

# **PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS**

A Thesis Submitted to  
Nirma University  
In Partial Fulfillment of the Requirements for  
The Degree of  
Doctor of Philosophy  
in  
Technology and Engineering

By:  
Kotecha Radhika (12EXTPHDE88)

Under the guidance of:  
Dr. Sanjay Garg  
(Professor, Computer Engineering Department)



Institute of Technology  
Nirma University  
Ahmedabad - 382481  
Gujarat, India  
April 2017

# Nirma University Institute of Technology

## Certificate

This is to certify that the thesis entitled "Privacy-Preserving Classification of Horizontally Partitioned Data Streams" has been prepared by Ms. Kotecha Radhika N. under my supervision and guidance. The thesis is her own original work completed after careful research and investigation. The work of the thesis is of the standard expected of a candidate for Ph.D. Programme in Computer Science and Engineering and I recommend that it be sent for evaluation.

Date: 5.04.2017

 Dr. Sanjay Garg


Signature of Guide

---

Forwarded Through:

 Dr. Sanjay Garg

(i) Name and signature of the Head of the Department

 Dr. Alka Mahajan

(ii) Name and signature of the Dean Faculty of Technology and Engineering

 Dr. M. Gupta

(iii) Name and signature of the Dean Faculty of Doctoral Studies and Research

To:

The Executive Registrar,

Nirma University

# Nirma University Institute of Technology

## Declaration

I, Kotecha Radhika Navinbhai, registered as Research Scholar, bearing Registration No. 12EXTPHDE88 for Doctoral Programme under the Faculty of Technology and Engineering of Nirma University do hereby declare that I have completed the course work, pre-synopsis seminar and my research work as prescribed under R. Ph.D. 3.5.

I do hereby declare that the thesis submitted is original and is the outcome of the independent investigations / research carried out by me and contains no plagiarism. The research is leading to the discovery of new techniques. This work has not been submitted to any other University or Body in quest of a degree, diploma or any other kind of academic award.

I do hereby further declare that the text, diagrams or any other material taken from other sources (including but not limited to books, journals and web) have been acknowledged, referred and cited to the best of my knowledge and understanding.

Date: 05/04/2017



Kotecha Radhika N. (12EXTPHDE88)

I endorse the above declaration made by the student.



Dr. Sanjay Garg (Guide)

# Abstract

With technological advancements, several real-world applications generate massive amount of data. Such data, also known as data streams, are continuously arriving at an unprecedented rate and contain valuable knowledge. Due to their effectiveness in supporting decision-making processes and knowledge discovery, the data mining techniques have attracted considerable interest and attention of research communities. Extracting patterns from such voluminous data streams requires development of new algorithms or modifications in the traditional data mining algorithms. In recent years, data stream classification has been an active area of research in data stream mining and is the focus of this work.

It is apparent that the power of these mining techniques may breach the privacy of individuals to whom the data refers and the field of privacy-preserving data mining (PPDM) has emerged in response to this issue. Specifically, PPDM techniques aim to perform a trade-off between efficiency in data mining and exposure (direct or via inference) of sensitive information in the original data. Further, not only the original data but also the data mining output can lead to disclosure of sensitive information. But when the data mining output reveals no private patterns, it can be reliably claimed that the privacy of underlying data is protected. Specifically, when the final goal is to release the output of data mining (a model), its effectiveness in preserving privacy is of the utmost concern. This research work focuses on preserving output-privacy, that is, on preventing inference using the released classifier.

But, data stream classification and privacy-preservation are two conflicting goals because the data stream classifier should be ready to predict at any point and has memory limitations whereas privacy-preserving methods may require multiple scans over the data. Hence, the crucial issue of privacy-preserving data stream classification (PPDSC) is emerging as a novel research area. This work proposes a systematic method named Diverse and Anonymized Hoeffding Tree (DAHOT) to address this issue. The algorithm uses Hoeffding tree as a base classifier for classifying data streams and a variant of  $k$ -anonymity as well as  $l$ -diversity principles to preserve the privacy of the output classifier.

Further, advancement in networking technologies has triggered mining of distributed data. Different organizations (data holders) want to undertake a joint data mining task to obtain certain global patterns. Such collaboration is essential because of the mutual benefits it brings. However, free sharing of data is restricted due to privacy and security concerns, leading to the need of privacy-preserving distributed data mining. The work focuses on horizontally partitioned (homogeneously distributed) data as numerous applications fall under this data model.

Since the work presented in this thesis targets classification of data streams, the emerged problem is framed as privacy-preserving classification of horizontally partitioned data streams. Several applications from diverse domains like credit-card fraud detection, disease outbreak detection, loan approval, etc. are examples of privacy-preserving classification of horizontally partitioned data streams.

As a solution, a novel framework is proposed in this thesis, where each participating site (data holder) induces a DAHOT classifier and third-party combines these local classifiers to form a global classifier. No private information is to be disclosed to the merger site too. Within this framework, a method named DAHOT-GPeCT is proposed that uses Genetic Programming (GP) for induction of a global classifier at the merger site from the local DAHOT classifiers induced by participating parties. Furthermore, a method named DAHOT-GPeCT-Ensemble, which is an extension of DAHOT-GPeCT is proposed. DAHOT-GPeCT-Ensemble uses a combination of GP and Ensemble learning to obtain a global privacy-preserving classifier from horizontally partitioned data streams.

Experimental results on synthetic and real data streams indicate that the proposed approaches have been effective in accurately classifying the horizontally partitioned data streams while preserving the required privacy.

Keywords: Classification, Data Streams, Output-Privacy-preservation, Horizontal Partitioning, Anonymization, Genetic Programming, Ensemble Learning.

# Acknowledgements

It gives me an immense pleasure to present this research work related to Privacy-preserving classification of horizontally partitioned data streams.

For the same, I owe the deepest gratitude to my guide Dr. Sanjay Garg, Professor and Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad. His treasure of knowledge has provided me a strong base and direction towards this work. His extremely admirable mentorship, continuous support, motivation, advises and the trust he showed in me are just few of the uncountable reasons to thank him.

I would like to convey my sincere thanks to my Research Progress Committee members, Dr. Suman Mitra and Dr. Aruna Tiwari for their valuable suggestions to improve my work.

I consider myself fortunate to have pursued my doctoral studies from Nirma University and am thankful for the opportunity. Dr. Alka Mahajan, the Director of Institute of Technology, Nirma University is a great source of inspiration.

I specially thank the management and higher authorities of my workplace V.V.P. Engineering College, Rajkot for their kind support. Exceptional thanks to all my colleagues for their boundless cooperation, always.

Words fail to appreciate the constant support given by my mother and father throughout the journey. I feel blessed to have extremely understanding parents, parents-in-law, siblings and the family in entirety.

I express a heartfelt thanks to my husband, Mr. Abhishek Munshaw for the unconditional cooperation and abundant patience which only he can have.

Finally, thank you almighty God, you and your blessings are omnipresent in all my accomplishments.

**Kotecha Radhika N.**

**12EXTPHDE88**



# Publications related to Thesis

- Radhika Kotecha and Sanjay Garg, “Data Streams and Privacy: Two Emerging Issues in Data Classification”, in Proceedings of 5<sup>th</sup> Nirma University International Conference on Engineering, pp. 1-6, IEEE (2015)
- Radhika Kotecha and Sanjay Garg, “Genetic Programming based Evolution of Classification Trees for Decision Support in Banking Sector”, in International Journal of Knowledge Engineering and Soft Data Paradigms, vol. 5, nos. 3/4, pp. 186-204, Inderscience Publishers (2016).
- Radhika Kotecha and Sanjay Garg, “Preserving Output-Privacy in Data Stream Classification”, in Progress in Artificial Intelligence, vol. 6, nos. 2, pp. 87-104, Springer (2017).
- Sanjay Garg and Radhika Kotecha, “DAHOT-GPeCT Ensemble for Output-Privacy-Preserving Classification of Horizontally Partitioned Data Streams”, in communication with IEEE Transactions on Cybernetics (2017).

# Contents

Certificate	ii
Declaration	iii
Abstract	iv
Acknowledgements	vii
Publications related to Thesis	ix
List of Figures	xiii
List of Tables	xv
Abbreviations	xvi
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	4
1.3 Scope of the Work . . . . .	5
1.4 Significant Contributions . . . . .	6
1.5 Thesis Organization . . . . .	7
<b>2 Background</b>	<b>9</b>
2.1 Data Stream Classification . . . . .	9
2.1.1 Data stream pre-processing . . . . .	13
2.1.2 Data stream classification techniques . . . . .	14
2.2 Privacy-Preserving Data Classification . . . . .	19
2.2.1 Privacy-preserving data classification techniques . . . . .	22
2.3 Genetic Programming . . . . .	31
2.4 Ensemble Learning . . . . .	35
2.4.1 Ensemble methods . . . . .	36

<b>3</b>	<b>Empirical Evaluation of Preliminary Components</b>	<b>39</b>
3.1	Data Stream Classification . . . . .	39
3.1.1	Data streams . . . . .	41
3.1.2	Implementation details . . . . .	42
3.1.3	Results . . . . .	43
3.2	Privacy-preserving Classification of Horizontally Partitioned Data . . . . .	46
3.2.1	Data sets . . . . .	47
3.2.2	Implementation details . . . . .	48
3.2.3	Results . . . . .	49
3.3	Summary . . . . .	52
<b>4</b>	<b>Proposed Framework</b>	<b>53</b>
4.1	Targeted Application . . . . .	53
4.2	Data Mining for Credit Decision-Making . . . . .	55
4.3	Generic View of Proposed Approach . . . . .	57
4.4	Implementation Environment Details . . . . .	59
4.4.1	Implementation of classifier sharing process . . . . .	60
4.5	Summary . . . . .	60
<b>5</b>	<b>Preserving Output-Privacy in Data Stream Classification</b>	<b>65</b>
5.1	Introduction . . . . .	65
5.2	Related Work . . . . .	67
5.3	Proposed Output-Privacy-Preserving Data Stream Classifier . . . . .	71
5.3.1	Proposed algorithm . . . . .	73
5.3.2	Discretization of data streams . . . . .	77
5.4	Experimental Details . . . . .	79
5.4.1	Data streams . . . . .	79
5.4.2	Baseline methods for comparison . . . . .	80
5.4.3	Evaluation criteria . . . . .	81
5.4.4	Evaluation methods . . . . .	82
5.5	Results and Discussion . . . . .	84
5.5.1	Empirical comparison of discretization methods . . . . .	84
5.5.2	Evaluation using periodic hold-out method . . . . .	85
5.5.3	Evaluation using traditional hold-out method . . . . .	96
5.6	Summary . . . . .	99
<b>6</b>	<b>Genetic Programming-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams</b>	<b>101</b>
6.1	Introduction . . . . .	101
6.2	Related work . . . . .	103
6.2.1	Application of genetic programming for classifier optimization . . . . .	104
6.3	Proposed Approach . . . . .	106
6.3.1	Proposed algorithm . . . . .	106

6.4	Experimental Evaluation and Analysis . . . . .	113
6.4.1	Data streams at participating sites . . . . .	113
6.4.2	Data streams at merger site . . . . .	114
6.4.3	Baseline methods for comparison . . . . .	116
6.4.4	Implementation details . . . . .	117
6.5	Results and Discussion . . . . .	118
6.6	Summary . . . . .	124
<b>7</b>	<b>Ensemble-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams</b>	<b>127</b>
7.1	Introduction . . . . .	127
7.2	Related Work . . . . .	130
7.3	An Improved Fitness Function . . . . .	134
7.3.1	Diversity-enforcing fitness function . . . . .	134
7.3.2	Multi-objective optimization . . . . .	136
7.3.3	Proposed multi-objective fitness function . . . . .	137
7.4	Proposed Approach . . . . .	139
7.4.1	Proposed algorithm . . . . .	140
7.5	Experimental Evaluation and Analysis . . . . .	140
7.5.1	Data streams . . . . .	141
7.5.2	Baseline methods for comparison . . . . .	142
7.5.3	Implementation details . . . . .	142
7.5.4	Results and discussion . . . . .	142
7.6	Classifier Updation . . . . .	147
7.6.1	Proposed approach . . . . .	148
7.6.2	Results and discussion . . . . .	151
7.7	Summary . . . . .	152
<b>8</b>	<b>Conclusions and Future Scope</b>	<b>155</b>
8.1	Conclusions . . . . .	155
8.2	Future Scope of Work . . . . .	156
	<b>Works Cited</b>	<b>159</b>

# List of Figures

2.1	General process of data stream classification . . . . .	12
2.2	Partitioning of data . . . . .	21
2.3	Taxonomy of privacy-preserving data classification techniques . . . . .	22
2.4	Sub-tree mutation . . . . .	33
2.5	Sub-tree crossover . . . . .	33
2.6	General process of Genetic Programming . . . . .	35
2.7	A general ensemble architecture . . . . .	36
3.1	Training accuracy of classifiers (in %) . . . . .	43
3.2	Prediction accuracy of classifiers (in %) . . . . .	44
3.3	Training time of classifiers (in s) . . . . .	44
3.4	Prediction time of classifiers (in s) . . . . .	45
3.5	Training accuracy of classifiers (in %) . . . . .	50
3.6	Prediction accuracy of classifiers (in %) . . . . .	50
3.7	Training time of classifiers (in s) . . . . .	51
4.1	Generic view of proposed approach . . . . .	58
4.2	Creating and connecting to the server created at the merger site . . . . .	61
4.3	Creating and authenticating users (the local sites) . . . . .	61
4.4	Authorizing the local sites for read/write permissions . . . . .	62
4.5	Login to the global site server by a local site . . . . .	62
4.6	Adding/Reading classifier) at the merger site . . . . .	63
4.7	Access Log maintenance at the server . . . . .	63
5.1	DAHOT traversal example . . . . .	77
5.2	Scenario of interleaved training and test instances in data stream . . . . .	83
5.3	Predictive accuracy (in %) using different discretization methods . . . . .	84
5.4	Predictive accuracy vs Privacy level ( $k$ ) on data stream 1 . . . . .	86
5.5	Predictive accuracy vs Privacy level ( $k$ ) on data stream 2 . . . . .	86
5.6	Predictive accuracy vs Privacy level ( $k$ ) on data stream 3 . . . . .	87
5.7	Predictive accuracy vs Privacy level ( $k$ ) on data stream 4 . . . . .	87
5.8	Predictive accuracy vs Diversity level ( $k$ ) on data stream 1 . . . . .	88
5.9	Predictive accuracy vs Diversity level ( $k$ ) on data stream 2 . . . . .	88
5.10	Predictive accuracy vs Diversity level ( $k$ ) on data stream 3 . . . . .	89

5.11	Predictive accuracy vs Diversity level ( $k$ ) on data stream 4 . . . . .	89
5.12	Training accuracy of classifiers (in %) using periodic hold-out evaluation . . . . .	90
5.13	Predictive accuracy of classifiers (in %) using periodic hold-out evaluation . . . . .	92
5.14	Training time of classifiers (in s) using periodic hold-out evaluation . . . . .	92
5.15	Prediction time of classifiers (in s) using periodic hold-out evaluation . . . . .	93
5.16	Number of nodes in classifiers using periodic hold-out evaluation . . . . .	94
5.17	Information loss of classifiers (in %) using periodic hold-out evaluation . . . . .	94
5.18	Training accuracy of classifiers (in %) using traditional hold-out evaluation . . . . .	96
5.19	Predictive accuracy of classifiers (in %) using traditional hold-out evaluation . . . . .	97
5.20	Training time of classifiers (in s) using traditional hold-out evaluation . . . . .	97
5.21	Prediction time of classifiers (in s) using traditional hold-out evaluation . . . . .	98
5.22	Number of nodes in classifiers using traditional hold-out evaluation . . . . .	98
5.23	Information loss of classifiers (in %) using traditional hold-out evaluation . . . . .	99
6.1	Framework of the proposed approach . . . . .	112
6.2	Predictive accuracy (in %) of classifiers on data streams at merger site . . . . .	119
6.3	Predictive accuracy (in %) of classifiers on data streams at site 1 . . . . .	119
6.4	Predictive accuracy (in %) of classifiers on data streams at site 2 . . . . .	120
6.5	Predictive accuracy (in %) of classifiers on data streams at site 3 . . . . .	120
6.6	Number of nodes in classifiers on data streams at merger site . . . . .	121
6.7	Information loss (in %) of classifiers on data streams at merger Site . . . . .	121
6.8	Training time (in s) of classifiers on data streams at merger site . . . . .	122
7.1	Bull's eye diagram for bias-variance quandary . . . . .	128
7.2	Genetic programming and ensemble learning for global privacy-preserving classifier induction . . . . .	139
7.3	Predictive accuracy (in %) of classifiers on data streams at merger site . . . . .	143
7.4	Predictive accuracy (in %) of classifiers on data streams at site 1 . . . . .	144
7.5	Predictive accuracy (in %) of classifiers on data streams at site 2 . . . . .	144
7.6	Predictive accuracy (in %) of classifiers on data streams at site 3 . . . . .	145
7.7	Number of nodes in global classifiers on data streams at merger site . . . . .	146
7.8	Information loss (in %) of classifiers on data streams at merger site . . . . .	146
7.9	Training time (in s) of global classifiers on data streams at merger site . . . . .	147
7.10	Predictive accuracy (in %) of classifiers before and after classifier update . . . . .	151

# List of Tables

2.1	Merits of data stream classification techniques . . . . .	20
2.2	Original loan approval dataset . . . . .	26
2.3	Publicly available data . . . . .	27
2.4	2-anonymous version of Table 2.2 . . . . .	27
2.5	2-diverse version of Table 2.2 . . . . .	28
3.1	Composition of data streams . . . . .	42
3.2	Composition of data sets . . . . .	48
6.1	Impact of $\lambda$ on fitness measure . . . . .	109
6.2	Composition of data streams . . . . .	114
6.3	GPeCT algorithm parameters . . . . .	118
7.1	Hardness of training instances based on classifiers . . . . .	135
7.2	Classifier fitness based on diversity-enforcing fitness measure . . . . .	136
7.3	Classifier fitness based on the proposed fitness measure . . . . .	139

# Abbreviations

ANNCAD	Adaptive Nearest Neighbor Classification for Data-Streams
AOG	Algorithm Output Granularity
BNN	Back-propagation Neural Network
BSF	Best So Far
CART	Classification And Regression Tree
CASTLE	Continuously Anonymizing Streaming Data via Adaptive Clustering
CM	Classification Metric
CVFDT	Concept-Adapting Very Fast Decision Tree
DA	Discriminant Analysis
DAHOT	Diverse And $k$ -Anonymized Hoeffding Tree
DDSM	Distributed Data Streams Mining
DEA-DA	Data Envelopment Analysis-Discriminant Analysis
DGH	Domain Generalization Hierarchy
EA	Evolutionary Algorithms
ECs	Equivalence Classes
ESDT	Ensemble of Sanitized Decision Tree Classifiers
FTP	File Transfer Protocol
GA	Genetic Algorithms
GP	Genetic Programming
GPeCT	Genetic Programming based evolution of Classification Trees
HAT-ADWIN	Hoeffding Adaptive Tree Using Adaptive Windowing



HT .....	Hoeffding Tree
kNN .....	k-Nearest Neighbor
LR .....	Logistic Regression
MOA .....	Massive Online Analysis
NBC .....	Naive Bayesian Classification
NCL .....	Negative Correlation Learning
NN .....	Neural Networks
NQA .....	Non-Quasi Attribute
OET .....	Orthogonal Evolution of Teams
PCDS .....	Privacy-Preserving Classification of Data Streams
PFC .....	Pairwise Failure Crediting
PID .....	Personal Identifier
PPDM .....	Privacy-Preserving Data Mining
PPDSC .....	Privacy-Preserving Data Stream Classification
QID .....	Quasi-Identifier
RF .....	Random Forest
SA .....	Sensitive Attribute
SCALLOP .....	Scalable Classification Algorithm by Learning Decision Patterns
SDTP .....	Sanitized Decision Tree Classifier on Partial Data Stream
SKY .....	Stream $k$ -Anonymity
SMC .....	Secure Multiparty Computation
SVM .....	Support Vector Machine
VFDT .....	Very Fast Decision Tree
WEKA .....	Waikato Environment for Knowledge Analysis

# Chapter 1

## Introduction

### 1.1 Motivation

In recent years, advancement in networking technologies and explosion in the availability of data from various sources has triggered abundant opportunities for collaboration, particularly collaboration in data mining (Zhan; Aggarwal and Yu). Several organizations related to financial services, healthcare, e-commerce, market analysis, national security, etc. are willing to undertake a joint data mining task to obtain certain global patterns because of the mutual benefits it brings. However, due to legal constraints or competition concerns, these organizations are reluctant to disclose their private data to each other or to any third party.

