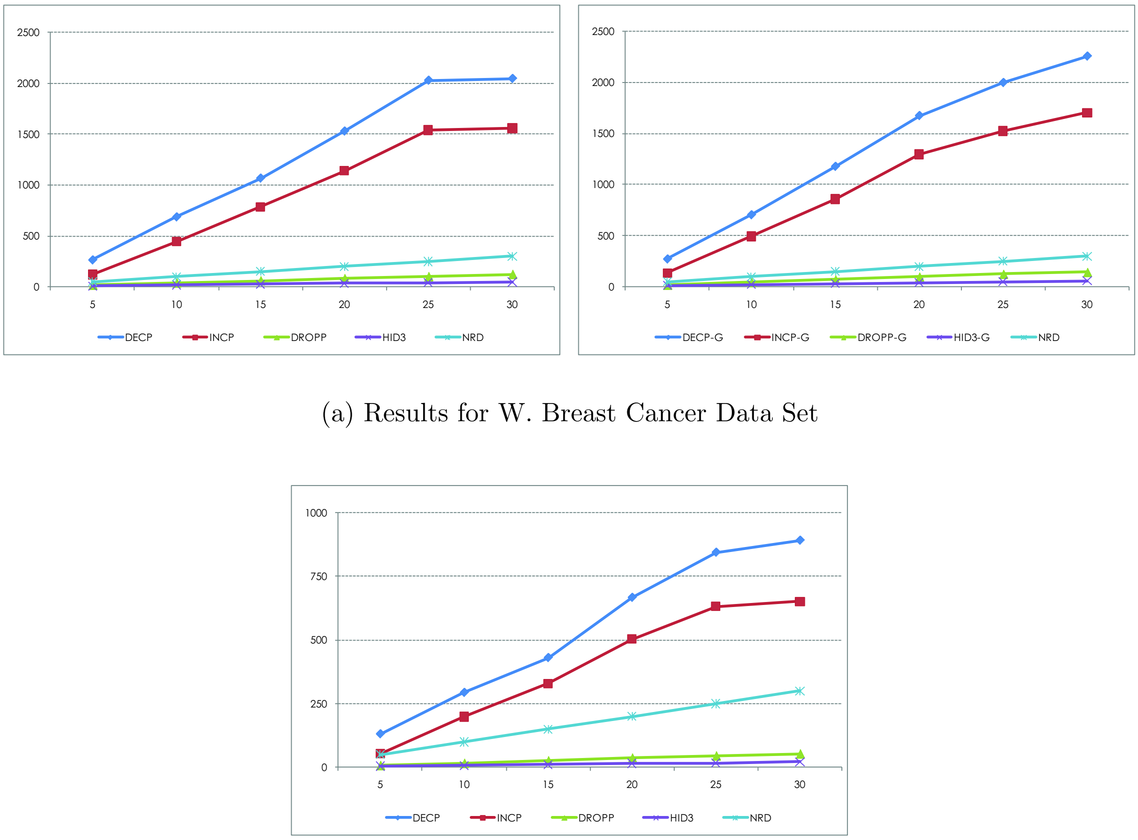


Figure 5.1: Average Direct Distance Results of Proposed Algorithms

The next performance criterion is the information loss caused by the suppression algorithms. We used three evaluation metrics in order to measure the information loss: the *Direct Distance*, *Sum of Kullback Leibler Distances*, and *Average Change in Mutual Information*. The details of these information loss metrics can be found in Section 2.4.1. As a benchmark, we used the Naïve Row Deletion (NRD) algorithm. The NRD algorithm suppresses a confidential data value via deleting the microdata tuple to which it belongs, i.e. replacing each and every data value forming the microdata tuple, including the confidential data value, with ν.

The first information loss metric we used was the average direct distance which measures the average number of changes introduced due to suppression of a single confidential data value. The average direct distance results for the suppression algorithms are shown in Figure 5.1. As can be seen from the figure, the HID3 and HID3-G algorithms cause the least amount of information loss in terms of average direct distance followed by the DROPP and DROPP-G algorithms. Actually, all of these algorithms are bounded by the NRD algorithm, as they aim at downgrading the microdata tuple instead of the classification models. On the other hand, the DECP, INCP, DECP-G and INCP-G algorithms perform relatively worse than the others, as they aim at downgrading the classification model instead. Since DECP-G, INCP-G, DROPP-G, and HID3-G aim at minimizing the level of generalization per attribute value, they perform slightly worse than their deletion counterparts.