Two such realistic scenarios are as follows (Zhan; Zhuojia and Xun): 1) Several supermarkets want to perform data mining on the joint data set representing buying behavior of their customers. As these businesses are competitors, neither of them is willing to reveal information of its customers to each other, whilst they are aware of the benefit brought by this collaboration. 2) Multiple pharmaceutical companies want to collaboratively conduct data mining to discover meaningful patterns among human genes but are disinclined to share their raw data with other parties participating in the collaboration.

## CHAPTER 1. INTRODUCTION

Conventionally, the identifying attributes like names, addresses and the like are trimmed out from the original dataset to assure that individuals' private information is not inferred. Some privacy regulations such as HIPAA laws (HIPAA) require that medical data should be released only after sufficient anonymization (Aggarwal and Yu). But, simply omitting the identifying attributes is not sufficient as the data also contains attributes depicting other personal information like date of birth, gender, zip code, etc. Such attributes, called quasi-identifiers (Aggarwal and Yu) can be linked with information from other publically available sources and the privacy of individual records can be breached.

For example, several hospitals may wish to collaboratively participate in a research that studies characteristics of various diseases by releasing data about their patients' diagnosis. Although the raw data, also called microdata is de-identified before releasing or applying data mining algorithm, the sensitive details and medical history of an 82 years old female patient living in a sparsely populated region may be identified by combining this individual's gender, age and zip code with an external dataset of voting registration.

As such threats against privacy are increasing; it is required to reconsider data mining algorithms from the privacy-preservation point of view and the field of privacy-preserving data mining (Agarwal and Srikant; Lindell and Pinkas; Kantarcioglu; Zhang, Wang, and Zhao) has emerged in response to this issue. Formally, privacy-preserving data mining (PPDM) techniques aim to perform a trade-off between efficiency in data mining and exposure of sensitive information of the subjects. Since multiple parties, each with its own set of records is involved in the data mining process, the problem is commonly referred to as privacy-preserving distributed data mining.

Recent developments in the field of privacy-preserving data mining focus on two major subjects: preserving the privacy of homogeneously distributed (or horizontally partitioned) data and preserving the privacy of heterogeneously distributed (or vertically partitioned) data.

Most of the previous work in privacy-preserving data mining considers only static data (Aggarwal and Yu; Hwanjo, Xiaoqian, and Vaidya; Agarwal and Srikant; Lindell and Pinkas; Kantarcioglu, Jin, and Clifton; Kantarcioglu; Friedman, Wolff, and Schuster;

## CHAPTER 1. INTRODUCTION

Tian et al.). Meanwhile, in today’s digital era, massive data streams are being generated through various sources. Extracting patterns and models from such voluminous data streams, known as data stream mining is a challenging task.

Several real-world applications of privacy-preserving data mining and data stream mining are associated with classification task . Thus, this work focuses on privacy-preserving data stream classification and in particular, privacy-preserving distributed data classification where data is distributed among collaborating parties (Aggarwal and Yu; Hwanjo, Xiaoqian, and Vaidya; Lindell and Pinkas). The focus is on homogeneously distributed data as numerous applications fall under this data model.

For example, several banks collect transactional information for credit card customers where the features collected, such as age, gender, balance, average monthly deposit, etc. are the same for all banks (Hwanjo, Xiaoqian, and Vaidya). Identifying whether a particular transaction is fraudulent or not is a problem called “privacy-preserving classification of homogeneously distributed data” where the privacy of customers’ data needs to be protected. Further, due to the growing threat of identity theft, credit card loss, etc., credit card transaction data are analyzed as data stream and the credit card fraud detection can be posed as a privacy-preserving horizontally partitioned data stream classification problem. Many such problems that abound in various diverse domains demand an efficient approach for privacy-preserving classification of horizontally partitioned data streams.

As another example, in medical application, continuous streams of data from hospitals or pharmacy stores can be used to detect any abnormal disease outbreaks or biological attacks (Aggarwal). Several insurance companies and hospitals analyze data on disease incidents, long-term effects of the disease, patient background, seriousness of the disease, etc. Organizations like the Center for Disease Control aim to identify disease outbreaks by training a classifier across the data streams (homogeneous) arriving at various hospitals (Aggarwal and Yu). Such organizations are seeking patterns that are indicative of disease outbreaks; which they use to classify a query instance as an outbreak or the opposite. Moreover, the new viruses or diseases may emerge on the go, while the new symptoms and/or new medical cases must be taken into consideration and added to the current learn-

ing model accordingly. Detecting disease outbreaks before-time prevents life-threatening infectious diseases like bird flu, swine flu, dengue, as well as threats of bioterrorism. This has made disease surveillance a national priority. Hence it is important and beneficial to have a privacy-preserving horizontally partitioned data stream classification technique that is capable of classifying potential outbreaks in medical data streams while respecting the private details of the patients.

Such and similar applications inspire to propose solutions for the task of privacy-preserving classification of horizontally partitioned data streams. But, data stream classification and privacy-preservation are divergent tasks because privacy-preserving methods may require multiple scans over the data which is not suitable for the voluminous data streams. Hence, the crucial issue of privacy-preserving horizontally partitioned data stream classification serves as a motivation for this work.

## 1.2 Objectives

Privacy-preserving classification of horizontally partitioned data streams is a requisite to several real-world applications like disease outbreak detection, credit-risk classification, etc. But data stream classification and privacy-preservation are two conflicting goals which get even complex when data is distributed among several nodes. The objectives of the work documented in this thesis are:

- To develop an efficient method for preserving output-privacy in data stream classification. That is, to trade-off accuracy and efficiency in data stream classification while preventing privacy-breach through record linkage and attribute linkage attacks.
- To develop a systematic method for privacy-preserving classification of horizontally partitioned data streams. That is, to optimize output-privacy-preserving data stream classification while producing a global model from horizontally partitioned data streams.

### 1.3 Scope of the Work

The work aims to develop an efficient method for privacy-preserving classification of distributed data streams. The scope of the work covers the following:

The work focuses on  $P$  data streams (denoted by  $D_1, D_2, \dots, D_P$  respectively) arriving at  $P$  different nodes (i.e.  $P$  parties). The data owners wish to collaboratively construct a classifier on the union of their data streams. The work assumes that there are 3 parties ( $P = 3$ ) who want to collaborate.

The data stream comprises of a sequence of records:  $\{x_1, x_2, \dots, x_n\}$ , where  $x_1$  is the first record and  $x_n$  is the record that recently arrived. Each record is a multi-dimensional feature vector with  $d$  attributes:  $\{A_1, A_2, \dots, A_d\}$ . Some of the records in the data stream have an associated class label  $C_i \in \{C_1, C_2, \dots, C_m\}$  forming a set of training instances. The class label associated with the remaining instances is to be predicted using the classifier induced from the training instances. The class label is considered as a sensitive attribute whereas the other attributes are considered as quasi-identifiers; a scenario prevalent in several applications. The work considers that all these distributed streams have the same schema. That is, the data streams are horizontally partitioned.

The database owner wishes to ensure that the sensitive information in the original data should not be revealed through either direct/indirect (via linking and inference) exposure and more importantly, through the data mining output. That is, the focus is on preserving output-privacy (preserving inference using the released classifier) to prevent record-linkage and attribute-linkage attacks.

There exists a central third-party that combines the results of the participating parties. No private information to be disclosed to the merger site too. Further, the channel between the participating sites and the merger site is assumed to be secured.

The work focuses on the problem of credit-risk classification in banking sector, an important application of privacy-preserving classification of horizontally partitioned data streams. Hence, all the experiments are conducted on data streams relevant to credit-risk classification with each data stream having only two classes.

The proposed approach is compared (for performance evaluation) with few relevant methods which are described in respective chapters of the work. Since no direct methods for privacy-preserving classification of horizontally partitioned data streams exist in literature, the performance of the proposed approaches is compared with approaches that combine data stream classification and privacy-preservation techniques.

## 1.4 Significant Contributions

The work in the thesis addresses the issue of privacy-preserving data stream classification and proposes an algorithm named DAHOT for the same. DAHOT takes as an input the data stream and induces an output-privacy-preserving decision tree classifier that provides high classification accuracy with small information loss, under the given anonymity and diversity requirement. An empirical evaluation of DAHOT indicates its efficacy in classifying massive data streams while preserving the required privacy.

Due an increasing demand of collaboration among competing business organizations for obtaining global patterns from data, the issue of privacy-preserving classification of horizontally partitioned data streams needs high attention. As an another contribution, the work in this thesis proposes a Genetic Programming based approach (named DAHOT-GPeCT) that evolves a global classifier from the privacy-preserving data stream classifiers (DAHOTs) induced by parties participating in the collaboration.

The work addresses the problem of decision-making in banking sector. The novelties introduced in application of Genetic Programming, which include initializing population using decision tree classifiers, variable size population, reverse rank selection strategy for mutation, etc. contribute extensively in such and similar decision-making.

Further, an approach that uses ensemble-based learning along with Genetic Programming is proposed to contribute an effective way of privacy-preserving classification of horizontally partitioned data streams. This approach, named DAHOT-GPeCT-Ensemble, uses a novel fitness function that results into an improved performance of the global classifier.

## 1.5 Thesis Organization

Rest of the thesis is organized as follows:

Chapter 2 (Background): This chapter presents the fundamentals of the concepts used in this work. The work in literature related to the four major components of the work are discussed. But, the literature specific to the usage of these components to accomplish the objectives of the work, is discussed, whenever the corresponding chapter is elaborated.

Chapter 3 (Empirical Evaluation of Preliminary Components): This chapter presents an empirical evaluation of existing methods for two major components that form the heart of the work: data stream classification and privacy-preserving data classification. Through experiments, this chapter identifies suitable techniques for each component.

Chapter 4 (Proposed Framework): This chapter presents the proposed framework to accomplish the goal of privacy-preserving classification of horizontally partitioned data streams. It describes the application targeted in this work, presents the relevant literature and covers the generic view of proposed approach as well as details of the implementation environment.

Chapter 5 (Preserving Output-Privacy in Data Stream Classification): This chapter proposes an algorithm to induce output-privacy-preserving classifier from the data streams arriving at individual sites. The proposed algorithm is implemented and compared with similar existing methods for performance evaluation.

Chapter 6 (Genetic Programming-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams): This chapter proposes a genetic programming based approach to induce a privacy-preserving classifier from horizontally partitioned data streams. This chapter also discusses the experimental results and compares it with results obtained using similar existing techniques.

Chapter 7 (Ensemble-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams): A technique that uses genetic programming, as well as ensemble-based learning is proposed in this chapter. The chapter also proposes a novel fitness function. Further, through empirical analysis, this chapter proves the effectiveness of



## CHAPTER 1. INTRODUCTION

proposed solution in addressing the work's objectives.

Chapter 8 (Conclusion and Future Work): In this chapter, the conclusions derived from the literature survey and analysis of experimental results conducted throughout the work are presented. Future scope of work in this field is also stated in this chapter.

*Works Cited* section consists of related research work cited in the thesis.

# Chapter 2

## Background

This chapter discusses four major components that compose the proposed work: 1) Data stream classification, 2) Privacy-preserving data classification, 3) Genetic Programming and 4) Ensemble learning. A detailed study on each of these components is presented here but the literature specific to usage of these components to address issues arising in the proposed approach is presented in the subsequent chapters.

### 2.1 Data Stream Classification

The advancements in hardware and software technologies have facilitated several applications to generate a vast amount of data. Such data, also known as data streams, are characterized as continuously arriving at unprecedented rates (Aggarawal). Examples of data streams include internet traffic streams, stock trading streams, streams generated by e-commerce sites, data generated from scientific projects, supermarket data, multimedia data, medical data streams, data streams generated through industry production processes, remote sensor and video surveillance streams, etc.

Since last few decades, the data mining techniques have been successfully applied to several real-world problems. As these data streams contain valuable knowledge, there is an enormous demand for analyzing and mining them. But, the traditional data mining methods assume that the data can be scanned multiple times in order to mine it. Due to

its large volume, performing several scans on it is not cost-effective. Extracting patterns and models from such continuous data streams, called data stream mining, is a challenging task.

One data mining task that has been an active area of research from the perspective of data streams is that of classification. Applications of data stream classification include e-mail spam detection, credit card fraud detection, malicious web page detection, intrusion detection, detection of any abnormal disease outbreaks from continuous streams of data arriving at hospitals, etc. Predicting class label of the incoming data streams, that is, data stream classification is an emerging issue because the classification algorithm needs to be executed endlessly.

Several issues arise in addressing efficient classification of data streams. Some of the primary research issues are discussed here (Aggarawal; Golab and Ozsu; Abdulsalam, Skillicorn, and Martin; Bifet et al.; and Wang et al.).

- Handling the continuous flow of data streams: The intrinsic feature of data streams is its speedy and continuous flow. The traditional data mining methods require all the data collected before applying mining tasks. A data stream classification method should be able to adapt with the data arriving continuously at varied rates.
- Interleaved labeled and unlabeled instances: Since the conventional classification techniques assume all the data collected at hand before mining, it divides the entire classification process into three phases: a training phase that uses labeled data to train a classifier model; a test phase that uses previously unseen data to test the model; and a deployment phase wherein the model is applied to classify the unlabeled data (Abdulsalam, Skillicorn, and Martin). On the other hand, data stream classification consists of only a single stream of data, in which labeled and unlabeled data are mixed collectively within the stream. Hence, the training, testing and deployment phases may require to be interleaved and the classifier should be ready to predict at any point.

- **Concept-drift:** Most existing algorithms assume that data streams come under stationary distribution, where the concept of data remains unchanged. When the underlying class (concept) of the data changes over time, it is referred as concept drift, which is quite frequent in real-world applications. In case of concept drift, there is a quandary: whether to update the classifier often and waste resources on changes that might be momentary (or insignificant) or else to update the model occasionally (which may result into degradation of the accuracy of the classifier).

There are three algorithmic approaches in order to tackle this quandary: 1) periodic approach; where the classifier is rebuilt periodically, 2) incremental approach; where the classifier is updated with every concept drift and 3) reactive approach, where changes are monitored and the classifier is rebuilt only if it no longer matches the underlying data. Each of this algorithmic approach has few benefits and limitations. The periodic approach is simple and has fixed communication and computation cost but wastes resources when there is no concept-drift. The incremental approach is accurate and efficient, but immediately updating the classifier might be a waste of resources if the drift is momentary. The reactive approach is suitable if monitoring of the classifier's match with the incoming data stream is done accurately and efficiently. Updating the classifier seldom will save resources, and rebuilding the classifier when it does not suit the data any longer will maintain the accuracy.

- **Memory Concerns:** The large volume of data streams and the requirement of classification algorithms to scan the data multiple times are two conflicting issues. The solution of this issue can either be availability of unbounded memory or application of data stream pre-processing techniques like load shedding, sampling, aggregation, data synopsis, etc. Otherwise novel classification techniques need to be designed to address this need of data streams.
- **Tradeoff between Accuracy and Efficiency:** Another issue in data stream classification is to tradeoff the accuracy of the classifier and the time complexity. Techniques that guarantee accurate output in a small time need to be found.

- Distributed data stream classification: A considerable number of applications have data streams distributed among multiple parties and patterns from the union of these data streams need to be extracted in order to obtain global trends in the market. Combining these voluminous data streams is infeasible and demands algorithms that efficiently classify these distributed data streams.

Few other than the above mentioned issues also prevail in data stream classification, however as these are the most significant ones, the proposed work tries to address each of them to a larger extent. Data stream classification is generally carried out by using novel classification techniques developed specifically for data streams or by pre-processing the data streams and then applying traditional classification techniques.

The general process of data stream classification is shown in Figure 2.1. The components within the block are elaborated in sub-sections 2.1.1 and 2.1.2.

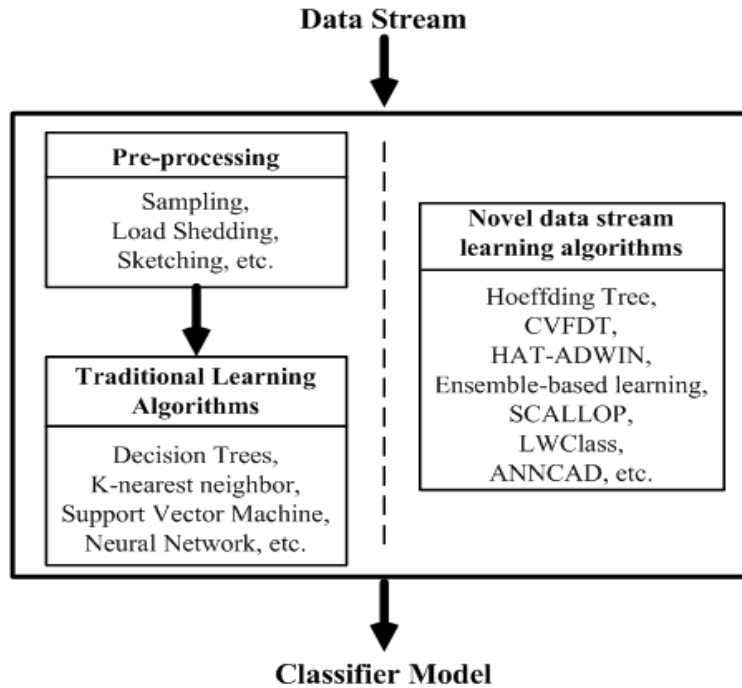


Figure 2.1: General process of data stream classification

### 2.1.1 Data stream pre-processing

In order to apply the conventional data mining techniques on data streams and address the aforementioned issues, these data streams need to be pre-processed and made suitable using the entrenched statistical and computational approaches. Some popular and successful data stream pre-processing techniques are as follows (Aggarawal; Golab and Ozsü):

- **Sampling:** It is a well-known technique that has been employed in several tasks since years. Sampling consists of choosing a subset of data stream and performing the analysis and mining process only on the subset instead of the entire data stream. Efficient sampling techniques guarantee the error bounds but may fail to detect anomalies in sensitive applications like surveillance system.
- **Load Shedding:** Load shedding involves ignoring a chunk or a sequence of data from the stream. This results into a decreased volume of data and hence addresses the issue of multiple scans and memory concerns. It has proved to be efficient in querying data streams but suffers from the same problems like sampling as the dropped chunk of data may represent a pattern that is interesting to study.
- **Sketching:** Sketching refers to the method of randomly projecting a subset of the features (attributes) and can be considered as vertical sampling of the data stream. One of the commonly used sketching techniques is Principal Component Analysis. A disadvantage of sketching is that it may discard some important and relevant features.
- **Synopsis Data Structure:** Synopsis data structures are created by applying transformation techniques like summarization that efficiently summarize the continuously arriving data streams. The synopsis can then be used for further analysis. Histograms, wavelet analysis, quantiles, etc. are used as synopsis data structures. For data streams arriving at very high speed, synopsis may be insufficient. Further, as

the synopsis discards some of the characteristics of the data, approximate answers are obtained on application of such data structures.

- **Aggregation:** Aggregation is a pre-processing technique similar to summarization but is independent of any mining task. It computes statistical measures like mean and variance to summarize the data stream and the mining algorithm is applied on this aggregated data. Aggregation techniques become ineffective when there are high fluctuations in the data distribution.

Despite of certain limitations, many of these techniques have been widely applied to pre-process the data streams and make them suitable for traditional data mining algorithms. The work in this thesis uses sampling-based technique as a baseline method for comparison. Other than these pre-processing techniques, some classification techniques specific to data streams have been proposed. The next sub-section describes the literature on such techniques.

### 2.1.2 Data stream classification techniques

This section describes the state-of-the-art data stream classification techniques and presents how well they address the demands of data streams.

- **Hoeffding tree classifier:** Domingos and Hulten proposed the concept of Hoeffding tree, a decision tree classifier that is capable of learning from data streams. They name its implementation as Very Fast Decision Tree (VFDT), but the generic term Hoeffding tree is used throughout the paper. The Hoeffding tree classifier learns from the data streams incrementally by examining each record only once while producing trees of quality similar to batch learned trees.