Apart from the average direct distance, we also measured the total direct distance



(a) Results for W. Breast Cancer Data Set

(b) Results for Car Evaluation Data Set

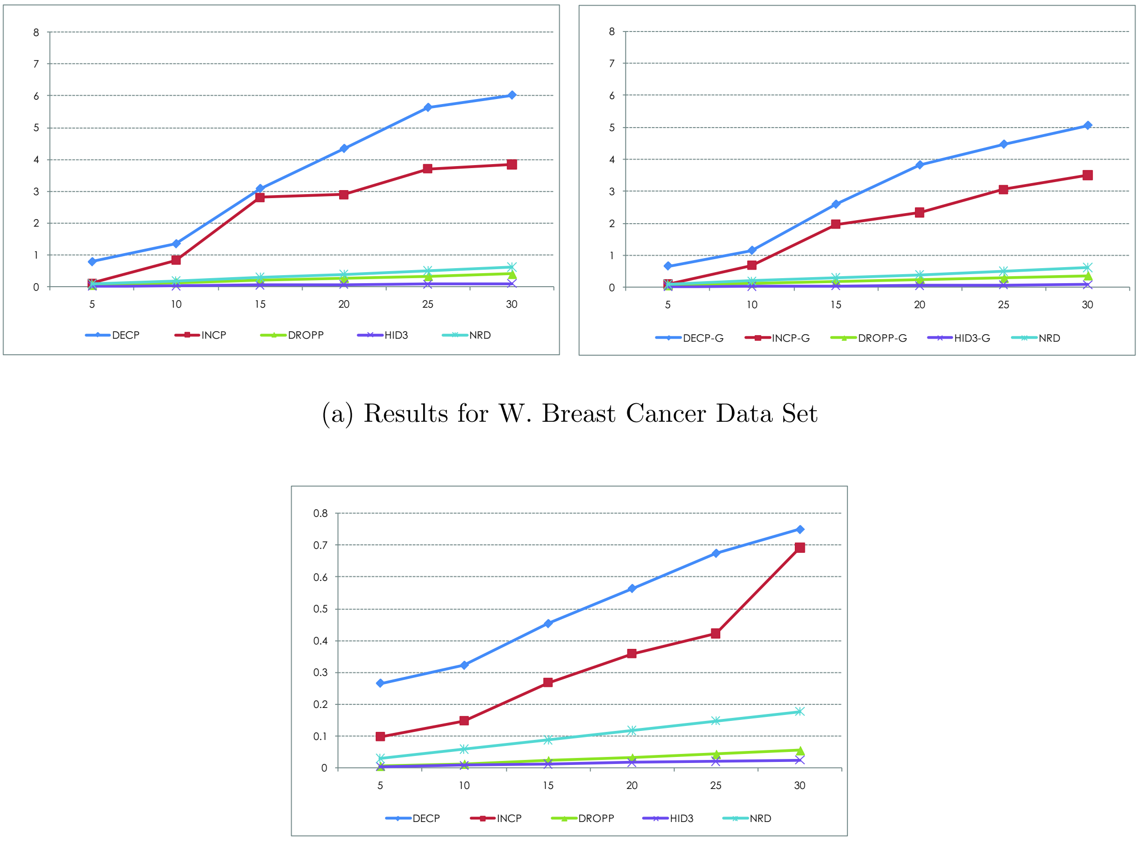
Figure 5.2: Total Direct Distance Results of Proposed Algorithms

versus the number of confidential data values suppressed. We realized this experiment for two sets of confidential data values one randomly selected from the Wisconsin Breast Cancer data set and one randomly selected from the Car Evaluation[4] data set. The same set of confidential data values are used throughout the rest of the experiments measuring the information loss and uncertainty. The results are shown in Figure 5.2. Among suppression algorithms employing deletion strategy, the HID3 algorithm causes the least amount of information loss in terms of direct distance followed by the DROPP, INCP, and DECP algorithms. The ordering is similar among suppression algorithms employing generalization strategy. However, these algorithms perform slightly worse

[4] Since the domain size of the attributes are small, this data set cannot be used to test the performance of suppression algorithms employing generalization modification strategy.

than their deletion counterparts.