While choosing the best splitting attribute at a given node, it may be sufficient to use only a small sample of the available training instances that pass through that node. Hence, the instances that arrive first on the data stream are used to choose the root attribute and subsequent instances are passed down the tree until they

reach the corresponding leaves. These instances at the leaves are used to select the splitting attribute there and the process is repeated recursively. The number of examples required at each node is decided using a statistical result called Hoeffding bound (Hoeffding). Consider  $\bar{r}$  is the observed mean of some real-valued random variable  $r$  with range  $R$  after  $n$  independent observations. The Hoeffding bound of equation 2.1 ensures with a probability  $1 - \delta$  that true mean of  $r$  is at least  $\bar{r} - \varepsilon$  where

$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \quad (2.1)$$

Let  $G$  be an attribute selection measure that is to be maximized. For information gain measure, the range  $R$  of  $G$  is  $\log_2(\text{\#classes})$ . After observing  $n$  examples at any given node, let  $a_1$  be the attribute with the highest  $\bar{G}$  value,  $a_2$  be the attribute with the second-highest  $\bar{G}$  value and let the difference between the two attributes be  $\Delta\bar{G} = \bar{G}(a_1) - \bar{G}(a_2)$ . If  $\Delta\bar{G}$  is greater than  $\varepsilon$ , then the Hoeffding bound guarantees with probability  $1 - \delta$  that the true  $\Delta G$  is greater or equal to 0 and  $a_1$  is the best splitting attribute.

A number of refinements to Hoeffding tree have been proposed in the literature such as tie-breaking when the value of  $G$  is almost similar for two or more attributes, eliminating consideration of attributes that are not promising, scanning the examples one more time if the need is, etc. (Domingos and Hulten; Kirkby). Since Hoeffding tree does not address the issue of concept-drift itself, researchers have proposed its variants to cope up with concept-drifting data stream.

- **Concept-Adapting Very Fast Decision Tree:** The method Concept-Adapting Very Fast Decision Tree (CVFDT) (Hulten, Spencer, and Domingos) works by maintaining a model consistent with respect to a sliding window of instances from the data stream. It creates an alternate subtree when it detects that the distribution of data is changing at a node, but this subtree replaces the old tree only after it becomes accurate. The disadvantage of CVFDT is that it does not define any optimal window size. A small window reflects the current distribution accurately whereas a large



size accommodates several examples to work on and increases accuracy when the concept is stable.

- Hoeffding Adaptive Tree using Adaptive Windowing: The shortcoming of CVFDT stated in the previous point is overcome by a technique named Hoeffding Adaptive Tree using Adaptive Windowing (HAT-ADWIN) (Bifet and Gavalda). ADWIN is a method that keeps a variable-length window of recently seen items which automatically grows when no change is apparent and shrinks it when the data changes. In HAT-ADWIN, all statistics of instances are stored in relevant nodes. When there is a concept-drift, HAT-ADWIN grows a classification tree identical to what VFDT would grow from a new stable distribution but the new tree replaces the old tree only when the former becomes accurate than the latter.
- Ensemble-based classifier: A general framework for classification of data streams with concept-drift is proposed by Wang et al. Instead of constantly modifying a single classifier model, the framework proposes to train a weighted classifier ensemble from sequential data chunks partitioned from the stream. A classifier is learned for each chunk and the weight of classifier is inversely proportional to its expected prediction error. Wang et al. shows that classifier ensembles outperform single classifiers while classifying concept-drifting data streams.
- Novel-class detection enabled technique: Most of the traditional data stream classification techniques assume a fixed number of classes. But in real-world, novel classes may emerge at any time in the data stream, which remain undetected by these traditional data stream classification methods until a classifier is trained with the novel classes. A data stream classification method proposed by Masud et al. automatically detects a novel class. It is assumed that the instances that belong to the same class should appear closer to each other (called cohesion) and should be distant from the instances that belong to other classes (called separation). Any test instance that is found to be separated from the training data may probably be an instance of a novel class. However, a novel class is assumed to be detected only if an

adequate number of such instances that exhibit strong cohesion appear in a stream.

- **Imbalanced data stream classifier:** Several real-world applications such as intrusion detection, disease outbreak detection, credit card transaction fraud detection, etc. generate imbalanced data streams where the number of data instances from one class is quite less as compared to the other class. Correctly classifying such minority class examples is a major issue. Like Masud et al., Godase and Attar also propose a method in which the incoming data stream is divided into chunks, a classifier is trained from each chunk and an ensemble of these classifiers is created. Here, the imbalanced data streams are classified by accumulating minority class instances from previous data chunks and adding them into the current training chunk. Using previous minority class instances give a benefit over the traditional method of synthetically creating such examples.
- **Naive Bayes:** Naive Bayes classifier can be adopted for Data Streams in its original form (Bifet et al.; Kirkby) and requires only maintaining a statistics table. For discrete valued attributes, only the class label counts per attribute value are required. Continuous numeric attributes are to be discretized a priori. An attribute with  $m$  unique attribute values and  $n$  possible classes can be stored in a table with  $mn$  entries only. On arrival of new training instances in the stream, tables are updated by merely incrementing the appropriate entries as per attribute values and class. Naive Bayes is a simple and successful data stream classification technique that is widely used in absence of concept-drift.
- **Scalable Classification Algorithm by Learning decision Patterns:** This rule-based classifier for data streams has been proposed by Ferrer-Troyano, Aguilar-Ruiz, and Riquelme. Despite the complicatedness in maintaining statistics for a rule-based classifier, this approach named Scalable Classification ALgorithm by Learning deciSiOn Patterns (SCALLOP) scales efficiently on data streams. The approach reads a pre-defined number of labeled instances and creates rule for each class. Thereafter, the process of rule set maintenance is executed for every newly arriving instance.

Within this process, if a new instance fortifies an existing rule, the support and confidence of that rule is re-computed. If a new instance is associated with at least one of the existing rule but isn't covered by the rule, then the associated rule is expanded within the growth bounds if it does not intersect with any of the existing rule that is associated with the instance's class label. Lastly, if a new instance weakens an existing rule, it's negative support and confidence is re-computed and if the confidence falls below a threshold, a new rule built from that instance is added to the rule set. Periodically, a rule-refinement process is performed wherein rules that belong to the same class and have small distance are merged. Also, rules whose support is less than a minimum bound or aren't covered by any instance are discarded. While classifying a new instance, if any rule covers it, then the instance is assigned the class associated with that rule. Otherwise, the class label is inferred using a voting-based mechanism among the existing rules.

- **Adaptive Nearest Neighbor Classification for Data-streams:** An incremental data stream classifier inspired by the popular k-nearest neighbor classification technique is proposed by Law and Zaniolo. This classifier named Adaptive Nearest Neighbor Classification for Data-streams (ANNCAD) uses a grid-based representation. The classification process of any new instance begins by taking a majority of votes by nearest neighbors located at fine level. In situation when the instances located at finer level cannot distinguish between the classes, the votes of instances located at coarser levels in the hierarchy are taken. ANNCAD addresses the issue of concept-drift by exponentially decreasing the weight of old instances. The flaw of this method is that the exponential fading factor may over-estimate or under-estimate the concept-drift resulting into a decrease in prediction accuracy.
- **Lightweight Classification:** An Algorithm Output Granularity (AOG) based technique known as Lightweight Classification (LWClass) is proposed by Gaber, Krishnaswamy, and Zaslavsky. Based on the data rate and available memory, instances are stored in the memory. When a new labeled instance  $x$  arrives, the algorithm

measures the distance between  $x$  and nearest instances from the memory. If the class label of the nearest neighbor is same as  $x$  and the distance is less than a pre-determined threshold, the average of the two instances is stored and the weight of this average entry is increased by 1. The weight is decreased by 1 if the class labels are different and the entry is deleted when the weight reaches 0. Any unlabeled instances are classified using the majority vote of k-nearest neighbor entries.

- **On-demand classification** A micro-cluster based technique called on-demand classification is proposed by Aggarwal et al. It is named on-demand classification method because instances are classified on demand when they arrive in the data stream. The method stores summarized statistics about the data stream in form of class-specific micro-clusters. The method works in two modules where the first module keeps storing the summarized statistics about the data streams whereas the second uses these statistics to classify new instances. Since the summary statistics are updated whenever new data arrives, the method has great flexibility and is suitable for several applications.

Table 2.1 summarizes the merits and usefulness of each of the data stream classification techniques described above. The summary aids in identifying suitable techniques for any application at hand.

## 2.2 Privacy-Preserving Data Classification

Due to recent advances in data storage and dissemination, privacy-preservation has emerged to be a major concern in devising a data mining system. Privacy-preserving data mining (Agarwal and Srikant; Lindell and Pinkas; Aggarwal and Yu) is a novel area of research which ensures that the sensitive information in the data being mined should be protected from either direct or indirect (via inference) exposure. In recent years, privacy-preserving data classification has received tremendous attention. The goal here is to build accurate classifiers without unveiling the privacy of the data being mined (Xu et al.).

Table 2.1: Merits of data stream classification techniques

<b>Technique</b>	<b>Merits</b>
Hoeffding Tree / VFDT	Classifies high-speed data streams without concept-drift with high efficacy
CVFDT	Classifies data streams with concept-drift efficiently using fixed-size window
HAT-ADWIN	Classifies data streams with concept-drift efficiently using variable length window that is adaptable to change in distribution of data stream
Ensemble-based classifier	Classifies concept-drifting data streams efficiently as compared to single classifier
Novel-class detection enabled technique	Detects novel classes that emerge in data stream but are not defined a priori
Imbalanced data stream classifier	Classifies minority class instances in data stream efficiently
Naive Bayes	Simple and useful method for classifying data streams without concept-drift
SCALLOP	Scalable rule-based classification technique for large data stream
ANNCAD	Incremental classifier based on popular k-nearest neighbor method made adaptable for data streams
LWCClass	Lightweight classifier that functions as per current data rate and available memory
On-demand classification	Flexible and efficient technique based on storing summarized statistics about data streams

## CHAPTER 2. BACKGROUND

One of the major applications that demand reconsidering data classification algorithms from the viewpoint of privacy-preservation is collaboration. Collaboration may be required between several participating parties like banks, competing supermarkets, hospitals, etc. (Zhan). The data is thus considered to be distributed among these participating parties, either homogeneously or heterogeneously (Verykios et al.; Clifton et al.). Figure 2.2 shows the two ways of data partitioning .

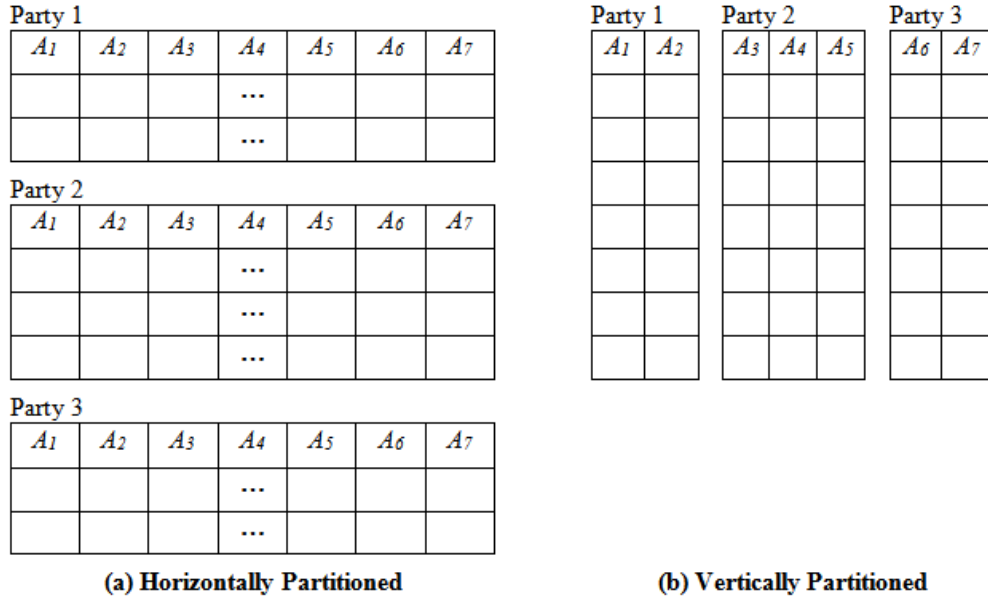


Figure 2.2: Partitioning of data

Vertically partitioned data (or heterogeneously distributed data) refers to data being collected by different sites or parties on the same individuals but with different attributes (feature sets). Horizontally partitioned data (or homogeneously distributed data) refers to different sites collecting similar kind of data over different individuals. Besides this, at times data may also be arbitrarily partitioned. That is, the instances as well as the attributes, both may be distributed between the parties.

The work in this thesis focuses on horizontally partitioned data streams (Hwanjo, Xiaoqian, and Vaidya; Samet and Miri; Kantarcioglu; Xiong, Chitti, and Liu).

### 2.2.1 Privacy-preserving data classification techniques

Several methods for privacy-preserving data classification are presented in the literature. Figure 2.3 shows a generic taxonomy of privacy-preserving data classification techniques followed by a brief description of these techniques and their usefulness.

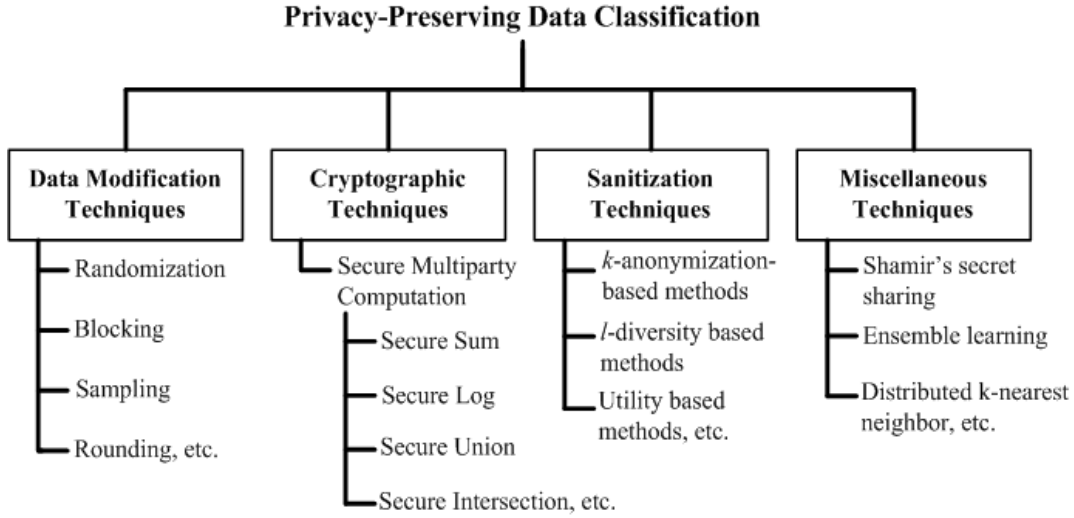


Figure 2.3: Taxonomy of privacy-preserving data classification techniques

Data modification and sanitization techniques need to be adapted in an environment where the data is distributed whereas the cryptographic and other miscellaneous techniques described herein are specifically designed for inducing classifiers from distributed data.

#### • Data modification techniques:

To achieve the goal of privacy-preserving data classification, a major solution category is to modify the data to be classified. Data modification techniques include randomization, blocking, sampling, rounding, etc. (Agarwal and Srikant; Zhang, Wang, and Zhao; Aggarwal and Yu).

Randomization approach has been widely used for privacy-preserving classification. Initially, in randomization approach, the data providers randomize the data and transmit it to the data miner, i.e. a sufficiently large noise is added to the data

with a goal that the true values of the records cannot be recovered. Further, a distribution reconstruction algorithm that reconstructs the original data distribution from the randomized data instances is employed and a classifier is built from the reconstructed data. Several distribution reconstruction algorithms like EM and Bayes reconstruction method have been used in the literature. The approach has the advantage of simplicity, but it lacks a formal framework that proves how much privacy is assured.

Blocking method is generally used in hiding rules. It replaces value of a privacy-sensitive attribute with an unknown value that does not exist in the domain of attributes. Rounding is a method that replaces the values of privacy-sensitive attributes with rounded values. But, rounding is applicable to continuous attributes only. In sampling, data of only a small sample of population is released. Several such data modification techniques have been proposed in the literature and found successful. The work proposed in this thesis uses the concept of sampling technique while releasing data for global classifier induction.

- **Cryptographic techniques:**

A large portion of literature in privacy-preserving data classification discusses the application of cryptographic techniques and Secure Multiparty Computation (SMC) for building classifiers in a privacy-preserving manner (Clifton et al.; Samet and Miri). For example, in the work by Yehuda and Benny, the training set is considered to be homogeneously distributed between two parties and the authors attempt to securely construct an ID3 decision tree. In such techniques, secure log algorithm, secure sum, etc. sub-protocols are used to securely calculate the conditional entropy for an attribute shared by the two parties and a classifier using ID3 is built securely. Classifiers so created resemble the classifier induced if the data is centrally accumulated.

Although such methods are secure enough, they demand a lot of computation and the cost of communication is also high.



- **Sanitization-based methods:**

Frequently for preserving privacy, the data attributes are partitioned into four categories: the personal identifier (PID), quasi-identifier (QID), non-quasi attribute (NQA) and sensitive attribute (SA). PIDs (such as name, social security numbers, etc.) uniquely identify individuals and are removed before mining or publishing the data. SAs (such as disease, credit worthiness, etc.) contain sensitive information and the entire process of privacy-preservation focuses on protecting the SA values from disclosure. QIDs (such as age, marital status, gender, ZIP code, etc.) do not directly reveal the identity of individuals, but if combined with other public datasets or available background knowledge of people referred in the data; these QIDs can link a target individual to a specific SA value. NQAs do not appear in any external tables and thus cannot be exploited for linking to a SA. Hence, unlike QIDs, they do not require anonymization.

Let  $X$  be a target victim and  $r_X$  be the record of  $X$  in the available data set  $D$ . Researchers in field of privacy-preserving data mining have proposed several models that are resistant to different attacks. These privacy models include  $k$ -anonymity (Samarati), (Sweeney),  $l$ -diversity (Machanavajjhala et al.),  $t$ -closeness (Li and Li), etc. and can be classified based on the following types of attacks (Fung et al.):

- Record Linkage: In this attack, an adversary can link  $r_X$  to a small group of records, say  $G$ . Moreover, with some additional knowledge, the adversary can precisely identify  $r_X$  in  $D$ .
- Attribute Linkage: This attack occurs once the record linkage attack has been applied on  $D$  and  $r_X$  has been associated to a small group of records, say  $G$ . In this attack, an adversary may not be able to uniquely identify  $r_X$  but can deduce SA of  $X$  based on the SAs linked to  $G$ . If SA of all the records in  $G$  is same, the SA of  $X$  can be easily found.
- Table Linkage: In record and attribute linkage attack, an adversary has the knowledge about the presence of  $X$  in  $D$ . Occasionally, the presence or absence

of  $X$  in  $D$  automatically reveals SA of  $X$ . A table linkage attack occurs when an adversary can confidently deduce the presence or absence of  $X$  in  $D$ .

$k$ -anonymity principle proposed by Samarati and Sweeney has received considerable attention in recent times and is accepted by both legislators and corporations. It models protecting the data to be mined or released from possible re-identification of individuals to whom the data refers. It takes a safe approach necessitating that each record in the dataset should be indistinguishable from no fewer than other  $(k - 1)$  records with respect to the QIDs that can be used for linking. The idea is to form groups of at least  $k$  records that share the same QID values. These groups are called equivalence classes (ECs). Generalization and Suppression (Wang, Yu, and Chakraborty; Fung, Wang, and Yu) are two of the most common techniques for anonymization, both of which preserve the truthfulness of the data. Generalization replaces the value of QIDs with more general values commonly using a value generalization hierarchy. Suppression replaces some values in the original data with a symbolic character (e.g., “\*”) to prevent the data from disclosure.

As an illustrative example, consider the loan approval dataset in Table 2.2.

In Table 2.2, *Name* is a PID, *Age*, *Education Level*, *Gender*, *ZIP code* and *Marital Status* are QIDs, *Salary* is a NQA and *Loan Approval* is a SA. To preserve the privacy, the PID is to be removed and the QIDs have to be anonymized. If anonymization is not performed, the adversary can use the publicly available data from Table 2.3 and find out the sensitive information about the individuals. Such malicious act by an adversary is a record linkage attack.

Table 2.4 shows a 2-anonymous version ( $k = 2$ ) of Table 2.2. That is, every record in Table 2.4 has at least one other record that shares the same value of all QIDs. Table 2.4 has 4 equivalence classes. Some records suppress the *Gender* attribute whereas some suppress *Marital Status* attribute. Further, the last one or two digits of *ZIP code* have been replaced by a “\*” for anonymization purpose. The values of *Age* attribute are replaced with ranges of values.

Table 2.2: Original loan approval dataset

<b>PID</b>	<b>NQA</b>	<b>QIDs</b>					<b>SA</b>
<b>Name</b>	<b>Salary</b>	<b>Age</b>	<b>Education Level</b>	<b>Gender</b>	<b>ZIP code</b>	<b>Marital Status</b>	<b>Loan Approval</b>
A	30000	26	Bachelors	Male	360001	Single	No
B	40000	39	Masters	Female	380054	Divorced	Yes
C	26000	33	Diploma	Male	360001	Married	Yes
D	52000	40	Masters	Male	380008	Married	Yes
E	24000	34	Bachelors	Male	360002	Married	Yes
F	42000	44	Masters	Male	380054	Divorced	No
G	20000	29	Diploma	Female	360001	Married	No
H	26000	35	Masters	Male	380054	Married	No
I	21000	43	Bachelors	Male	360005	Divorced	No
J	19000	37	Bachelors	Male	360005	Single	No

The  $k$ -anonymity model is broader to some extent and here only its basic functionality is described. Since it is proficient in preserving the privacy in several real-life applications and presents a theoretical foundation for privacy-related legislation (Friedman, Wolff, and Schuster), the framework proposed in this thesis uses it for preserving the privacy of data streams.

$k$ -anonymity can be guaranteed using two ways (Friedman, Wolff, and Schuster): One approach is to anonymize the dataset first and then mine the  $k$ -anonymous data. Another approach involves mining the data first and then performing anonymization on the data mining result. The first approach may hide some facts that are vital for data mining. Further, as described in the beginning of this section, sanitizing the

## CHAPTER 2. BACKGROUND

Table 2.3: Publicly available data

<b>Name</b>	<b>Age</b>	<b>Education Level</b>	<b>Gender</b>	<b>ZIP code</b>	<b>Marital Status</b>
A	26	Bachelors	Male	360001	Single
B	39	Masters	Female	380054	Divorced
C	33	Diploma	Male	360001	Married
D	40	Masters	Male	380008	Married
E	34	Bachelors	Male	360002	Married
F	44	Masters	Male	380054	Divorced
G	29	Diploma	Female	360001	Married
H	35	Masters	Male	380054	Married
I	43	Bachelors	Male	360005	Divorced
J	37	Bachelors	Male	360005	Single

Table 2.4: 2-anonymous version of Table 2.2

<b>EC</b>	<b>Salary</b>	<b>Age</b>	<b>Education Level</b>	<b>Gender</b>	<b>ZIP code</b>	<b>Marital Status</b>	<b>Loan Approval</b>
1	20000	[25-35)	Diploma	*	360001	Married	Yes
	26000	[25-35)	Diploma	*	360001	Married	No
2	30000	[25-35)	Bachelors	Male	36000*	*	No
	40000	[25-35)	Bachelors	Male	36000*	*	Yes
3	21000	[35-45)	Bachelors	Male	360005	*	No
	19000	[35-45)	Bachelors	Male	360005	*	No
4	36000	[35-45)	Masters	*	3800**	Divorced	Yes
	52000	[35-45)	Masters	*	3800**	Married	Yes
	24000	[35-45)	Masters	*	3800**	Divorced	No
	42000	[35-45)	Masters	*	3800**	Married	No

data mining output is more effective. Since the target of the work proposed in this thesis is to preserve the privacy of the data mining output, the second approach is adopted in the proposed framework. However,  $k$ -anonymity is susceptible to attribute linkage attack (also known as homogeneity attack). This attack occurs when all the values of SA within an EC are same. Thus even if an EC has  $k$  or more records, an adversary can easily discover the SA value of an individual within an EC. In Table 2.4, both of the records in the third EC block have the same value of SA depicting that loan applications of individuals I and J were rejected.

Table 2.5: 2-diverse version of Table 2.2

EC	Salary	Age	Education Level	Gender	ZIP code	Marital Status	Loan Approval
1	20000	[25-35)	Diploma	*	360001	Married	Yes
	26000	[25-35)	Diploma	*	360001	Married	No
2	30000	<45	Bachelors	Male	36000*	*	No
	40000	<45	Bachelors	Male	36000*	*	Yes
	21000	<45	Bachelors	Male	36000*	*	No
	19000	<45	Bachelors	Male	36000*	*	No
3	36000	[35-45)	Masters	*	3800**	Divorced	Yes
	52000	[35-45)	Masters	*	3800**	Married	Yes
	24000	[35-45)	Masters	*	3800**	Divorced	No
	42000	[35-45)	Masters	*	3800**	Married	No

This limitation of  $k$ -anonymity is overcome by a stronger notion of privacy called  $l$ -diversity. The main idea behind  $l$ -diversity is that the values of SA in each EC should be well-represented (Machanavajjhala et al.). Table 2.5 represents a 2-diverse version of Table 2.2. It can be seen that the attribute linkage attack against a 2-anonymous table is prevented by a 2-diverse table.

Thus,  $k$ -anonymity and  $l$ -diversity principles prevent record linkage and attribute linkage attacks respectively. Any data classification method that uses these principles can suitably perform privacy-preserving data classification. As stated earlier,

apart from these two attacks, few other attacks like table linkage also occur on data, but the privacy models proposed to prevent such attacks aren't presented here as they are out of the scope of this work.

Rather than sanitizing the data first and then mining it, it can also be beneficial to mine the data first and then perform anonymization, or to perform both the tasks together. Friedman, Wolff, and Schuster propose a method for directly building a  $k$ -anonymous decision tree from a private table. Here, both mining and anonymization are carried out in a single process. In the beginning, the decision tree consists of only the root representing all the tuples in a private table. At any given stage of induction, while splitting a node in the tree, the algorithm selects the attribute in the quasi-identifier with the highest gain (considering Information Gain or Gini Index as attribute selection measures), only if the split does not violate  $k$ -anonymity. Whenever splitting a quasi-identifier causes a breach of anonymity, a generalized version of that attribute is selected as a potential candidate for splitting the node. The algorithm terminates when no further node can be inserted without compromising  $k$ -anonymity. Friedman, Wolff, and Schuster show how this hybrid method of mining and anonymization together is better than doing the said tasks separately. The proposed framework uses a variant of this technique.

- **Miscellaneous techniques:**

Peng et al. proposes an ensemble method to perform privacy-preserving distributed data classification among sites. When the data is homogeneously distributed, each site constructs a decision tree classifier from the local data available with it, and a central trusted party integrates these results by producing a classifier ensemble. This ensemble is used by all the sites to classify the unseen data. This method is simple and produces accurate classifiers.

Emekci et al. proposes a privacy-preserving decision tree learning method based on the ID3 algorithm and Shamir's secret sharing for homogeneously distributed data. Shamir's secret sharing method operates in three phases and is used to compute

summation of the secret values over  $n$  parties without revealing the secrets to other parties. In the first phase, each party has a secret value and they choose a random polynomial of degree  $n-1$ . The constant term in the polynomial is the secret value. Also, each party creates a random number of its own and reveals it to others. Using Shamir's secret sharing algorithm, each party computes the share of all other parties based on the random numbers revealed by them and sends the respective shares to all the parties. In the second phase, each party performs the summation of the shares it obtains from other parties and sends this intermediate result to all parties. In the final phase, each party solves the set of equations to find the sum of secret values using the intermediate results received from the second phase. Hence, this method lets all the parties find the conditional entropy of each attribute from the data at all the parties. Finally, from this conditional entropy, the best attribute at a node can be determined and the tree can be induced. This method is scalable up to a large number of parties but suffers from information leakage issues.

A framework with a multi-round algorithm for classification of homogeneously distributed data using privacy-preserving  $k$ -Nearest Neighbor (kNN) classifier is proposed by Xiong, Chitti, and Liu. In case of distributed environment, an instance's  $k$  nearest neighbors may be distributed among several nodes. That is, each node will contain a few data tuples that are  $k$  nearest neighbors of each query instance. Hence, the classification process is divided into two steps: In the first step, the tuples in the database at each node that belong to  $k$  nearest neighbors of the query instance  $q$  (locally) are selected. Further, a privacy-preserving algorithm is applied to identify  $k$  nearest neighbors between the tuples in the union of the databases and query instance  $q$  (globally). In the second step, each node classifies  $q$  locally and all the nodes cooperate to determine the classification of  $q$  globally, in a privacy-preserving way. Higher the value of  $k$ , more the privacy is protected.  $K$ -nearest neighbor being a lazy learning technique, this method does not output any classifier model which is generally required by several applications.

In a nutshell, the vast literature concerning privacy-preserving data classification facilitates selection of suitable methods for different applications. In the next chapter, few of these techniques have been implemented and evaluated for performance comparison.

## 2.3 Genetic Programming

Genetic Programming (GP), pioneered by Koza, is an evolutionary algorithm that follows the principle of “Survival of the Fittest” laid down by Charles Darwin. GP has emerged as an extension of Genetic Algorithms (GA), however, unlike GA, which generally uses fixed sized binary strings to codify individuals; GP uses a variable sized tree structure.

GP is an innovative search and optimization technique that has been widely applied to solve numerous real-world problems of classification (Lee et al.; Espejo, Ventura, and Herrera). Due to its flexible nature, GP can be employed to induce classifiers using different kinds of representations like decision trees, classification rules, etc. The induction proceeds by searching a space of candidate classifiers and eventually producing an efficient one. GP is also commonly used for optimizing algorithmic parameters.

GP uses several key parameters and components, which are described as follows (Koza; Jabeen and Baig):

- GP individuals: GP works on a population of individuals unlike the other techniques that operate on only one solution. Generally, the initial population of GP is generated by forming variable sized trees randomly using the primitive set of functions (that form internal nodes) and terminals (that form leaf nodes). The terminal and function set used in GP is typically driven by the nature of the problem domain. Terminals may be numeric constants whereas the function set encompasses the basic operations applicable on the terminals with a goal to obtain the desired output. GP can prevent trees from growing too large by setting a maximum tree size threshold.
- Population size: The number of individuals in GP population may range from 30 to 10,000 as per the requisite and nature of the application.



- **Fitness function:** The fitness function is a measure to evaluate the performance of a GP individual and guide the search process of GP. Fitter individuals are more apt to be selected to take part in the procreation of the next generation of individuals, thus, increasing the probability that its genetic material will survive throughout the evolutionary process. Any preference criterion (e.g. accuracy in case of classification) can be expressed in terms of the fitness function.
- **Genetic operators:** Genetic operations are applied on individuals to produce new and expectedly efficient individuals for the next generations. Reproduction, crossover and mutation are the genetic operators used.
  - **Reproduction:** To ensure that the fitness of the best individual in a population is never less than the best of previous generations, the reproduction operator is used. Reproduction consists of simply copying some individuals of a generation's population directly to the next. Elitism, that is a reproduction method, copies few individuals with largest fitness to the population of next generation.
  - **Mutation:** Mutation operator injects new genetic material into the population through a randomly generated individual. A mutation point is randomly chosen in an individual (GP tree) and the sub-tree rooted at that mutation point is replaced by the newly generated individual. The mutation process is performed to introduce diversity in individuals and is pictured in Figure 2.4.
  - **Crossover:** Crossover operator selects two individuals from the population and copies them to a mating pool. A crossover point is randomly chosen in each individual (GP tree) and the sub-trees rooted at those crossover points are swapped. The two newly produced individuals are added to the new population. This is pictured in Figure 2.5.
- **Selection method:** The literature presents various methods to select individuals for applying genetic operations and forming the new population. Some popular methods are: rank selection, proportional selection, tournament selection, etc. (Koza).

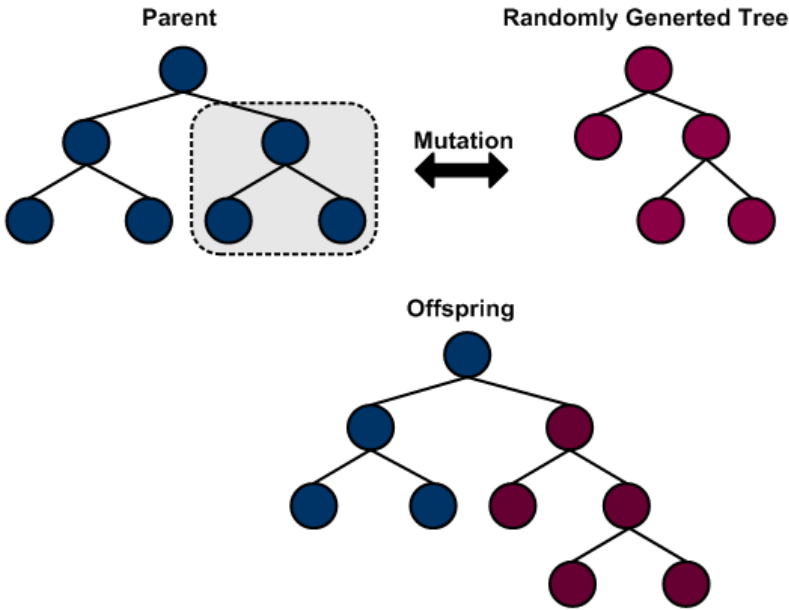


Figure 2.4: Sub-tree mutation

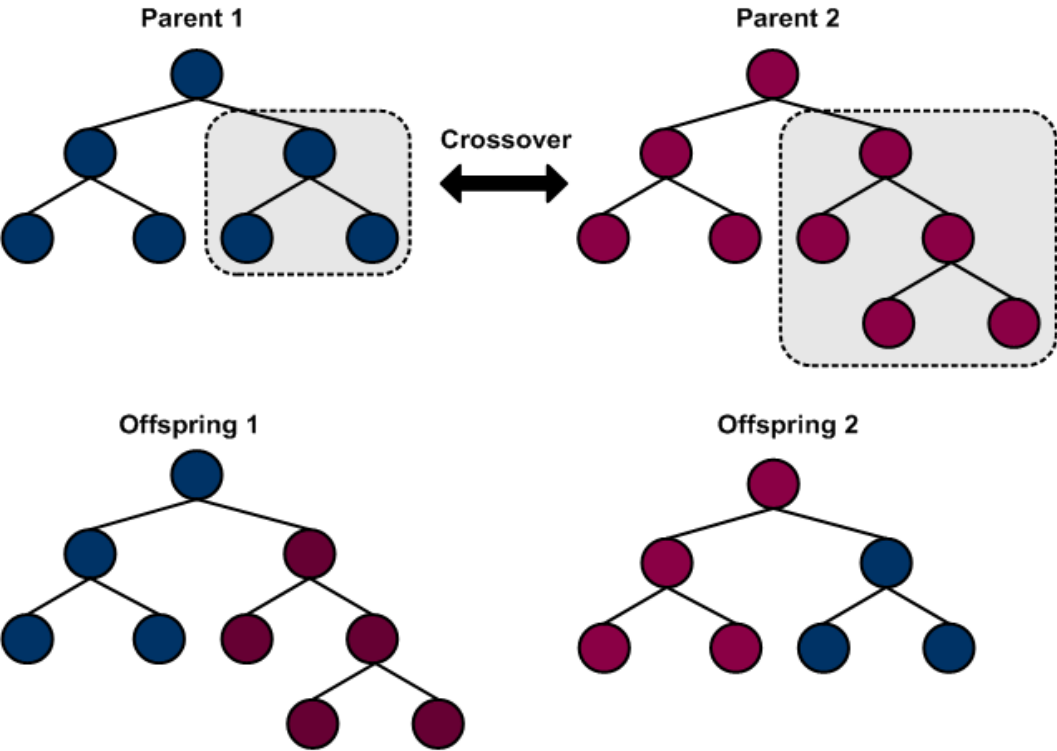


Figure 2.5: Sub-tree crossover

- Using rank selection, individuals in the population are ranked based on their fitness, from first to last. The probability of selection of an individual is based on the rank of an individual.
  - In proportional selection, the probability of selection of an individual is proportional to its fitness function value. Proportional selection can be thought as spinning a roulette wheel where each segment on the wheel corresponds to an individual and the size of the segment is proportional to the corresponding individual's fitness.
  - Under tournament selection method, few individuals from the population are randomly selected for a tournament. Based on their fitness, these individuals within the tournament compete for being selected to pass genetic material into the next generation.
- Maximum number of generations: GP process continues for a number of generations. The maximum number of generations for which GP can run before terminating varies from 20 to thousands.
  - Termination Criteria and solution: The GP runs terminate when either the maximum number of generations have been produced or a problem-specific successful solution has been obtained. The best-of-run individual from the last generation is then designated as the final result.

The basic GP process is depicted graphically in Figure 2.6:

GP initiates by forming a random initial population of functions and terminals. A run begins by evaluating the fitness of each individual and applying genetic operation as per the rate of each operator. Based on their fitness, the individuals get selected for undergoing genetic operations (1 individual for reproduction as well as mutation and 2 individuals for crossover). A generation is incremented when the required individuals are produced. The GP cycle terminates when desired number of generations is reached or the termination criterion is fulfilled.

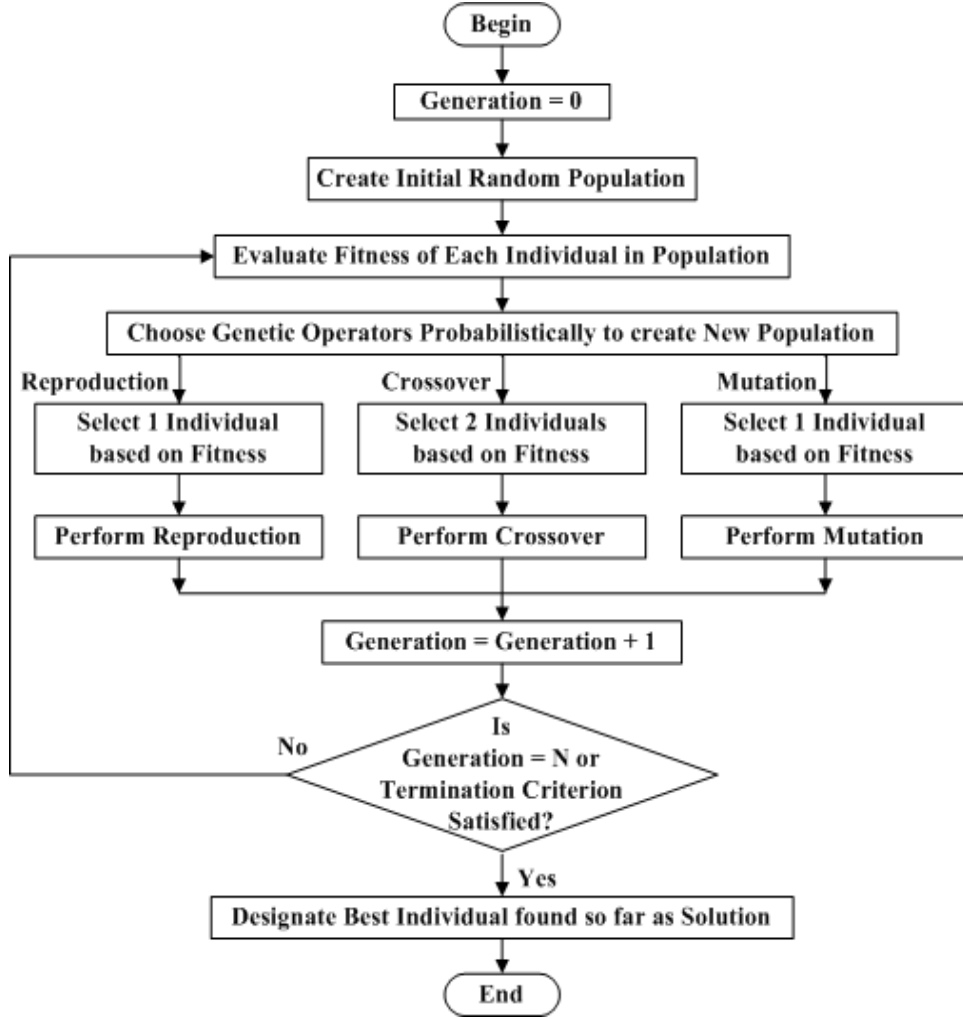


Figure 2.6: General process of Genetic Programming

The efficiency of GP has been evaluated extensively by researchers and a detailed literature of application of GP for classification is presented in Chapter 6. Further, a GP-based approach is employed in this work for privacy-preserving classification of horizontally partitioned data streams is proposed in Chapter 6.

## 2.4 Ensemble Learning

Traditional data mining techniques train a single model to solve the target problem. Ensemble methods (Zhou; Han, Kamber, and Pei) also known as ensemble learning or

committee-based learning, refers to procedure of training multiple learners rather than inducing on a single learner to discover useful patterns from the data. Ensemble learning works by forming a committee of decision-making models and using the combined opinion of the entire committee to predict the output of a new query. The general architecture of an ensemble is shown in Figure 2.7.