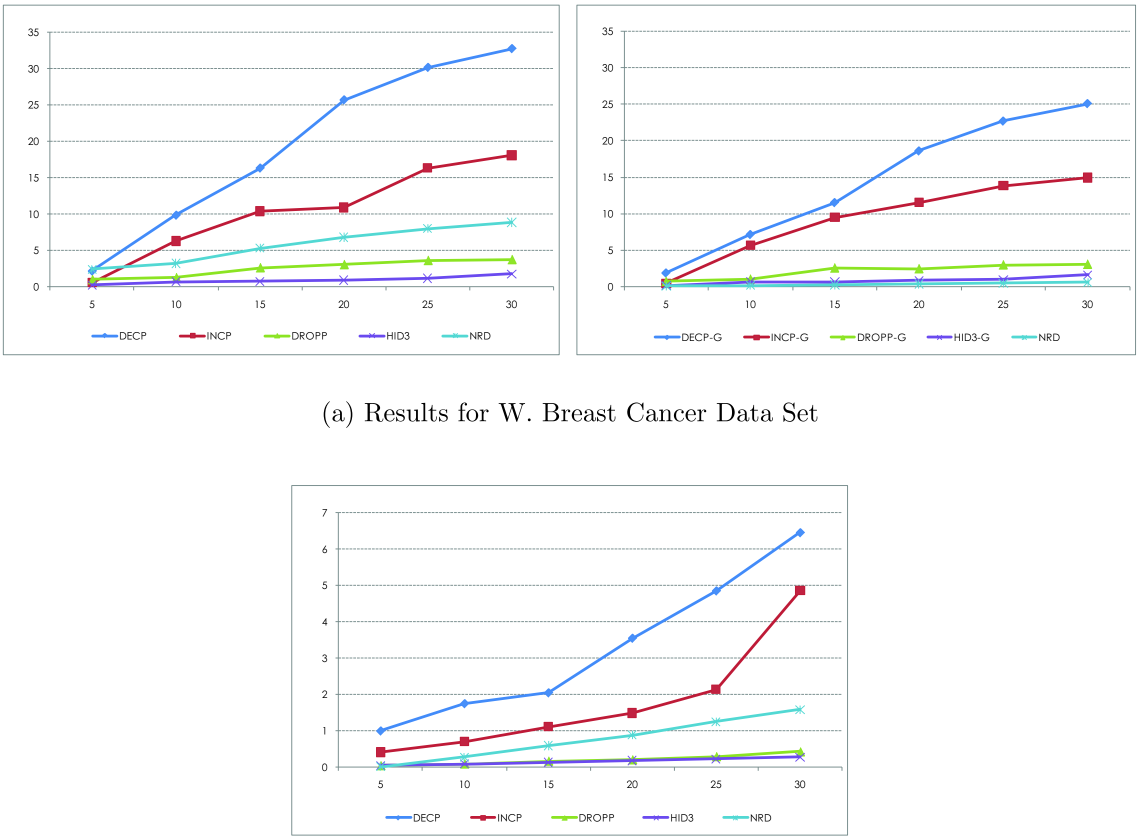
The second information loss metric we used was the sum of Kullback Leibler distances which measures the distance between the first order probability distributions of the original and the new data sets. The performance of suppression algorithms in terms of sum of Kullback Leibler distances is shown in Figure 5.3. Among suppression algorithms employing deletion strategy, the HID3 algorithm causes the least amount of information loss in terms of sum of Kullback Leibler distances followed by the DROPP, INCP, and DECP algorithms. Unlike the direct distance, the algorithms employing generalization modification strategy perform better than their deletion counterparts with respect to sum of Kullback Leibler distances.



(a) Results for W. Breast Cancer Data Set

(b) Results for Car Evaluation Data Set

Figure 5.3: Sum of Kullback Leibler Distance Results of Proposed Algorithms

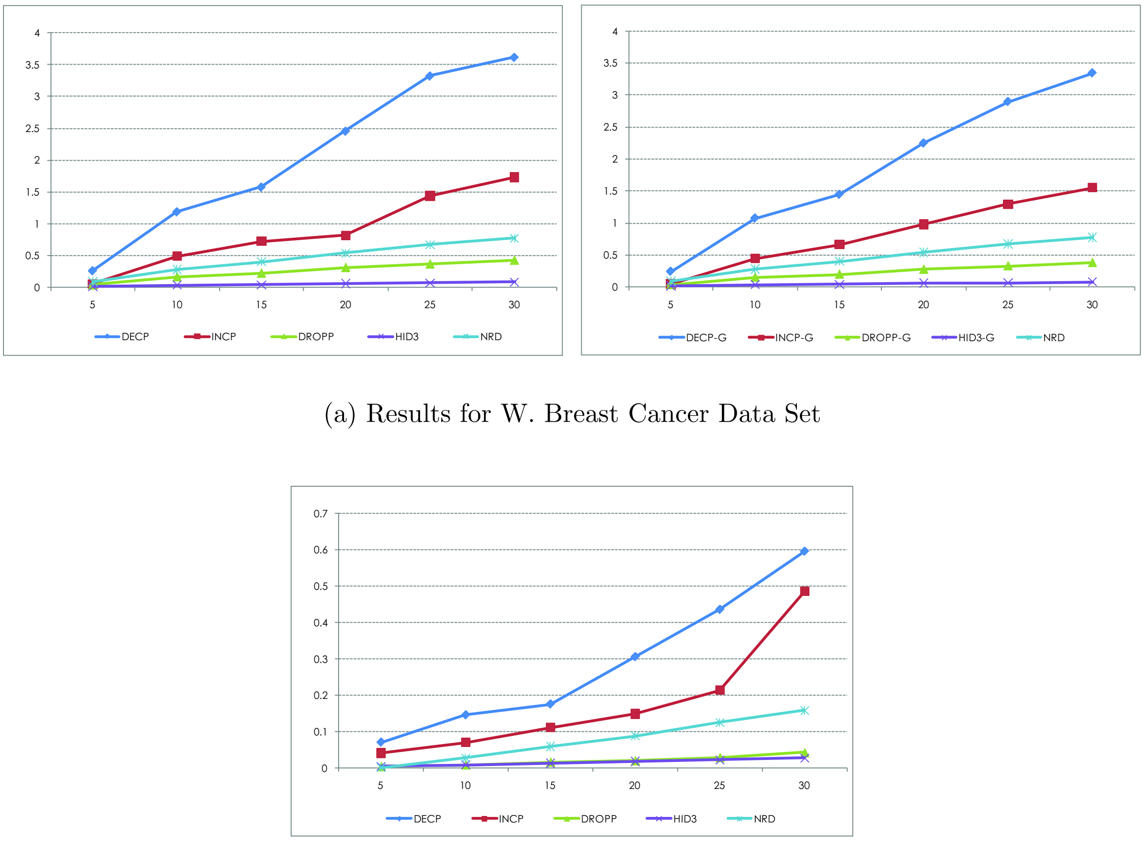


(a) Results for W. Breast Cancer Data Set

(b) Results for Car Evaluation Data Set

Figure 5.4: Average Change in Mutual Information Results of Proposed Algorithms

The last information loss metric we used was the average change in mutual information which measures the average distance between the second order probability distributions of the original and modified data sets. The performance of suppression algorithms in terms of average change in mutual information is shown in Figure 5.4. The results show that; (1) the HID3-G and HID3 algorithms distort correlations least followed by the DROPP-G, DROPP, INCP-G, INCP, DECP-G, and DECP algorithms, and (2) algorithms downgrading the classification model prevent inference of confidential data values better than the ones downgrading the tuple itself, as they distort correlations within the data sets, (3) algorithms employing generalization modification strategy perform better than their deletion counterparts.



(a) Results for W. Breast Cancer Data Set

(b) Results for Car Evaluation Data Set

Figure 5.5: Sum of Conditional Entropy Results of Proposed Algorithms

The final performance criterion is the uncertainty introduced by the suppression algorithms. We used the sum of conditional entropies in order to measure the expected value of uncertainty introduced into the modified data sets. The performance of suppression algorithms in terms of sum of conditional entropies is shown in Figure 5.5. The results show that; (1) the HID3-G and HID3 algorithms introduce the least uncertainty followed by the DROPP-G, DROPP, INCP-G, INCP, DECP-G, and DECP algorithms, and (2) algorithms downgrading the classification model prevent inference of confidential data values better than the ones downgrading the tuple itself, as they cause more uncertainty within the data sets, (3) algorithms employing generalization modification strategy introduce less uncertainty than their deletion counterparts.

We can summarize the presented experimental results as follows:

1. There is a tradeoff between the rate of successful suppressions and the information loss caused by the suppression process.

2. The DECP and DECP-G algorithms achieve the highest success rate while causing the highest amount of information loss and uncertainty. This justifies Lemma 3.1 and Lemma 4.1 which state that the upper bound for the number of data values that can be modified by these algorithms is equal to $(n-1)(N-1) < nm$.

3. The INCP and INCP-G algorithms achieve the second highest success rate while causing the second highest information loss and uncertainty. It is followed by the DROPP, DROPP-G, HID3 and HID3-G algorithms. This ordering is completely due to (1) the characteristics of the Wisconsin Breast Cancer and Car Evaluation data sets satisfying the inequality $m >> n$, i.e. the number of transactions is much more than the number of attributes, and (2) the upper bounds for the number of data values that can be modified by the algorithms. For a data set satisfying the inequality $n >> m$, the order of success, information loss, and uncertainty will be reversed.

4. The success rate of algorithms employing generalization strategy is similar to their deletion counterparts.

5. The information loss caused by algorithms employing generalization strategy is less than their deletion counterparts. The only exception to this is the direct distance: algorithms employing generalization strategy change more attribute values than the one employing deletion strategy. Since DECP-G, INCP-G, DROPP-G, and HID3-G aim at minimizing the level of generalization per attribute value, they perform slightly worse than their deletion counterparts in terms of direct distance.