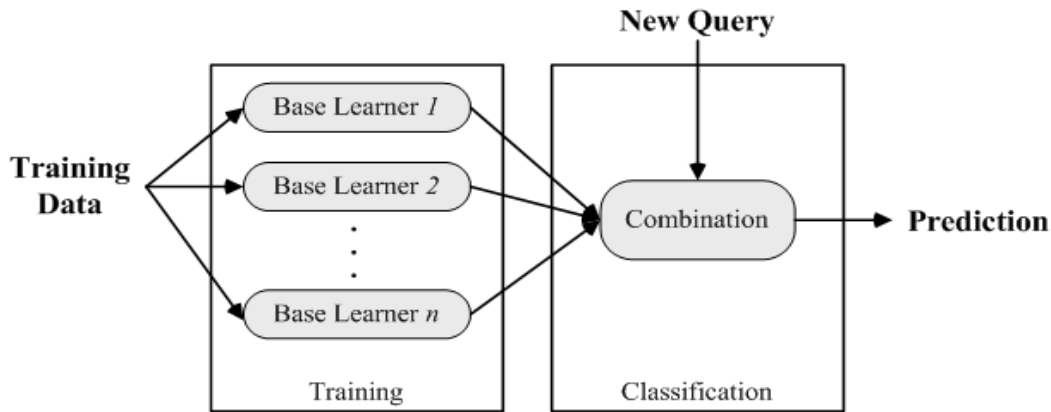


Figure 2.7: A general ensemble architecture

Ensemble classifiers are gaining increasing attention of researchers due to its efficacy as compared to single classifiers. The base learners of an ensemble classifier can be neural networks, decisions trees, etc. When all the base learners are of same type, i.e. use the same learning algorithm, the ensemble is referred as a homogeneous ensemble whereas when the base learners use different learning algorithms, the ensemble is known as heterogeneous ensemble. The computational cost of inducing several learners is not too large as compared to single learner because several learners are commonly being induced for tuning the algorithmic parameters. Further, the cost of combining the base learners is small and acceptable.

### 2.4.1 Ensemble methods

This sub-section describes few of the most popular ensemble learning methods.

- Bagging: Bagging (Breiman), which stands for bootstrap aggregation is an effective and successful method of ensemble learning. Originally designed for decision trees,

bagging can be used with various classifiers. It works by inducing  $k$  base classifiers, each trained on data sampled from the original training dataset. Bagging uses sampling with replacement and hence some instances may be used by more than one classifier whereas some instances may not be used by any classifier. To classify a previously unseen instance  $x$ , each classifier yields its class prediction, counted as one vote for that class. The bagged classifier (ensemble) assigns the class with highest number of votes to  $x$ .

Although bagging is mainly used for predicting categorical class labels, continuous attribute values are predicted by averaging the output of each learner. Bagging is a simple technique and its effectiveness increases when the base classifiers are unstable. The accuracy of Bagging-based ensemble classifier is high because employing multiple classifiers reduces the variance.

- Random Forests: As its name suggests, Random Forest (RF) (Breimann; Zhou) forms a forest of trees (decision trees) induced using randomness trait. It is an extension of Bagging and is based on randomized feature selection. While inducing the decision tree base classifier, at each node, RF selects a subset of attributes randomly and follows the traditional node splitting procedure using only those selected attributes.

It is to be noted that RF introduces randomness only in filtering attributes for different nodes. Once an attribute is selected at a node, the split point is chosen similar to traditional decision tree induction. Since nodes in decision trees of RF have to be evaluated for only a subset of attributes, the training time of RF is less than Bagging of traditional decision tree classifiers.

- Boosting: Boosting (Han, Kamber, and Pei; Schapire et al.) is a popular meta-algorithm initially proposed for classification only. Boosting iteratively induces a chain of classifiers from the training data by sampling with replacement. Initially, the training instances are assigned equal weights. But, after the induction of each classifier  $h_i$ , the weights are updated so as to make the successive classifiers  $h_{i+1}$

concentrate more on the training instances misclassified by  $h_i$ . Weights of misclassified instances are increased whereas weights of correctly classified instances are decreased. To classify a previously unseen instance  $x$ , the votes of each base classifier in the boosted ensemble classifier  $H$  are combined. Further, each classifier's vote is weighted by its accuracy and the class that receives the highest votes is assigned to instance  $x$ . Boosting is little time-consuming but works efficiently due to the accuracy weighted voting mechanism.

Apart from these, several techniques of ensemble learning like Decorate (Zhou), etc. have been proposed and evaluated by researchers. The literature shows that ensemble classifiers have proved to be much more successful in solving real-world problems as compared to single classifiers. Chapter 7 utilizes ensemble learning method to improve the performance of privacy-preserving distributed data stream classifier.

## Chapter 3

# Empirical Evaluation of Preliminary Components

Subsequently after a detailed study of the existing literature on the sub-areas of the work presented in this thesis, this chapter presents an empirical evaluation of the two major components that form the heart of the work: data stream classification and privacy-preserving classification of horizontally partitioned data. The goal of the experiments is to identify suitable techniques for each component and utilize the merits of these techniques while proposing a solution for “privacy-preserving classification of horizontally partitioned data streams”.

### 3.1 Data Stream Classification

Conventional data classification algorithms assume data instances can be retrieved for a small cost. Further, these conventional algorithms publish the classifier after analyzing the entire dataset. The key requirements of any data stream classifier include working in limited time, processing examples without performing multiple scans on the data and having readiness to publish the classifier at any point when needed. The ability of existing

---

Part of this chapter appears in: Radhika Kotecha and Sanjay Garg, “Data Streams and Privacy: Two Emerging Issues in Data Classification”, Proceedings of 5<sup>th</sup> Nirma University International Conference on Engineering, IEEE (2015)



data stream classifiers to fulfill these requirements has been evaluated in this section.

Further, the focus of batch learning algorithms is to reuse the data so as to get the maximum from the limited data available. This is not a concern while learning from data streams as abundant data is available. The bifurcation between which data instances to use for training and which to use for testing the model is made by the evaluation procedure of every learning algorithm.

In order to extract the most out of the data, batch learning generally uses a k-fold cross-validation method of evaluation, with k set equal to 10 most frequently. But as the size of the data increases, repeating the training several times is not feasible due to limited time constraints. Hence, for large datasets, batch learning focuses on reducing the number of folds or using a single hold-out set as it would require less computational effort. Data stream classification is an emerging area of research and a limited study on the evaluation methods is conducted as compared to literature on batch setting. But, the survey of Bifet et al. states that the most commonly used evaluation procedure for data stream classification is to employ a single hold-out set. A precise measurement of accuracy can be obtained by setting aside a large number of instances for testing intention without making the classification learning algorithm starve for training instances. This conclusion is consistent with the argument that evaluation procedure of data stream mining algorithms is simple as no changes in batch setting evaluation method is required. Moreover, rather than assuring the reliability of a data mining model by several repeated runs, a large number of testing instances can be employed to guarantee reliability.

A possible way to create a hold-out set of instances for evaluating a data stream classifier can be to collect a bunch of instances that have not been employed for training. These instances from the data stream are then used as test instances.

But, since the data stream is considerably large, instead of using a single hold-out, it is preferable to evaluate the model periodically. This method, called periodic hold-out evaluation is used to track the performance of the model over the time and to provide even more precise information about the accuracy of the model. The method involves periodically using batch of instances as testing instances. The final accuracy of the model

is derived from the aggregate results of testing.

Another method for evaluating data stream algorithms, called ‘prequential’, interleaves testing along with training. In this method, each individual instance is used to test the classifier before employing it for training and the accuracy is updated incrementally. Using this method, the classifier is always being tested on instances it hasn’t seen before. This method has a disadvantage that the classifier is punished for making mistakes earlier irrespective of its capability of achieving high accuracy eventually. This effect decreases after observing a large number of instances, but as per Bifet et al. the accuracy of classifier evaluated using this prequential stays always less than the accuracy obtained by evaluating using hold-out method.

After analysis of the relative advantages of both the methods, the hold-out method of evaluation as it proves to be satisfactory in most of the cases and is foundation of several experimental frameworks presented in literature.

Most of the existing work does not make explicit comments on memory utilization. Instead, limits on usage of memory are enforced either by allocating predefined memory or analyzing the number of scans the algorithm makes on the data.

Details of experiments performed and results obtained for some of the popular data stream classification algorithms are presented herewith.

### 3.1.1 Data streams

Experiments are performed on both synthetic and real-life data streams with the number of instances varying from lakhs to millions. Table 3.1 lists out the four data streams used along with their details. Forest Covertype and Waveform data streams are available on the UCI machine learning repository (Lichman) whereas Loan Approval and Rotating Hyperplane are synthetically generated data streams. Further, Waveform and Rotating Hyperplane data streams have concept-drifts.

Table 3.1: Composition of data streams

<b>Data Stream</b>	<b>No. of Attributes</b>	<b>No. of Instances</b>	<b>No. of Classes</b>	<b>Concept- drift</b>
<b>Forest Covertypes</b>	54	5.8 lakhs	7	No
<b>Waveform</b>	40	1 million	3	Yes
<b>Loan Approval</b>	9	10 million	2	No
<b>Rotating Hyperplane</b>	10	10 million	2	Yes

### 3.1.2 Implementation details

In order to identify an efficient data stream classifier for the problem targeted in this work, five different data stream classification algorithms have empirically been evaluated and compared on the data streams described in Table 3.1. Experiments are performed for Naive Bayes classifier, Rule-based classifier, Hoeffding tree classifier (or VFDT), Hoeffding Adaptive Tree using Adaptive Windowing (HAT-ADWIN), and Accuracy Weighted Ensemble classifier.

As described in Chapter 2, Naive Bayes and Rule-based classifier SCALLOP are simple techniques for data stream classification and hence are considered here for evaluation. Hoeffding tree classifier is evaluated as it is a variant of the successful decision tree classifier specifically designed for data streams. HAT-ADWIN uses Hoeffding tree classifier but addresses the issue of concept-drift using windowing technique. Hence, its performance is assessed in presence and absence of concept-drifts (using respective data streams). Accuracy Weighted Ensemble classifier is evaluated as ensemble methods have proved to be extremely successful in literature. The base learners of this ensemble consist of traditional decision tree classifiers built from chunks of data using random subset of features. An ensemble of 5 learners is induced and evaluated. Before testing phase, some of the examples are accumulated at random time intervals and are used to calculate the accuracy of each member of the ensemble. When the classifier is to be applied to unseen instances (for testing or prediction of unlabeled instances), the approach by Wang et al. is

followed. That is, classifiers in the ensemble are weighted proportionally to their accuracy.

All five techniques have been implemented in Massive Online Analysis (MOA) (Bifet et al.). MOA is an open source framework for data stream mining that includes a collection of machine learning algorithms for evaluation. MOA is related to the popular open-source workbench Waikato Environment for Knowledge Analysis (WEKA) that includes implementations of extensive series of batch machine learning techniques.

Having considered the relative merits of different methods of evaluation, periodic hold-out evaluation is employed in the experiments conducted. The popular setting (Han, Kamber, and Pei) of dedicating 66% of data stream instances for training and rest for testing is applied. Periodically, testing is performed on a bunch of instances held out. The final classification accuracy and time reported is the aggregate of the results obtained periodically.

### 3.1.3 Results

The results of training accuracy, predictive accuracy, training time and prediction time of all five classifiers on the stated four data streams are presented using Figure 3.1 to Figure 3.4 respectively.

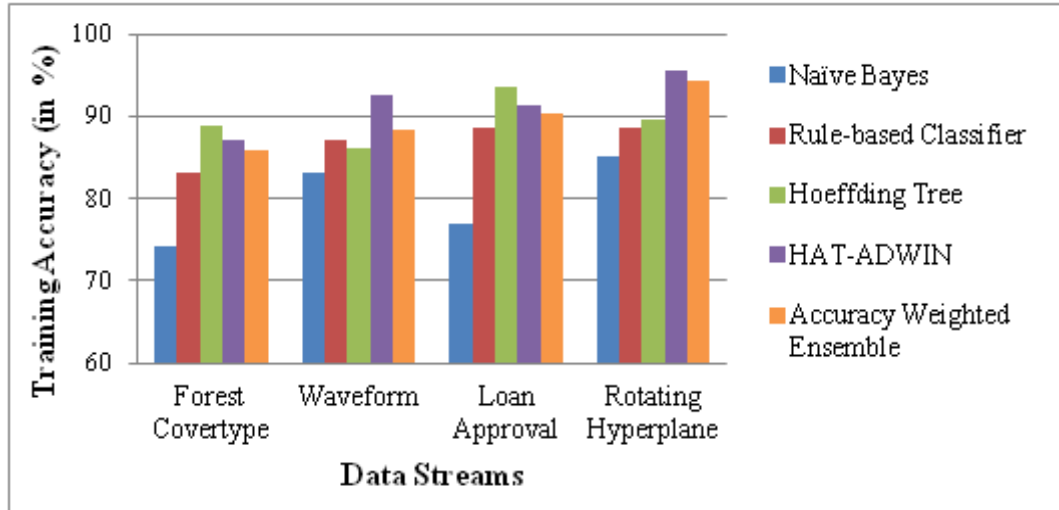


Figure 3.1: Training accuracy of classifiers (in %)

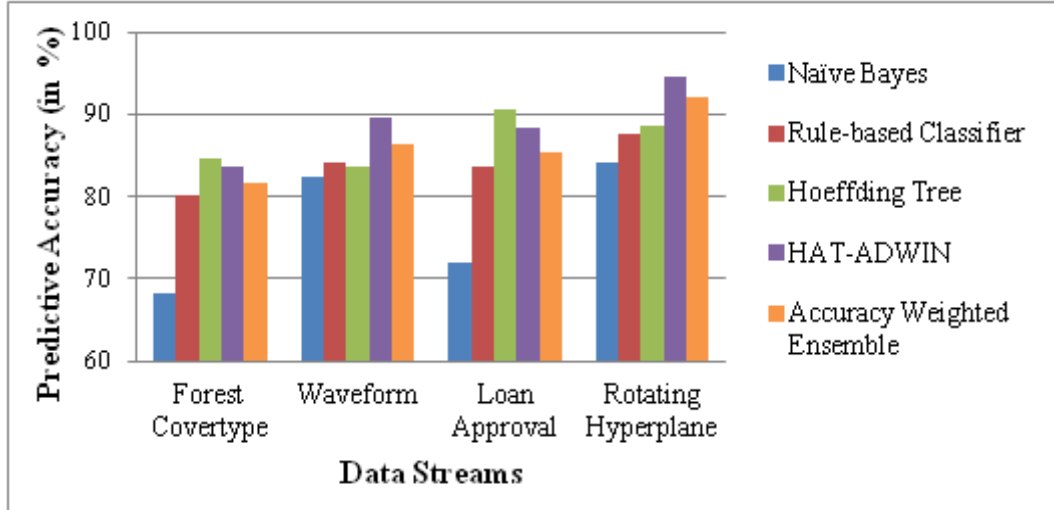


Figure 3.2: Prediction accuracy of classifiers (in %)

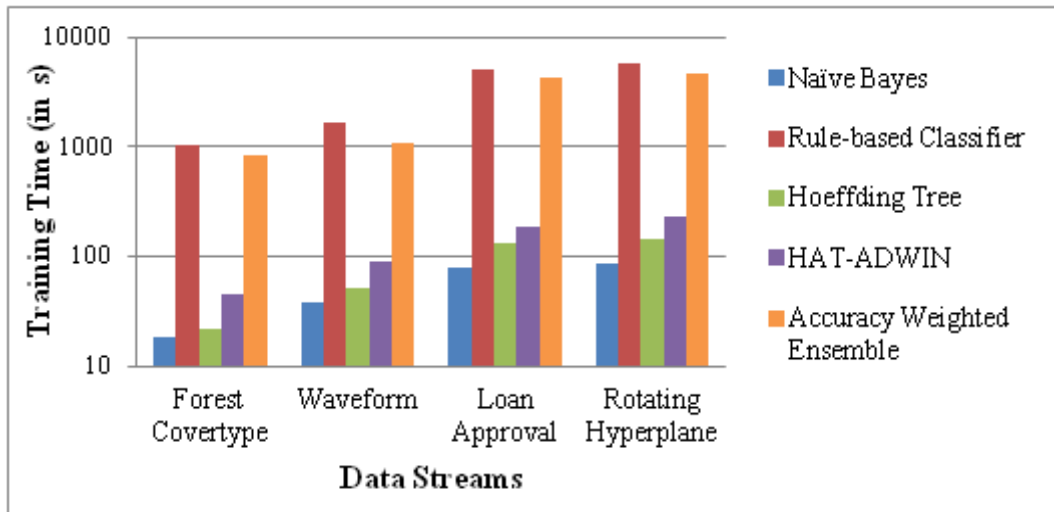


Figure 3.3: Training time of classifiers (in s)

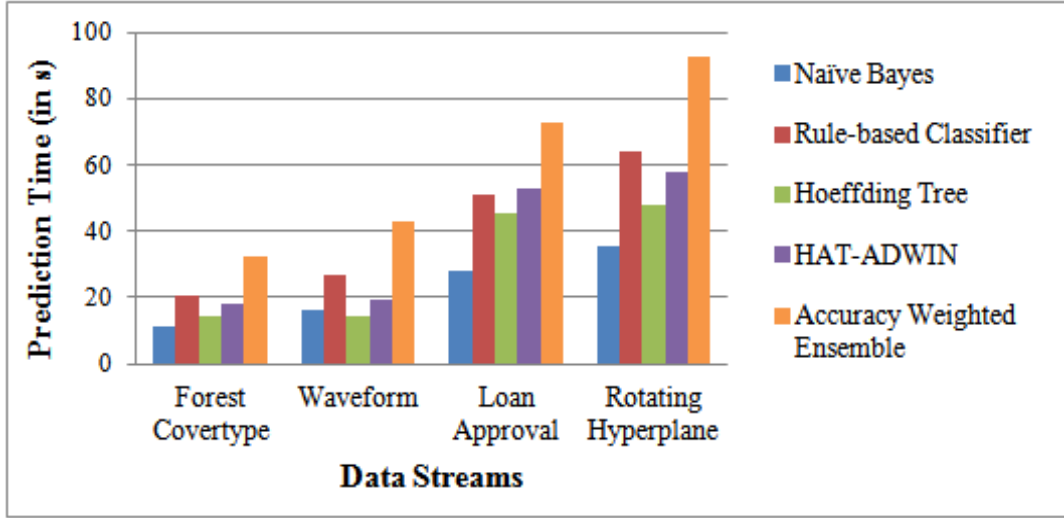


Figure 3.4: Prediction time of classifiers (in s)

From the results, it can be seen that Naive Bayes classifier is quick in training and evaluating all types of data stream instances as it requires only maintaining a statistics table. But its prediction accuracy is low as it assumes that attributes are independent from one another. In conditions when very limited time is available for classifier learning and prediction, Naive Bayes technique can be applied with a little compromise on accuracy.

Rule-based classifier requires a lot of time for training due to the procedure followed for maintaining the rule set on arrival of each new example and rule-refinement. The prediction time gets high when no rule covers some of the test instances and a voting mechanism is to be applied. But, the training and prediction accuracy of rule-based classifier is higher than Naive Bayes classifier due to its precise procedure of rule-set induction and maintenance. However, when a concept-drift occurs, the old rule-set does not get completely replaced and hence this classifier offers very little support to the issue of concept-drift.

From the results of Figure 3.1 and Figure 3.2, it can be observed that in absence of concept-drift, the training as well as prediction accuracy of Hoeffding tree classifier has always remained highest among all the classifiers considered. In fact, Hoeffding tree classifier shows remarkably good results in an acceptable time, for both training as well as prediction. But when the data streams have concept-drifts; as in Waveform and Rotating

Hyperplane streams, the accuracy of Hoeffding tree classifier falls due to its inability to cope up with concept-drifts.

For data streams with concept-drift, the Accuracy Weighted Ensemble classifier is accurate enough as the ensemble member learned from the instances with concept-drift will be given more weight gradually. In absence of concept-drift, its accuracy is comparable to the Hoeffding tree classifier which is specifically designed for classifying data streams. But with the increase in number of attributes or data instances, the time required in training and prediction is high. The large amount of time makes it less suitable but the approach can be modified and its merits can be used to build an efficient classifier.

In terms of accuracy, HAT-ADWIN can be regarded as an efficient classifier in presence as well as absence of concept-drift. However, the time taken by this approach for training is higher as compared to Hoeffding tree classifier due to its periodic procedure of examining a concept-drift in the data. But, as a result of this periodic examination, the accuracy of HAT-ADWIN does not degrade when there is a concept-drift in the data. Further, as compared to Accuracy Weighted Ensemble classifier, the training as well as prediction time required by HAT-ADWIN is very less and training plus prediction accuracy is high.

It can be concluded that Hoeffding tree classifier is an efficient candidate for data stream classification and the windowing technique of its variant HAT-ADWIN is advantageous in dealing with concept-drift too. Thus, the proposed solution approach to the target problem “privacy-preserving classification of horizontally partitioned data streams” is based on Hoeffding tree classifier and concept of windowing to address concept-drift.