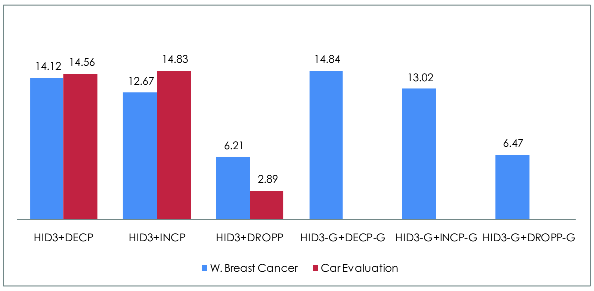


Figure 5.6: Average Direct Distance Results of Hybrid Algorithms

5.3 Results and Analysis of Hybrid Algorithms

In this study, we merged each Naïve Bayesian suppression algorithm with the Decision Tree suppression algorithms in a round robin fashion[5], to demonstrate the performance of the hybrid algorithms against both classification models.

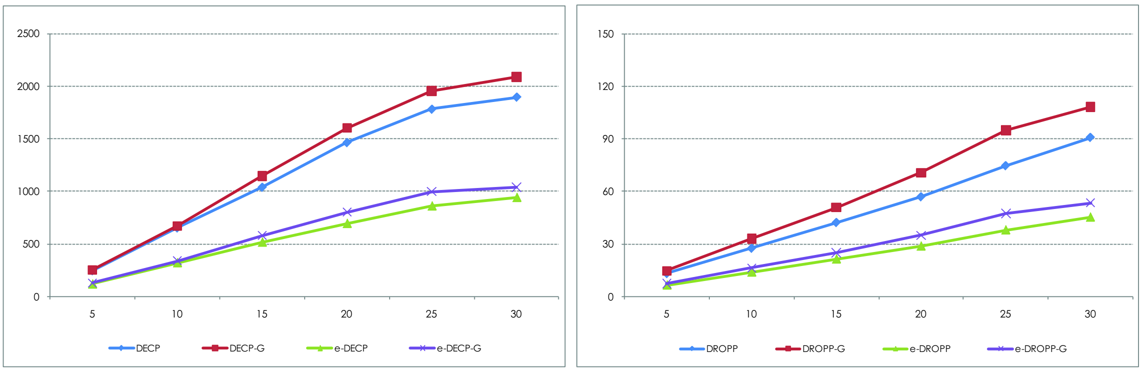
First, we measured the percent of successful suppressions achieved by each hybrid algorithm against both classification models. The hybrid algorithms successfully suppressed all confidential data values with respect to both classification models, and achieved 100% success rate. Next, we measured the average direct distance for the hybrid algorithms as shown in Figure 5.6. The HID3+DROPP and HID3-G+DROPP-G algorithms cause the least amount of information loss in terms of average direct distance followed by the HID3+INCP, HID3-G+INCP-G, HID3+DECP, and HID3-G+DECP-G algorithms.

5.4 Results and Analysis of Enhanced Algorithms

In this study, we also enhanced the DECP, DECP-G, DROPP, and DROPP-G algorithms to suppress multiple confidential data values, and thus introduce less side effects.

[5] For example, for HID3+DECP hybrid algorithm, first the HID3 algorithm is executed to suppress the confidential data value against decision tree classification based inference. Then, the DECP algorithm is executed to suppress the confidential data value against probabilistic classification based inference.

In order to demonstrate the performance of the enhanced algorithms compared to the original ones, we used two new sets of randomly chosen confidential data values. The first set of confidential data values is used to compare the performance of DECP variants, while the second set is used to compare the performance of DROPP variants. Using these two sets we measured the total number of changes introduced due to suppression of multiple confidential data values. As shown in Figure 5.7, the enhanced algorithms performed remarkably better than the original versions, and reduced the side-effects by nearly 50%.



(a) Results for DECP Variants

(b) Results for DROPP Variants

Figure 5.7: Total Direct Distance Results of Enhanced Algorithms on W. Breast Cancer Data Set

Chapter 6

SUMMARY AND CONCLUSION

In tandem with the advances in networking and storage technologies, the private sector as well as the public sector has increased their efforts to gather, manipulate, and commodify information on a large scale. These pervasive data harvesting efforts coupled with the increasing need to share the data with other institutions or with public raised concerns about privacy. Widespread usage of powerful data analysis tools and data mining techniques, enabling institutions to extract previously unknown and strategically useful information from huge collections of data sets, and thus gain competitive advantages, has also contributed to the fears about privacy. Data mining techniques can be used for many reasons including but not limited to national security warning and national security decision making [1] for government agencies, and providing better business intelligence and customer relationship management for enterprises. On the other hand, they can also be used by adversaries to infer hidden confidential information about individuals from the disclosed data sets, and thus pose a great threat to privacy.