## 3.2 Privacy-preserving classification of horizontally partitioned data

As described in Chapter 2, several techniques for privacy-preserving data classification have been proposed and implemented by researchers. The issue of preserving privacy gets even more complicated when the data is partitioned between multiple sites. Since the

target problem and the scope of the work covers classifying horizontally partitioned data, it is required to have an evaluation and analysis of performance of the existing privacy-preserving classification techniques on data that is horizontally partitioned. Some of the efficient techniques proposed in the literature haven't been applied for distributed, or specifically, homogeneously distributed data. Moreover, application of these techniques on few common datasets is required for a detailed and precise analysis of their suitability for the target problem.

Hence, several experiments are performed to compare the performance of different ways of inducing privacy-preserving classifier where the data tuples are stored at multiple autonomous sites. The goal of each of these techniques is to build a global classifier from the data that is horizontally partitioned between multiple participating sites. As stated in the motivation section of Chapter 1, these parties are competitors but are collaborating for mutual benefits in business. Hence, it is required to induce a privacy-preserving classifier without disclosing the raw data of any party to the other participating parties or to the public.

Details of datasets used, experiments performed and results obtained for some of the popular privacy-preserving data classification methods are presented in the subsequent sub-sections.

### **3.2.1 Data sets**

For comparing the performances of different methods for privacy-preserving data classification, experiments on four data sets from various real domains were conducted. These data sets are available on UCI machine learning repository (Lichman) and its details are described in Table 3.2. These data sets contain private information which is to be protected from disclosure. Specifically, the class attribute is considered as a sensitive attribute and all other attributes are considered as quasi-identifiers which need to be protected otherwise the adversary can use the publically available data and find out the sensitive information about the individuals.



Table 3.2: Composition of data sets

Data Set	No. of Attributes	No. of Instances	No. of Classes
Transfusion	5	748	2
Diabetes	9	768	2
Bank Marketing	8	4521	2
Spambase	57	4601	2

### 3.2.2 Implementation details

Experiments are conducted to compare the performance of following four different approaches where each approach uses the well-known decision tree induction algorithm CART (Breiman et al.) as the base learner:

- (i) Trusted third party: This approach assumes the existence and availability of trusted third party where the data of all the sites is accumulated centrally. A classifier is induced at this central site and this global classifier is then sent to all the participating parties.
- (ii) Secure multiparty computation: In this approach, all the parties calculate the attribute measures using SMC and produce a local classifier at their respective site. The induced classifier is same at all the sites and is used as a global classifier by each of them.
- (iii) Ensemble classifier: Within this approach, each site constructs a classifier from the local data available with it and these classifiers are integrated by producing a classifier ensemble either at one of the participating sites or a third party site. The global ensemble induced hence is used by all the sites to classify the unseen data.
- (iv) Hybrid k-anonymous decision tree ensemble classifier: In literature, k-anonymous decision tree classifier (Friedman, Wolff, and Schuster) has been described which is

applicable for classifying instances at a single site. In the experiments, this method has been adopted for privacy-preserving classification of horizontally partitioned data by using the approach of method (iii), i.e. the popular ensemble classifier induction method. That is, each site induces its own  $k$ -anonymous decision tree and a global classifier ensemble from these local classifiers will be created at a non-trusted third party or at either of the parties. The value of parameter  $k$  is set equal to 5% of the total number of training instances.

All the algorithms have been implemented in MATLAB 7.8.0 (R2009a). Further, throughout the experiments, it was assumed that 3 parties want to collaboratively conduct the data classification.

To evaluate the performance of these algorithms, holdout method is chosen with 3/4th of the data used for training and rest for testing. To create an environment where the data is homogeneously distributed, the training data is divided into three random and overlapping parts with each party owning one component.

### 3.2.3 Results

The results of training accuracy, predictive accuracy and training time taken by all four methods on the stated data sets are depicted in Figure 3.5 to Figure 3.7 respectively. The prediction time taken by each of these classifiers is extremely small and hence is not shown.

Since the classifier obtained by assuming a trusted third party and the one induced using SMC is same, their training and prediction accuracies are mentioned together. Irrespective of the accuracy obtained, the former approach is not feasible because, in the competitive era, it is difficult to trust a third party.

Further, it can be observed that when a classifier is induced using SMC, the accuracy is high; but, as a large amount of communication is required, the time taken in training a classifier is very high. Moreover, with the increase in number of attributes, the communication required also increases. This is because each party has to share each of its

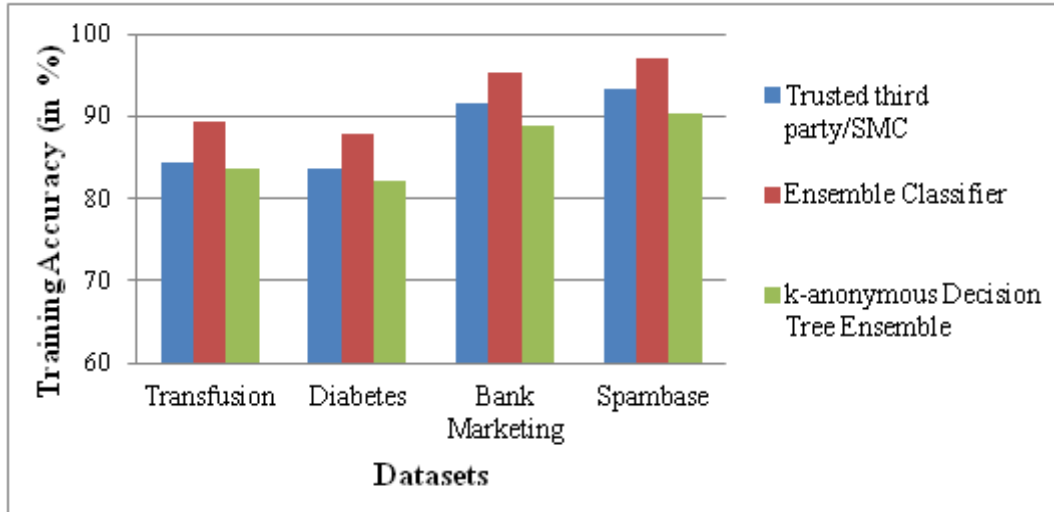


Figure 3.5: Training accuracy of classifiers (in %)

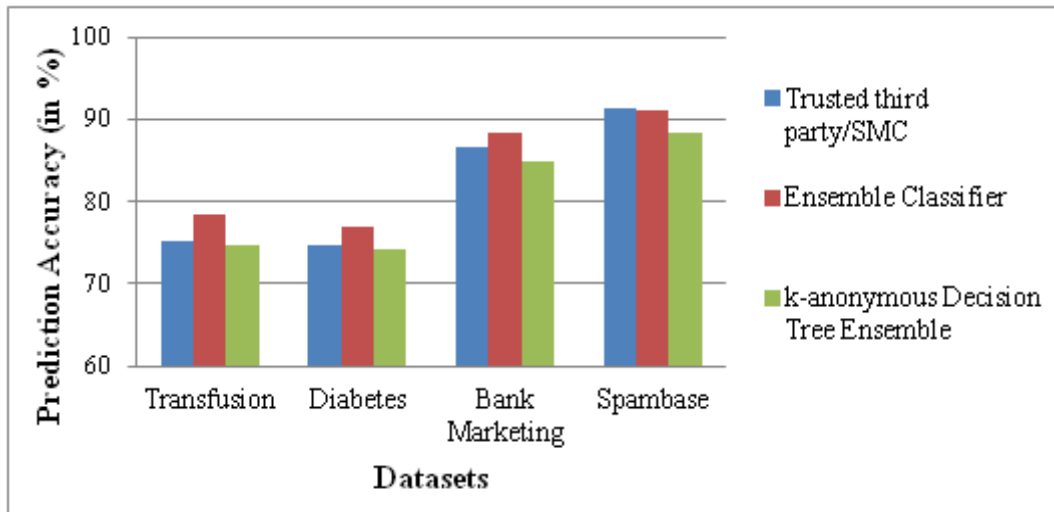


Figure 3.6: Prediction accuracy of classifiers (in %)

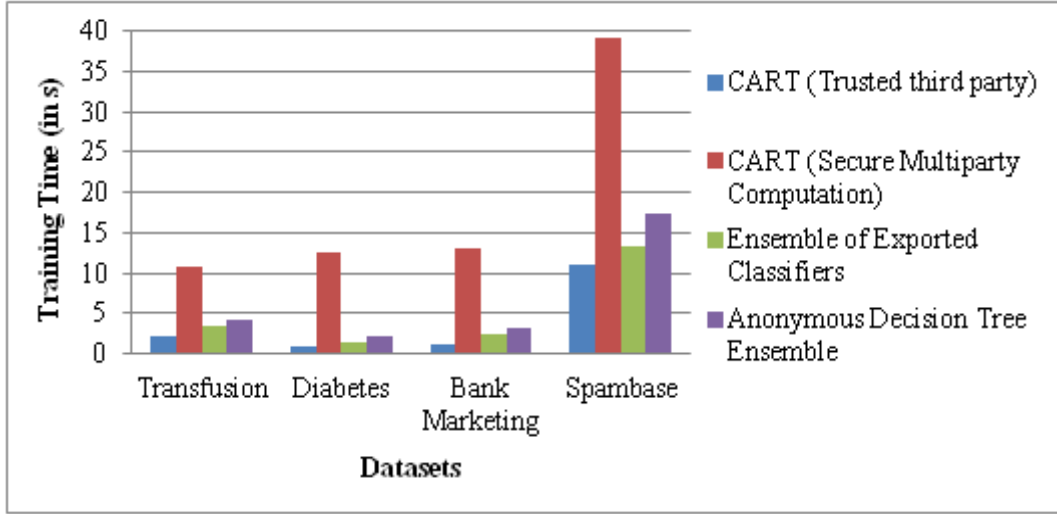


Figure 3.7: Training time of classifiers (in s)

attribute-value pair with the other parties. Hence, SMC is not a very suitable technique in today's big data epoch.

As ensemble classifiers produce more accurate results, the approach is quite suitable for privacy-preserving classification of homogeneously distributed data and the same is proved experimentally. But, conclusions about data at other sites can be easily derived from the classifiers released by those sites and privacy can be breached. Hence, it is not preferable that the participating parties release the classifiers in raw form without sanitization.

The final approach of  $k$ -anonymous decision tree classifier ensemble overcomes this disadvantage and preserves privacy to a greater extent. Also, unlike traditional privacy protection techniques such as data swapping and adding noise, information preserved using  $k$ -anonymization remains truthful. From the results in Figure 3.5 to Figure 3.7, it is clear that  $k$ -anonymous decision tree ensemble has good accuracy and the training time is also acceptable.

Thus, the proposed approach uses the concept of  $k$ -anonymous decision tree classifier ensemble to generate a privacy-preserving data classification algorithm for homogeneously distributed data.

### 3.3 Summary

This chapter empirically evaluates the existing and popular methods for data stream classification and privacy-preserving data stream classification which form the two major components of the target problem “privacy-preserving classification of horizontally partitioned data streams”.

After a deep study of the literature about these two components in Chapter 2, the experiments in this chapter remark Hoeffding tree as a suitable technique for classifying data streams and windowing as a recommendable *modus operandi* to address the issue of concept-drift in the continuously arriving data streams. Further, an ensemble of decision tree classifiers clubbed with the principle of  $k$ -anonymity is identified as an efficient approach for inducing a privacy-preserving classifier.

Looking at the merits of these approaches and their suitability for the target problem of the work in this thesis, a framework that encompasses these techniques is proposed in the next section.

# Chapter 4

## Proposed Framework

This chapter presents the framework proposed for achieving the goal of privacy-preserving classification of horizontally partitioned data streams. The key details included are a real-world application targeted by the work, brief survey of literature on data mining techniques applied to address the targeted application, generic view of the proposed approach and the implementation environment details.

### 4.1 Targeted application

Several real-world applications demand efficient privacy-preserving classification of horizontally partitioned data streams. One such application concerning decision making in banking sector is illustrated in this section and is focused throughout the work.

Due to substantial competition, banks are focusing on client-driven lending and the scope of obtaining considerable collateral from customers is shrinking. Customers pledge collateral to a bank only if no other bank are lending without such indemnity. Also, an increasing number of clients are defaulting on financial obligation by banks via credit cards or loans. As per BIS, credit risk is characterized as the potential that a bank borrower will fail to fulfill his commitment partially or fully, which is the major risk faced by banks. In banking sector, along with the loan officer's subjective assessment, success of a bank gets exceedingly dependent on a model that guides decisions of providing credits to customers.

## CHAPTER 4. PROPOSED FRAMEWORK

Specifically considering the risk assessment, several decision-making models are required in banking sector. For example: 1) New Applicant Classifier: A model that classifies a new credit applicant's repayment ability into 'good' or 'bad' groups. Such a decision support model utilizes financial conditions and demographic as well as characteristics of the new applicant. Also, along with the banks, such a model would be in the interests of borrowers too as they can identify their chances of obtaining credit. 2) Behavior Classifier: A model that classifies if an existing customer will 'default' or 'won't default'. The model also considers past payment history of customers. Further, this model not only helps banks to act upon existing customers that might delinquent but also allows analyzing the behavior of such customers and using it while lending to new customers.

In a nutshell, predicting a borrower's ability to repay financial obligations is one of the essential processes in banks' credit management decisions. Such credit risk evaluation can reduce loss and uncertainty. Much literature (Huang, Chen, and Wang; Lee et al.; Lessmann et al.; Wang et al.; Yeh and Lien; Yu, Wang, and Lai) examines the application and accuracy of classification techniques for the said task. Banks collect and analyze information of its past borrowers and build a model (say a decision tree classifier) that assigns new credit applications as either 'credit-worthy' or 'credit-risky' based on whether the applicant would default on the financial obligation.

Aiming to attract good customers and dissuade the fraudulent, the data owners (banks) may wish to release the data mining output, i.e. the classifier in public and let potential customers estimate the likelihood of getting credit obligations. The banks may also want to share their classifier among each other due to the mutual benefits it brings. However, privacy and security concerns restrict disclosing the characteristics of the clients. It is assumed that all attributes such as age, gender, marital status, income, etc. are available to the public, but the class attribute, i.e. the credit worthiness is known only to the banks. It would be unacceptable if someone uses the classifier to discover which past clients failed to repay the obligations. Hence, it is extremely important to enable 'privacy-preserving' and particularly, 'output-privacy-preserving' classification for this area. Moreover, since data in the bank is continuously arriving and classifying an

application may be required at ‘any-time’, the application can be posed as a data stream classification task.

Further, the features collected, such as age, gender, balance, average monthly deposit, etc. are the same for all banks (Lindell and Pinkas). That is, the data streams spanned across the banks are horizontally partitioned. Thus, the problem in its entirety can be considered as an application of privacy-preserving classification of horizontally partitioned data streams.

Albeit much research, the need to re-consider the recent advances in data mining (i.e. privacy requirements and streaming nature of data) for decision-making in banking is taken as the target application for this work. Undoubtedly, the proposed approach is applicable for any real-world application relating to privacy-preserving classification of horizontally partitioned data streams.

## 4.2 Data mining for credit decision-making

Much literature explores the application of data mining techniques to improve assessment of creditworthiness of customers; some of which are highlighted in this section.

Performance of four different methods: discriminant analysis (DA), logistic regression (LR), neural networks (NN) and data envelopment analysis–discriminant analysis (DEA–DA) is compared by Tsai et al. on a loan dataset with 1807 instances. Results show that DEA–DA and NN perform better than DA and LR. The authors use money attitude of customers as an attribute while inducing default predicting models.

Huang et al. apply support vector machine (SVM) for credit rating analysis on United States and Taiwan market datasets. They compare its performance with back propagation neural network (BNN) and show that SVM achieves only a minor improvement over BNN in terms of predictive accuracy. Also, number of instances in the datasets is less than 4000. Accuracy of six different data mining methods for predicting probability of default of credit card clients are evaluated and compared by Yeh and Lien. Data of customer’s default payments in Taiwan is used and artificial neural network and classification trees are



shown to perform more accurately among others. The dataset has considerable instances but lacks comparison with several hybrid and newer data mining techniques.

Stating the long training time and interpretative difficulties of neural network, Lee explores the performance of Classification and Regression tree (CART) and Multivariate Adaptive Regression Splines (MARS) and demonstrate the efficiency and suitability of these methods for credit scoring.

Huang, Chen, and Wang use a hybrid of genetic algorithms with SVM classifier for credit scoring and a genetic algorithm based feature selection technique is proposed in literature (Oreski, Oreski, and Oreski) followed by application of neural network as a classifier for predicting a borrower's loan repayment ability. Experimental results in the papers demonstrate the potential of hybrid techniques for assessing creditworthiness. Sustersic, Mramor, and Zupan also use GA-NN hybrid to develop consumer credit scoring models on accounting dataset of clients in financial institutions. They too use genetic algorithm for variable selection and BNN for model construction. BNN shows good results but the number of instances in the datasets used is small (in hundreds).

Further, Tsakonas et al. propose encoding genetic programming individuals using neural logic networks for bankruptcy prediction. The paper considers both interpretability and classification accuracy but again the dataset used is very small, i.e. contains only 118 instances. A multistage neural network ensemble learning approach is proposed by Yu, Wang, and Lai to assess credit risk. The authors illustrate the performance of the proposed approach on two credit datasets. But since these datasets are extremely small, a sampling approach is employed. The potential of ensemble classifiers as efficient alternatives for implementing credit scoring tasks is also shown by Twalaa.

Finlay applied ensemble classifier methods to classify customers with good and bad credit risks. Results demonstrate that multiple classifiers outperform single classifiers. Importantly, unlike most of the work in the literature, the methods are applied on two real-world datasets with larger number of instances (88,789 and 138606 instances). Wang et al. have also shown the effectiveness of decision tree ensemble as compared to single decision tree and neural networks on two datasets with 690 and 1000 instances respectively.

## CHAPTER 4. PROPOSED FRAMEWORK

Lessmann et al. updates benchmarking study of Baesens et al. concerning classification algorithms for credit scoring. They compare 41 classifiers with respect to 6 performances measures on 8 real-world datasets of credit scoring. The datasets are of varying sizes and the paper presents one of the most comprehensive studies of classifier performances for credit scoring. Florez-Lopez and Ramon-Jeronimo highlight that the existing data mining methods applied for credit scoring obtain high accuracy at the expense of interpretability. Thus, they propose using a classifier ensemble of merged decision trees called correlated-adjusted decision forest in order to induce an accurate as well as comprehensible classifier for credit risk problems. But again, the method is applied for a dataset that has just 1000 instances.

To sum up, the existing research applies various data mining techniques, hybridization of traditional data mining techniques with soft computing methods, advanced techniques like ensemble classifier, etc. Interpretability of the model is one of the most important factors for managerial decisions in financial sector. However, either the existing works do not consider interpretability or uses a small number of instances in the dataset. Further, the crucial issue of privacy leakage is not majorly addressed. Hence, the work aims to design an accurate and interpretable privacy-preserving classifier from the data streams concerning decision-making in banking sector.