In this dissertation, we have precisely formulated the problem of suppressing a confidential data item, and designed algorithms to avoid probabilistic and decision tree classification based inference. We have selected Naïve Bayesian and ID3 as typical representatives of probabilistic and decision tree classifiers respectively, and developed our algorithms accordingly. More specifically we have designed and implemented the following algorithms:

1. The DECP and INCP algorithms suppress a single confidential data value against Naïve Bayesian classifiers. These two algorithms downgrade the Naïve Bayesian classifiers using the deletion modification strategy. Details of DECP and INCP algorithms are also presented in [7;8].
2. The DROPP and HID3 algorithms suppress a single confidential data value against Naïve Bayesian and ID3 classifiers respectively. These two algorithms downgrade the microdata tuple containing the confidential data value using the deletion modification strategy. Details of DROPP and HID3 algorithms are also presented in [7;8].
3. The DECP-G and INCP-G algorithms suppress a single confidential data value against Naïve Bayesian classifiers. These two algorithms downgrade the Naïve Bayesian classifiers using the generalization modification strategy.
4. The DROPP-G and HID3-G algorithms suppress a single confidential data value against Naïve Bayesian and ID3 classifiers respectively. These two algorithms downgrade the microdata tuple containing the confidential data value using the generalization modification strategy.
5. The e-DECP and e-DROPP algorithms suppress multiple confidential data values against Naïve Bayesian classifiers using the deletion modification strategy.
6. The e-DECP-G and e-DROPP-G algorithms suppress multiple confidential data values against Naïve Bayesian classifiers using the generalization modification strategy.

Our experimental results presented in this dissertation have shown that:

- The proposed algorithms are able to suppress confidential data values, so that they cannot be predicted by their target classification models.

- The proposed algorithms are also able to block the inference channels introduced by other classification models.

- The hybrid versions of the algorithms are able to block the inference channels introduced by Naïve Bayesian and ID3 classifiers with substantially less side effects.

- Similarly, the generalization versions of the algorithms distort the microdata less when compared to their deletion counterparts.

- The enhanced versions of the algorithms are able to suppress multiple confidential data values and reduce the side effects by 50%.

6.1 Future Work

Some promising directions for future work include:

1. Development of suppression algorithms to avoid different classification algorithms based inference. Specifically, it would be interesting to design variants of suppression algorithms against Bag-of-words classifiers, Bayesian Networks, Logistic Regression Classifiers, and Hyperplane classifiers (Perceptron, Winnow Perceptron, and Support Vector Machines).

2. Development of algorithms to suppress multiple confidential data values that are not necessarily instances of the same attribute, or do not have the same value, or do not belong to the same tuple.

3. Development of a generic suppression technique which employs information theoretic concepts for distorting the microdata. Such a technique should be able to avoid any classification algorithm based inference.

4. Development of suppression algorithms to handle evolving (i.e. continuously updated) microdata.

5. Development of suppression algorithms that can hide confidential data values from horizontally or vertically distributed microdata.

Bibliography

[1] Report to Congress regarding the Terrorism Information Awareness Program, May 20, 2003.

[2] UCI Machine Learning Repository, *http://www.ics.uci.edu/ mlearn/MLSummary.html.*

[3] USC Annenberg School – Center for the Digital Future, The Highlights of the Digital Future Report, Year Five, Ten Years Ten Trends, *Available at http://www.digitalcenter.org/pdf/Center-for-the-Digital-Future-2005-Highlights.pdf.*

[4] Wikipedia, Privacy – Wikipedia, the free encyclopedia, *Available at http://en.wikipedia.org/wiki/Privacy*, 2005.

[5] N. R. Adam, J. C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Survey*, vol. 21, no. 4, p. 515-556, 1989.

[6] C. Aggarwal, On k-Anonymity and the Curse of Dimensionality, In *Proceedings of the 31st VLDB conference*, 2005.

[7] A. Azgın Hintoğlu, Y. Saygın, Suppressing microdata to prevent classification based inference, *VLDB Journal*, vol.19, no.3, June 2010.

[8] A. Azgın Hintoğlu, Y. Saygın, Suppressing microdata to prevent probabilistic classification based inference, In *Proceedings of the Workshop on Secure Data Management (SDM'05)*, 2005.

[9] A. Brodsky, C. Farkas, A. Jajodia, Secure databases: Constraints, inference channels and monitoring disclosure, *IEEE Trans. Knowledge and Data Engineering*, vol. 12, no. 6, p. 900-919, 2000.

[10] L. Chang, I.S. Moskowitz, Parsimonious Downgrading and Decision Trees Applied to the Inference Problem, *Proceedings of the Workshop of New Security Paradigms*, p. 82-89, 1999.

[11] T.M. Cover, J.A. Thomas, Elements of Information Theory, *John Wiley & Sons*, 1991.

[12] L. H. Cox, , Suppression methodology and statistical disclosure control, *J. Am. Stat. Assoc.*, vol. 75, no. 370, p. 377-385, 1980.

[13] S. Dawson, S. D. C. di Vimercati, P. Lincoln, P. Samarati, Minimal data upgrading to prevent inference and association, In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ACP Press, p. 114-125, 1999.

[14] H. Delugach, T. Hinke, Wizard: A database inference analysis and detection system, *IEEE Trans. Knowledge and Data Engineering*, vol. 8, no. 1, p. 56-66, 1996.

[15] D. E. Denning, Commutative filters for reducing inference threats in multilevel database systems, In *Proceedings of IEEE Symposium on Security and Privacy*, p. 134-146, 1985.

[16] D. E. Denning, *Cryptography and Data Security*, Addison-Wesley, 1982.

[17] J. Domingo-Ferrer (editor), Inference Control in Statistical Databases, *Lecture Notes in Computer Science*, vol. 2316, Berlin: Springer-Verlag, 2002.

[18] J. Domingo-Ferrer, A. Solanas, A. Martinez-Balleste Privacy in Statistical Databases:k-Anonymity Through Microaggregation, In *Proceedings of IEEE Granular Computing*, 2006.

[19] J. Domingo-Ferrer, V. Torra, Ordinal, Continious and Heteregoneous k-Anonymity through Microaggregation, *Data Mining and Knowledge Discovery*, vol. 11, no.2, p. 195-212, 2005. l,,

[20] J. Domingo-Ferrer, V. Torra, Practical Data-Oriented Microaggregation for Statistical Disclosure Control, *IEEE Transaction on Knowledge and Data Engineering*, vol. 14, no. 1, p. 189-201, 2002.

[21] S. Dreiseitl, S. Vinterbo, L. Ohno-Machado, Disambiguation Data: Extracting Information from Anonymized Sources, In *Proceedings of the 2001 American Medical Informatics Annual Symposium*, p. 144-148, 2001.

[22] C. Farkas, S. Jajodia, The inference problem: A survey, *SIGKDD Explorations*, 2003.

[23] M. Fischetti, J.J. Salazar, Models and Algortihms for the 2-Dimensional Cell Suppression Problem in Statistical Disclosure Control, *Mathematical Programming*, p. 283-312, 1999.

[24] M. Fischetti, J.J. Salazar, Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints, *Journal of American Statistical Association*, Vol. 95, No. 451, p. 916-928, 2000.

[25] J. Geurts, Heuristics for cell suppression in tables, *Technical Paper, Netherlands Central Bureau of Statistics*, 1992.

[26] S.L. Hansen, S. Mukherjee, A Polynomial Algorithm for Optimal Univariate Microaggregation, *IEEE Transaction on Knowledge and Data Engineering*, vol.15, no. 4, p. 1043-1044, 2003.

[27] T.H. Hinke, H.S. Delugach, A. Chandrasekhar, A fast algorithm for detecting second paths in database inference analysis, *Journal of Computer Security* , vol. 3, no. 2, 3, p. 147-168, 1995.

[28] T.H. Hinke, H.S. Delugach, R.P. Wolf Protecting databases from inference attacks, *Computers and Security*, vol. 16, no. 8, p. 687-708, 1997.

[29] V. S. Iyengar, Transforming data to satisfy privacy constraints, In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[30] S. Jajodia, C. Meadows, Inference Problems in Multilevel Secure Database Management Systems, *Information Security-An Integrated Collection of Essays*, M.D. Abrams, S. Jajodia, and H.J. Podell, eds., p. 570-584, IEEE C. S. Press, 1989.

[31] M.Y. Kao, Data security equals graph connectivity, *SIAM Journal on Discrete Mathematics*, vol. 9, p. 87-100, 1996.

[32] J.P. Kelly, B.L. Golden, A.A. Assad, Cell suppression:Disclosure protection for sensitive tabular data, *Networks*, vol.22, p. 397-417, 1992.

[33] W. Klosgen, Knowledge discovery in databases and data privacy, *IEEE Expert*, April 1995.

[34] M. Laszlo, S. Mukherjee, Minimum Spanning Tree Partitioning Algorithm for Microaggregation, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, p. 902-911, 2005.

[35] N. Li, T. Li, t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the 23nd International Conference on Data Engineering (ICDE'07)*, 2007.

[36] A. Martinez-Balleste, A. Solanas, J. Domingo-Ferrer, J. M. Mateo-Sanz, A Genetic Approach to Multivariate Microaggregation for Database Privacy, In *Proceedings of 23rd IEEE Internation Conference on Data Engineering*, p. 180-185, 2007.