### 4.3 Generic view of proposed approach

The generic view of proposed approach to accomplish the goal of privacy-preserving classification of horizontally partitioned data streams is shown in Figure 4.1. The approach comprises of three major phases: 1) Induction of local output-privacy preserving classifiers from data streams, 2) Induction of global classifier and 3) Optimization of global classifier. These phases are elaborated in the following:

#### **Phase 1: Induction of local output-privacy preserving classifiers from data streams**

In order to induce a global classifier from the horizontally partitioned data streams arriving at different participating parties, local privacy-preserving classifiers are generated from

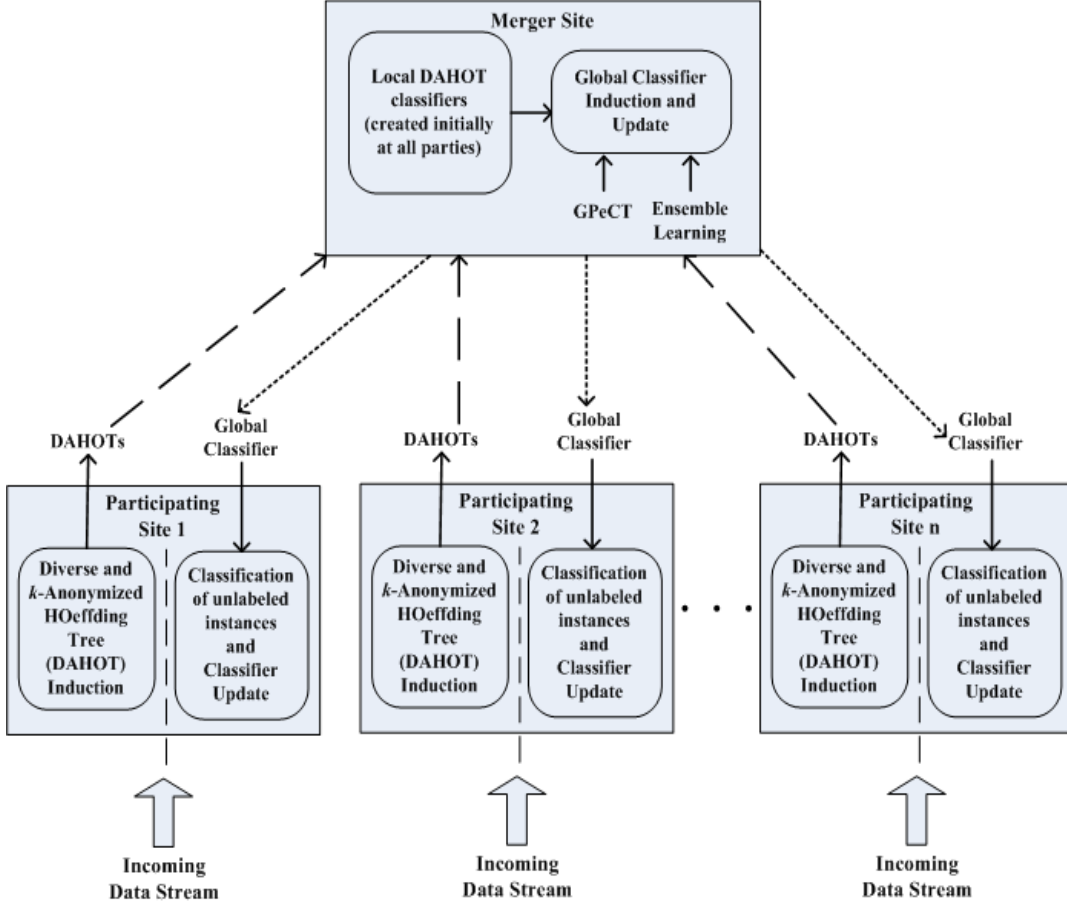


Figure 4.1: Generic view of proposed approach

data stream appearing at each party. These classifiers are then combined to form the global classifier.

In phase 1 of the proposed work, an approach named **Diverse and k-Anonymized Hoeffding Tree (DAHOT)** is proposed for preserving output-privacy in data stream classification at local sites. The approach uses Hoeffding tree algorithm to create a data stream classifier and  $k$ -anonymity as well as  $l$ -diversity principles to preserve the privacy of the output (i.e. the classifier). The goal is to output an anonymized version of this decision tree classifier and make it available for public usage. The proposed approach uses a ‘mine and anonymize’ strategy which first forms a decision tree classifier and whenever the output is to be published, it sanitizes the classifier. A detailed narration of the approach in Phase 1, experimental evaluation and analysis is presented in Chapter 5.

### Phase 2: Induction of global classifier

As shown in Figure 4.1, the output-privacy-preserving DAHOT classifiers generated in phase 1 are transferred to the merger site. At the merger site, a global ensemble classifier is constructed to address privacy-preserving classification of horizontally partitioned data streams which is then transferred to the participating sites.

The global classifier is induced using an approach named DAHOT-GPeCT. Within this approach, the DAHOT classifiers received from participating parties form the initial population and are evolved using Genetic Programming. GPeCT is named from **G**enetic **P**rogramming based **e**volution of **C**lassification **T**rees and includes novel features that add to the proficiency of GP.

Chapter 6 presents an in-depth analysis of the approach proposed in Phase 2.

### Phase 3: Optimization of global classifier

The global classifier induced in Phase 2 is further improved in Phase 3 using ensemble learning along with GP. Instead of using only the best-of-run obtained at the last generation of GP, the best few individuals from last run are combined to produce an ensemble classifier. Using buffering and windowing mechanism, this classifier is periodically updated.

The efficacy of this proposed approach known as DAHOT-GPeCT-Ensemble in privacy-preserving classification of horizontally partitioned data streams is analyzed and presented in detail in Chapter 7. The 3-phase induction approach proposed in Figure 4.1 accomplishes the work's objectives presented in Chapter 1.

## 4.4 Implementation environment details

Each of the algorithms proposed in all the phases is implemented in MATLAB 7.8.0 (R2009a) and the experiments are performed on a machine with Intel Core i3-350M 2.27GHz processor and 2 GB of RAM, all running on Windows 7 Platform. For Hoeffding tree induction,  $n_{min}$  is set to 200 and  $\delta$  is set to  $10^{-7}$ . These values of  $n_{min}$  and  $\delta$  are adopted from the default settings in original papers on Hoeffding tree induction

(Domingos and Hulten; Hulten, Spencer, and Domingos). Further, as mentioned in scope of the work, throughout the experiments, it is assumed that three parties want to collaboratively conduct the data classification. More parties can easily be accommodated but the results presented in this work are derived using three parties only.

The DAHOT classifiers generated at each participating site are to be periodically transferred to the merger site since the goal is privacy-preserving classification of ‘horizontally partitioned’ data streams. It is to be noted here that the processing of the algorithm used for generating a global classifier is not distributed. Only the classifiers are to be exchanged between the sites.

For the same, File Transfer Protocol (FTP) is used. An FTP server is created at the merger site. All the participating sites can transfer their DAHOT classifiers to the server and can also read from it. The merger site will read these classifiers (added at the server by participating sites) and induce the global classifier. The participating sites can also read the global classifier produced. The detailed process of classifier sharing is presented in next subsection.

### 4.4.1 Implementation of classifier sharing process

An FTP server has been created using FileZilla, which is an open source software (cross-platform FTP application), distributed without charges under the GNU General Public License. Figure 4.2 to Figure 4.7 indicate the process of file sharing between local sites and the merger site.

## 4.5 Summary

This chapter describes the application targeted throughout the work, relevant literature survey, outline of the proposed approach and overall implementation environment details. The subsequent three chapters portray the phases of the proposed approach presented in this chapter.

CHAPTER 4. PROPOSED FRAMEWORK

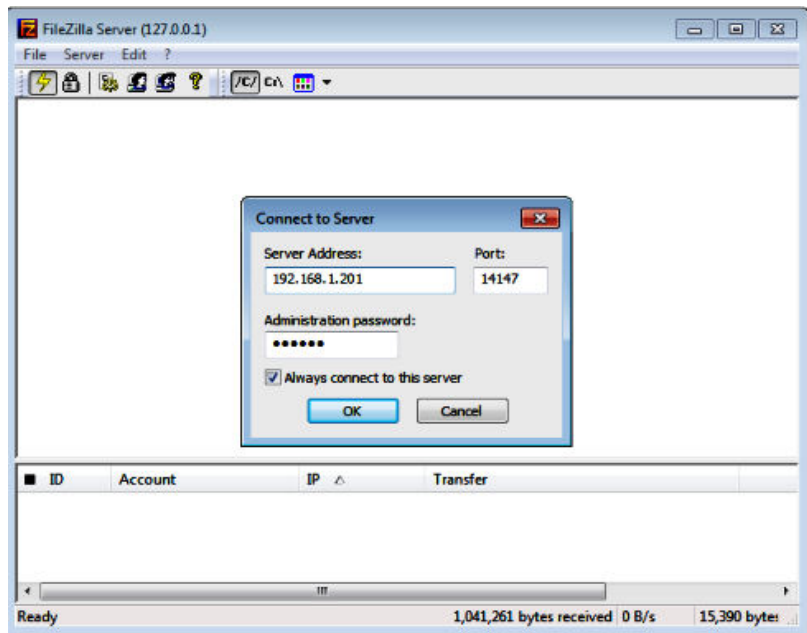


Figure 4.2: Creating and connecting to the server created at the merger site

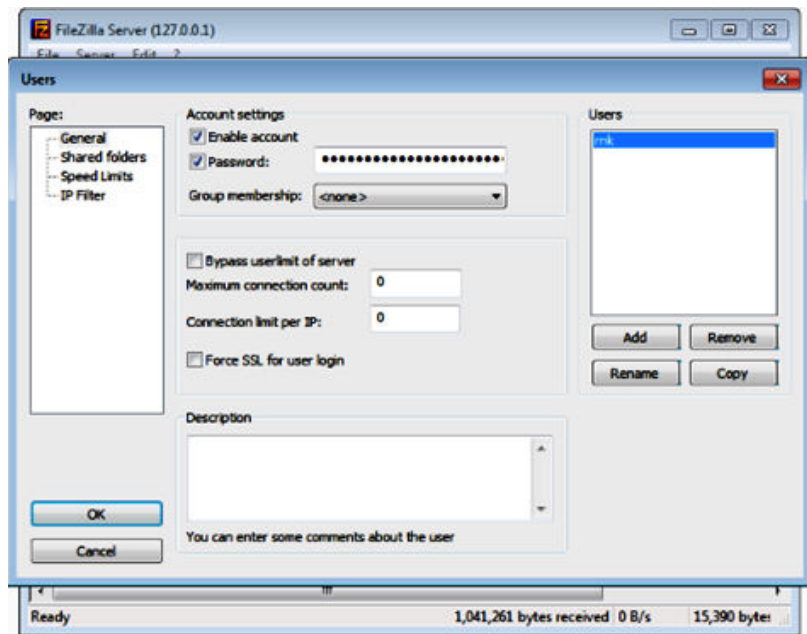


Figure 4.3: Creating and authenticating users (the local sites)

## CHAPTER 4. PROPOSED FRAMEWORK

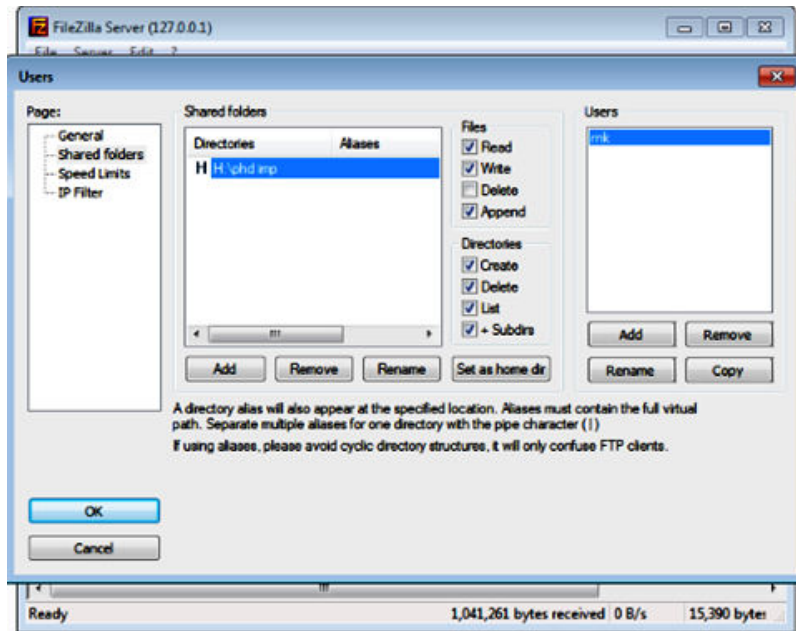


Figure 4.4: Authorizing the local sites for read/write permissions

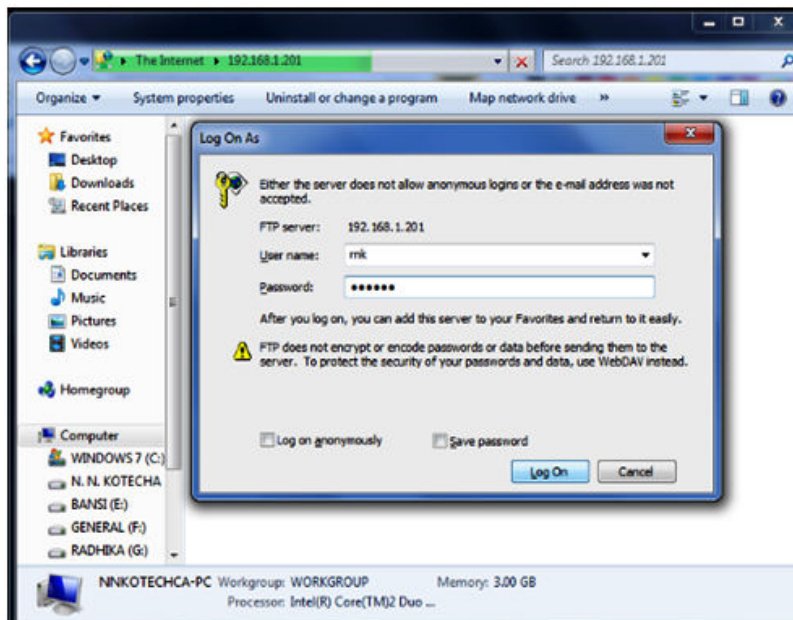


Figure 4.5: Login to the global site server by a local site

CHAPTER 4. PROPOSED FRAMEWORK

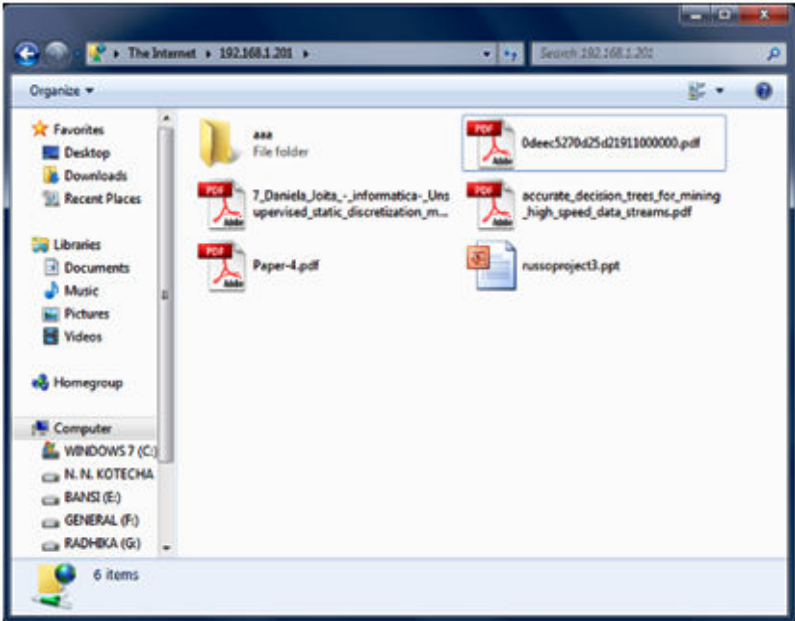


Figure 4.6: Adding/Reading classifier) at the merger site

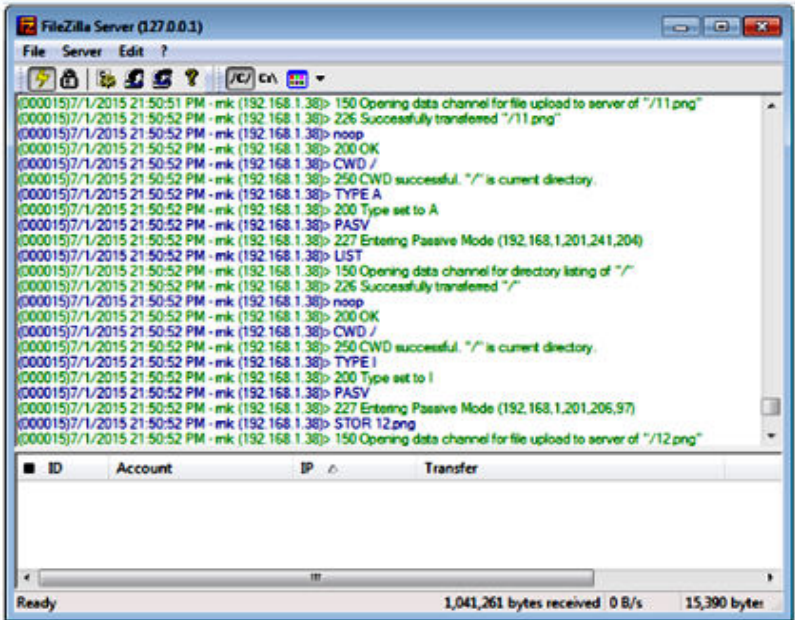


Figure 4.7: Access Log maintenance at the server





# Chapter 5

## Preserving Output-Privacy in Data Stream Classification

### 5.1 Introduction

Due to their effectiveness in supporting decision-making processes and extracting patterns locked within large collections of data, data mining techniques have attracted considerable interest and attention of research communities. It is apparent that the power of these techniques may breach the privacy of individuals to whom the data refers and the field of privacy-preserving data mining (Agarwal and Srikant; Lindell and Pinkas; Kantarcioglu; Zhang, Wang, and Zhao) has emerged in response to this issue. Specifically, the sensitive information in the original data should be protected from either direct or indirect (via linking and inference) exposure during the mining process. Further, not only the original data but also data mining output can lead to the disclosure of sensitive information.

Thus, the rapidly rising area of privacy-preserving data mining research has two main considerations: The first addresses sanitizing the sensitive raw input data prior mining it (Samarati; Fung, Wang, and Yu). The second focuses on sanitizing the data mining output to prevent inference of sensitive patterns from it (Wang and Liu). Most of the existing

---

Part of this chapter appears in: Radhika Kotecha and Sanjay Garg, “Preserving Output-Privacy in Data Stream Classification”, *Progress in Artificial Intelligence*, vol. 6, nos. 2, pp. 87-104, Springer (2017)

literature proposes methods only for the first consideration. But, since these methods attempt to preserve maximum utility and statistical information of the raw input data, the mining output may still breach privacy (Wang and Liu). Instead, it is suggested that by cleverly modifying the raw patterns obtained using data mining; the maximum utility of the patterns is preserved. Further, as per (Friedman, Wolff, and Schuster), a privacy-preserving classifier can be obtained efficiently as compared to a classifier obtained using sanitized data and the authors suggest first performing mining and then sanitization. If the data mining output reveals no private patterns, it can be reliably claimed that the privacy of underlying data is protected. Moreover, when the final goal is to release the output of data mining (a model); its effectiveness in preserving privacy is of the utmost concern. This work focuses on preserving output-privacy and proposes a method that prevents inference using the released classifier.

Further, the continuously arriving data streams provide great data mining opportunities but require serious consideration over the privacy implications in mining it. However, it has captured fairly limited attention thus far. Hence, the crucial issue of privacy-preserving data stream classification (PPDSC) is emerging as a novel research area. This chapter presents a more systematic method for preserving the privacy of data stream classifier output. The goal of this output-privacy-preserving data stream classifier is to prevent record linkage and attribute linkage attacks as described in Chapter 2. Henceforth in this work, privacy-preservation refers to preventing these attacks.

Chapter 2 presents an overview about the wealth of literature on classification of data streams and an empirical evaluation of some of these techniques is shown in Chapter 3. These studies suggest that Hoeffding Tree (Domingos and Hulten) is one of the most simple and efficient classification techniques for data streams. Specifically, when a classifier model is to be output, Hoeffding tree is suitable and interpretable. Further, several real world applications demand reasons for classifying any data instance into a particular class. Hence a symbolic classifier like Hoeffding tree is favorable and the proposed approach uses it as the base method for classifying data streams. Certainly Hoeffding tree classifier isn't capable to deal with concept-drift and the work in this chapter assumes no concept-drift

occurs in the data stream. The proposed approach is extended in Chapter 7 to handle the concept-drift.

Further, as shown in Chapter 2, the literature contains a large number of techniques that offer strong assurance to avoid information disclosure and preserve privacy of individuals. Among them, the work in this thesis focuses on  $k$ -anonymity as it covers the basic principle of privacy and is practical as compared to other other models (Ayala-Rivera et al.).  $k$ -anonymity has been studied intensively (Samarati; Fung, Wang, and Yu; Aggarwal et al.; Bayardo and Agrawal; Bertino et al.; Fung, Wang, and Yu; Iyengar; LeFevre, DeWitt, and Ramakrishnan), is conceptually simple and can be applied in most scenarios. Although few works (Machanavajjhala et al.; Sun et al.) have pointed out that  $k$ -anonymity is vulnerable to attribute linkage attack and have proposed variants of  $k$ -anonymity, Tian et al. argue that the newly developed models too use  $k$ -anonymity as a base. In addition to  $k$ -anonymity, the work also applies the  $l$ -diversity principle (Machanavajjhala et al.) to overcome the limitations of  $k$ -anonymity.