[37] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, ℓ-Diversity: Privacy Beyond k-Anonymity, In *Proceedings of the 22nd IEEE International Conference on Data Engineering*, 2006.

[38] O. L. Mangasarian, W. H. Wolberg, Cancer diagnosis via linear programming, *SIAM News*, vol. 23, no. 5, p. 1-18, 1990.

[39] D. Marks, Inference in MLS database systems, *IEEE Trans. Knowledge and Data Engineering*, vol. 8, no. 1, p. 46-55, 1996.

[40] H. Nissenbaum, Protecting Privacy in an Information Age: The Problem of Privacy in Public, *Law and Philosophy*, vol. 17, p. 559-596, 1998.

[41] A. Oganian, J. Domingo-Ferrer, On the Complexity of Optimal Microaggregation for Statistical Disclosure Control, *Statistical Journal of United Nations Economic Commission for Europe*, vol. 18, no. 4, p. 345-354, 2001.

[42] A. Øhrn, L. Ohno-Machado, Using Boolean Reasoning to Anonymize Databases, *Artificial Intelligence in Medicine*, vol. 15, no. 3, p. 235-254, 1999.

[43] D.E. O'Leary, Knowledge discovery as a threat to database security, In G. Piatetsky-Shapiro and W. Frawley, editors, *Knowledge Discovery in Databases*, p. 507-516, AAAI Press/The MIT Press, Menlo Park, California, 1991.

[44] D.E. O'Leary, Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines, *IEEE Expert: Intelligent Systems and Their Applications*, vol. 10, no. 2, p. 48-52, April 1995.

[45] G. Piatetsky-Shapiro, Knowledge discovery in databases vs. personal privacy, *IEEE Expert*, April 1995.

[46] X. Quian, M.E. Stickel, P.D. Karp, T.F. Lunt, and T.D. Garvey, Detection and Elimination of Inference Channels in Multilevel Relational Database Systems, In *Proceedings of IEEE Symp. Security and Privacy*, p. 196-205, 1993.

[47] P. Samarati, L. Sweeney, Protecting Privacy when Disclosing Information: k-anonymity and its Enforcement through Generalization and Suppression, *IEEE Symposium on Research in Security and Privacy*, 1998.

[48] P. Samarati, Protecting Respondents' Identities in Microdata Release, *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no.6, p. 1010-1027, Nov. 2001.

[49] G. Sande, Automated cell suppression to reserve confidentiality of business statistics, In *Proceedings of the 2nd International Workshop on Statistical Database Management*, p. 346-353, 1983.

[50] G. Sande, Exact and approximate methods for data directed microaggregation in one or more dimensions, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, p. 459-476, 2002.

[51] P. Selfridge, Privacy and knowledge discovery in databases, *IEEE Expert*, April 1995.

[52] A. Solanas, A. Martinez-Balleste, J. M. Mateo-Sanz, J. Domingo-Ferrer, Towards Microaggregation with Genetic Algorithms, In *Proceedings of the Third IEEE Conference on Intelligent Systems*, p. 65-70, 2006.

[53] P. Stachour, B. Thuraisingham, Design of LDV: A multilevel secure relational database management system, *IEEE Trans. Knowledge and Data Engineering*, vol. 2, no. 2, p. 190-209, 1990.

[54] T. Su, G. Ozsoyoglu, Inference in MLS database systems, *IEEE Trans. Knowledge and Data Engineering*, vol. 3, no. 2-3, p. 147-168, 1991.

[55] L. Sweeney, k-Anonymity: A model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, p. 557-570, 2002.

[56] L. Sweeney, Information Explosion, *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, L. Zayatz, P. Doyle, J. Theeuwes and J. Lane (eds), Urban Institute, Washington, DC, 2001.

[57] B. Thuraisingham, Security checking in relational database management systems augmented with inference engines, *Computers and Security*, vol. 6, p. 479-492, 1987.

[58] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, Large Margin Methods for Structured and Interdependent Output Variables, *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453-1484, 2005.

[59] V. Torra, Microaggregation for categorical variables: a median based approach, In J. Domingo-Ferrer and V. Torra, editors, *Privacy in Statistical Databases*, vol. 3050, p. 162174, 2004.

[60] K. Wang, B.C.M. Fung, P. S. Yu, Template-Based Privacy Preservation in Classification Problems, *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, p. 466-473, 2005.

[61] L. Willenborg, T. De Waal, Statistical Disclosure Control in Practice, *Lecture Notes in Statistics*, vol. 111, Springer Verlag, New York, 1996.

[62] J. Zhang, V. Honavar, AVT-NBL: An Algorithm for Learning Compact and Accurate Naive Bayes Classifiers from Attribute Value Taxonomies and Data., *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, p. 289-296, 2004.