Since the eventual goal of any data mining application is to output a model or a pattern, the focus is on sanitizing the data mining output rather than conventional method of preserving privacy of data mining input. To preserve inference using the data mining output, the proposed approach first performs mining on the raw data and then imposes privacy protection on the mining results. Specifically, an algorithm named **D**iverse and  $k$ -**A**nonymized **H**oeffding **T**ree (DAHOT) is proposed that utilizes the Hoeffding tree algorithm as well as  $k$ -anonymity and  $l$ -diversity principles to produce a classification tree for preserving output-privacy in data stream classification.

## 5.2 Related work

This section presents, in brief, the existing literature related to anonymizing data that is dynamic: either the data is incrementally updated or is continuously arriving. Further, the work carried out to create privacy-preserving classifiers from the data streams is highlighted herein.

Xiao and Tao; Pei et al. and Fung et al. consider incremental privacy-preserving publishing scenario. The solutions proposed in the work are as follow: Consider  $R_1$  is the privacy-preserving release of  $D_1$ . For an incremental update with  $D_2$  instances, the proposed approaches generate new privacy-preserving release  $R_2$  of  $D_1 \cup D_2$  such that even if an attacker jointly analyzes  $D_1$  and  $D_2$ , there can be no privacy leakage. However, these incremental approaches cannot be applied for preserving the privacy of data streams because: 1) Existing approaches assume only one record per individual. But a data stream to be published may contain multiple records per individual. 2) Existing approaches scan the dataset multiple times which is not suitable for data streams with high velocity. Hence, approaches that consider preserving the privacy of data streams continuously are preferred.

Li, Ooi, and Wang proposed an algorithm called SKY (Stream K-anonYmity) that facilitates  $k$ -anonymity on data streams continuously while preserving maximum possible information in the anonymized data stream. The approach uses a pre-defined domain generalization hierarchy (DGH) to create a specialization tree (directed tree), where each node is a vector  $\langle v_l, \dots, v_m \rangle$ . Each value  $v_i$  in the vector is drawn from the DGH with the root of specialization tree being the most general node. For every new tuple in the data stream, the algorithm scans the specialization tree looking for the most specific generalization node containing that tuple. If the set of matching tuples of the node satisfy  $k$ -anonymity, then that tuple is anonymized using the node and is output instantly. Otherwise, the tuple is stored until the node satisfies  $k$ -anonymity. Moreover, a time delay constraint is applied on each tuple in the buffer. Once the time delay is exceeded, the tuple is anonymized by the node's parent and is output. Empirical evaluation in the paper shows the effectiveness of SKY algorithm.

Cao et al. proposed a cluster-based scheme called Continuously Anonymizing SStream-ing data via adaptive cLustEring (CASTLE) that performs  $k$ -anonymization on the incoming data streams dynamically. For  $n$  QIDs, CASTLE creates clusters as  $n$ -dimensional intervals. Initially, there are no clusters in the memory. When the first data instance arrives, CASTLE creates a cluster out of it. Progressively for each new instance, CASTLE

selects a cluster that needs the smallest enlargement to include that instance in it. CASTLE satisfies specified delay constraints by checking if an instance in any cluster is about to expire. Such an instance is output without delay, considering either of the two options: If the cluster containing the expiring instance is greater than or equal to  $k$  instances, all the tuples in that cluster are output with its generalization. Whereas, if the cluster with expiring instance has less than  $k$  instances, CASTLE selects a neighboring cluster for merging such that size of the consequent cluster is greater than or equal to  $k$  and the enlargement is minimum. Performance study in the paper shows that CASTLE efficiently anonymizes the quality of output data stream.

Zhou et al. describe that a smaller delay can improve the efficacy of published data and propose the following method to obtain  $k$ -anonymous data stream: The approach maintains a list of equivalence classes which is initially empty. For every new instance in the data stream, an equivalence class is created and the instance is added to it. If the next tuple is generated by a different object, it is added to the existing equivalence class; otherwise, the instance is inserted into a new equivalence class created by it. At any time, if the class has more than  $k$  instances in it, those  $k$  instances are published after generalizing and the published class is eliminated from the list of equivalence classes. Further, the paper proposes a randomized algorithm that considers information loss of equivalence classes. Maintaining statistics or creating an accurate privacy-preserving classifier out of the data streams is left as a future work.

Chao, Chen, and Sun propose an approach for privacy-preserving classification of data streams (PCDS). The PCDS approach works in two steps. In the first step, the private data streams are perturbed using data splitting and perturbation method. In the second step, the perturbed data streams are mined using the weighted average sliding window algorithm. The classifier is rebuilt when the classification error goes beyond a determined threshold. Although PCDS can accurately mine data streams, it requires multiple passes over the stream during the process: calculating variance of each attribute, perturbing the private attribute, classifying the perturbed stream, etc.

Chhinkaniwala and Garg also propose using perturbation for preserving privacy of data streams and further cluster the streams using standard clustering algorithms. They propose tuple value based multiplicative data perturbation approach: Tuple value of an instance is calculated as an average of normalized values of all attributes. These tuple values are then multiplied with value of instance's sensitive attribute resulting into a data stream with statistical characteristics of the data stream preserved. The pre-processing phase is followed by application of K-means clustering algorithm over a fixed sized sliding window. The efficiency of the proposed approach is demonstrated by performing experiments on two data streams with 65,000 and 45,000 instances respectively. But the approach considers preserving privacy of only numeric attributes and a method for nominal attributes is not addressed.

Chhinkaniwala, Patel, and Garg have also proposed an approach for privacy-preserving classification of data streams. A sliding window based data perturbation technique is used for preserving privacy of the data streams. The approach maintains a fixed size window containing the tuples and works as follows: For each sensitive attribute, the mean of values of all tuples in the window is calculated and the first tuple value is replaced by this mean value. The perturbed tuple is popped off and the next tuple in the data stream is added to the window. The process is repeated for every tuple in the data stream. The authors also use multiplicative data perturbation using rotation perturbation approach to preserve privacy of data streams. Hoeffding tree is induced from the perturbed data streams and experimental results have been performed on dataset with 45,000 instances. But, again the paper considers only numeric attributes to be sensitive and no solution for sensitive nominal attributes is proposed.

Xu et al. discuss the case of privacy-preserving classification of multiple data streams. As the streams are coming from different owners, joining all the streams would breach privacy. Also, joining the high-speed streams is not suitable. Hence, the authors adopt Naive Bayesian Classification (NBC) for this scenario and perform sliding window join (join on data stream in current window) wherein the necessary join statistics are computed without actually performing the join. The classifier obtained is similar to a NBC obtained

by applying join on the streams without disclosing private data.

Alongside the advantages, these methods have certain limitations as follows: If the attacker has knowledge of a sequence of instances in the data stream, both SKY and CASTLE might fail in adequately preserving the privacy. Further, creating a classifier using the approach proposed by Zhou et al. as well as the sliding window based approaches (Chao, Chen, and Sun; Chhinkaniwala and Garg; Chhinkaniwala, Patel, and Garg; Xu et al.) requires a pre-processing stage which consumes more time and memory. Also, these methods require reconstructing the classifier every time the window on the input data stream slides forwards.

Due to these limitations, the methods described in this section do not fit into the work’s goal of preserving output-privacy in data stream classification and hence these methods are not considered in the experimentation too. Instead, hybrids of the methods for data stream classification and privacy-preserving are used. The method proposed in this chapter overcomes these limitations: it neither requires frequent reconstruction of classifier nor it is susceptible to attack using a continuous sequence of instances. The next section describes this proposed method.

### 5.3 Proposed Output-Privacy-Preserving Data Stream Classifier

In this section, an algorithm named **D**iverse and **A**nonymized **H**oeffding **T**ree (DAHOT) is proposed for preserving output-privacy in data stream classification. The algorithm uses Hoeffding tree as a base classifier for classifying data streams and  $k$ -anonymity as well as  $l$ -diversity models to preserve the privacy of the output classifier.

The anonymized version of this decision tree classifier is to be output and made available for public usage. The proposed algorithm uses a ‘mine and anonymize’ strategy which first forms a decision tree classifier and whenever the output is to be published, it anonymizes the classifier. While constructing the decision tree classifier, the algorithm



inserts a new splitting node in the tree by choosing the best attribute, using the Hoeffding bound, and updates the tree accordingly. When the output classifier is to be published, the number of instances at the leaf nodes is calculated. The nodes that violate  $k$ -anonymity property are pruned.

The  $k$ -anonymous decision tree classifier proposed by Friedman, Wolff, and Schuster performs multiple passes over the data to find out the best splitting attribute, i.e., each instance is read multiple times. Also, the existing methods of privacy-preserving data stream classification perform multiple scans over the data stream in order to preserve the privacy. The proposed algorithm makes only two passes over the entire data stream: firstly to create the Hoeffding tree classifier and second to sort the examples to appropriate leaves during the anonymization process. For updating the classifier, only some of the instances are rescanned, which is explained in Section 5.3.1. Although the data stream mining techniques prefer scanning the instances only once, rescanning the previously-seen instances is permissible for time and space efficiency reasons (Domingos and Hulten), as follows: Firstly, rescanning is performed to enforce privacy constraints on the data stream classifier. In absence of rescanning, as in most of the existing works, PPDSC may require two stages: a privacy-preservation stage and a classifier induction stage. This extra stage of privacy-preservation will, in turn, require multiple passes over the data as well as consume more time. Secondly, devoid of rescanning, PPDSC will require the classifier to store a large amount of statistics (to ensure privacy constraints) and thus will need excess of memory (Kirkby).

Further, the  $k$ -anonymity model suffers from two major limitations (Friedman, Wolff, and Schuster) which have been identified and addressed by the algorithm. Firstly, it may be difficult for the data owner to tell apart which of the attributes will appear in external tables. The proposed approach isn't affected due to this limitation, since the scope of the work in this thesis assumes that class label (appearing at the leaves of the decision tree) is the only sensitive attribute. The assumption is acceptable since our task is classification and people may be aware of the basic details of an individual but are interested in the target concept (such as whether or not a client defaulted on a loan,

whether or not a person suffers from a particular disease, etc.). The second limitation is due to  $k$ -anonymity's susceptibility to homogeneity attack which is addressed by  $l$ -diversity model as described in Chapter 2. The work considers that leaves of decision trees provide the frequency of each of the classes rather than just the majority class. Diversity is enforced by altering the privacy constraint to one that requires a certain amount of diversity in class values at leaf nodes. That is, each leaf should have at least  $d$  tuples per class.

The outline of the proposed algorithm is presented in Algorithm 1, 2 and 3 accompanied by a detailed description of it.

### 5.3.1 Proposed algorithm

The proposed algorithm is shown in algorithm 1.

The algorithm begins by initializing the tree data structure DAHOT with a single root node at line 1. Each instance in the data streams (line 2) is filtered down to the appropriate leaf node  $l$  of the tree DAHOT based on the splitting nodes present in the tree (line 3). Also, counts  $n_l$ , the number of examples seen at leaf  $l$  and the counts  $n_{lpqr}$  at this leaf are updated at line 3 and 4 respectively. For discrete-valued attributes, the counts  $n_{lpqr}$  represent the number of examples of class  $p$  that reach the leaf  $l$ , where the attribute  $q$  has the value  $r$ . Continuous attributes need to be discretized, which is elaborated in section 5.3.2. Line 6 shows that selection of the best attribute to split is done only when a mix up of classes is observed at the leaf and when a minimum number of examples  $n_{min}$  has been accumulated. The latter is done in order to improve the efficiency of the algorithm.

Further, the attribute selection measure (say information gain)  $\overline{G}$  is calculated for each attribute  $A_i$  (line 6). If attribute  $A_1$  is found to be the best attribute (line 7) and  $A_2$  is found to be the second-best attribute (line 8); and the difference between their gains is greater than the Hoeffding bound  $\varepsilon$  (line 9 and 10), then  $X_1$  is selected as the splitting attribute (line 11). For each branch with the corresponding attribute-value pair (line

---

**Algorithm 1** DAHOTree( $D, n_{min}, \delta, k, d$ )
 

---

**Input:** Data stream,  $D$ ; the minimum number of examples before evaluating attributes,  $n_{min}$ ; 1 minus preferred probability of selecting the correct attribute at any node in tree,  $\delta$ ;  $k$ -anonymity parameter,  $k$ ; diversity parameter,  $d$

**Output:** Diverse and  $k$ -anonymized Hoeffding tree, DAHOT

1. Let DAHOT be a tree with a single leaf  $l$  (the root) and initialize  $n_{lpqr}$ , the number of examples of class  $p$  that reach the leaf  $l$ , where the attribute  $q$  has the value  $r$
  2. **for all** training examples in the data stream  $D$  **do**
  3.     Filter down the example into an appropriate leaf  $l$  of tree DAHOT and increment  $n_l$ , the number of examples seen at  $l$
  4.     Update the counts  $n_{lpqr}$  at  $l$
  5.     **if**  $n_l \bmod n_{min} = 0$  **and** examples seen at  $l$  belong to more than one class **then**
  6.         Compute  $\overline{G}(A_i)$  for each attribute of instances on  $l$
  7.         Let  $A_1$  be attribute with highest  $\overline{G}$
  8.         Let  $A_2$  be attribute with second-highest  $\overline{G}$
  9.         Calculate Hoeffding bound  $\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n_l}}$
  10.        **if**  $(\overline{G}(A_1) - \overline{G}(A_2)) > \varepsilon$  **then**
  11.            Convert  $l$  to an internal node that splits on  $A_1$
  12.            **for all** branches of the split **do**
  13.                Add a new leaf with initialized counts  $n_{lpqr}$
  14.            **end for**
  15.        **end if**
  16.     **end if**
  17.     **if** classifier output is to be published at any point **then**
  18.         **if** classifier output is to be published the first time **then**
  19.             Re-scan the entire data stream  $D$  seen so far
  20.         **else if** updated classifier output is to be published **then**
  21.             Scan the newly arrived instances in data stream and instances stored at active nodes
  22.         **end if**
  23.     Sort the examples into appropriate leaves of tree DAHOT
  24.      $T = \text{TreeTraversal}(\text{DAHOT})$
  25.     **for all** node  $v$  in  $T$  **do**
  26.         **if**  $v$  is a leaf node **then**
  27.             **PrivacyTest**(DAHOT,  $|D|$ ,  $v$ ,  $k$ ,  $d$ )
  28.         **else**
  29.             **if** both children of  $v$  are marked as *pass* **then**
  30.                 Mark node  $v$  as *pass*
  31.             **else**
  32.                 Prune node  $v$  to form a new leaf containing instances of its children
  33.                 **PrivacyTest**(DAHOT,  $|D|$ ,  $v$ ,  $k$ ,  $d$ )
  34.             **end if**
  35.         **end if**
  36.     **end for**
  37.     **end if**
  38. **end for**
-

---

**Algorithm 2** TreeTraversal(DAHOT)

---

**Input:** Binary tree, DAHOT

**Output:** DAHOT traversal sequence

1. Let  $Q$  be a queue and  $S$  be a stack
  2. Let  $n$  be the root node of DAHOT
  3. Enqueue  $n$  to  $Q$
  4. **while**  $Q$  is non-empty **do**
  5.   **if**  $n$  is a non-leaf node in DAHOT **then**
  6.     Enqueue left-child of node  $n$  to  $Q$
  7.     Enqueue right-child of node  $n$  to  $Q$
  8.   **end if**
  9.   Dequeue the front element of  $Q$  and Push it on  $S$
  10.   Designate the new front element in  $Q$  as  $n$
  11. **end while**
  12. **while**  $S$  is non-empty **do**
  13.   Pop the top element of  $S$  and add it to DAHOT traversal sequence  $T$
  14. **end while**
  15. return  $T$
- 

---

**Algorithm 3** PrivacyTest(DAHOT,  $N$ ,  $v$ ,  $k$ ,  $d$ )

---

**Input:** Binary tree, DAHOT; total number of examples seen so far in data stream,  $N$ ;  
node for testing against anonymity,  $v$ ;  $k$ -anonymity parameter,  $k$ ; diversity parameter,  
 $d$

**Output:** Anonymity-checked node,  $v$

1. Calculate  $n$ , the total number of examples at  $v$
  2. Calculate  $n_{lp}$ , the number of examples of class  $p$  at  $v$
  3.  $n_k = k * N$
  4.  $n_d = d * n$
  5. **if**  $n \geq n_k$  **and**  $n_{lp} \geq n_d$  for each class  $p$  **then**
  6.   Mark node  $v$  as *pass*
  7. **end if**
-

12), a new leaf node is created and the examples at the branching node are forwarded down the split to initialize the count  $n_{l_{pqr}}$  (line 13). This process, beginning from line 2, is executed in a loop for every training instance. As the attribute selection measure, we employ Gini Index used in the popular Classification and Regression Tree (CART) algorithm (Breiman et al.) that creates binary trees.

Data streams must be capable of producing the output at any given point (line 17-23). When the output classifier is to be published for the first time, all the instances in the stream are rescanned to compute the number of instances at the leaf nodes. This rescanning of all the instances in the data stream is carried out only once and the nodes that violate  $k$ -anonymity property are pruned. At this stage, the instances are divided and stored in parts corresponding to the leaves of the published classifier.

Whenever the data stream receives new labeled instances, the classifier is updated. Publishing this updated classifier requires re-scanning of: 1) instances that arrived after the last classifier was published 2) the instances at the nodes (called active nodes) that are expanded with the newly arrived instances. This rescanning occurs only when the user requires an updated version of the classifier to be published (which may occur on arrival of a large number of new instances in the data stream). This rescanning affects the performance of DAHOT only in terms of the time taken in reading and sorting the instances at appropriate leaves.

To produce an anonymous classifier, the classifier tree DAHOT is traversed as depicted in Algorithm 2 and the traversal sequence is stored in  $T$  (line 24). The methodology of traversing DAHOT described in algorithm 2 is explained in Figure 5.1. It is a combination of bottom-up traversal and breadth-first search technique. Figure 5.1(a) shows an example tree and Figure 5.1(b) shows its corresponding traversal sequence. The traversal is carried out such that any node is visited only after both of its children are visited.

Each node  $v$  in the traversal sequence  $T$  is tested for anonymity (line 25-36) as follows: If  $v$  is a leaf node, the anonymity test of Algorithm 3 counts the number of examples,  $n$ , appearing at  $v$ . If this count is less than the anonymity requirement  $k * N$  (where  $N$  is the number of instances seen so far in the data stream), the node  $v$  fails the  $k$ -anonymity

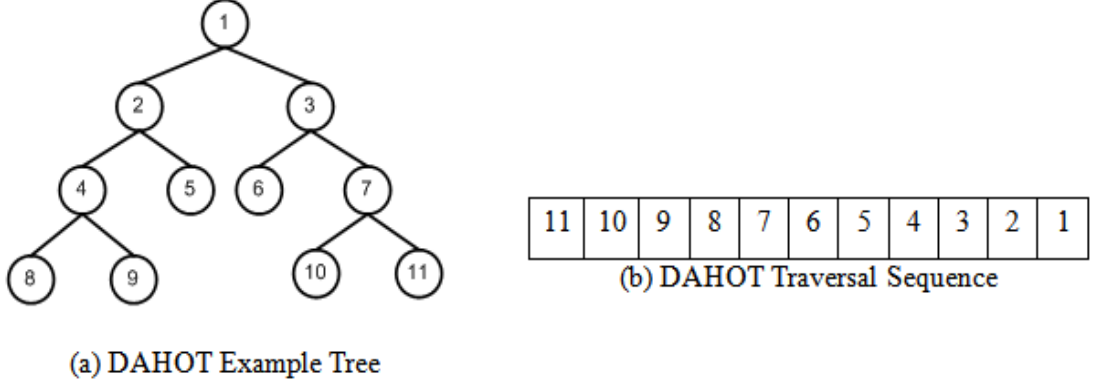


Figure 5.1: DAHOT traversal example

test. Otherwise, the node is resistant to linking attack as it makes each record belonging to its path (from root to leaf  $v$  of the tree) indistinguishable from at least  $(k - 1)$  other individuals. Further, to prevent homogeneity attack, the node  $v$  is checked against the class diversity parameter  $d$ . If  $n_{lp}$ , the number of examples per class is at least  $d$  times the total number of examples ( $n$ ) at  $v$ , then the node  $v$  passes the privacy test. The range of both the privacy parameters  $k$  and  $d$  is  $[0, 1)$ .

If  $v$  is a non-leaf node, it passes the privacy test only if both of its children (considering binary trees) pass the test. Otherwise, node  $v$  is pruned and converted to a leaf node, say  $v'$ . All the instances that belonged to the children of node  $v$  are now assigned to the newly created leaf  $v'$ . Further,  $v'$  being a leaf node, is again made to undergo the privacy test and marked *pass* if it satisfies the required constraints  $k$  and  $d$ .

Thus, a  $d$ -diverse and  $k$ -anonymous Hoeffding tree classifier is induced from the data streams. The tree obtained using this algorithm satisfies the need of a data stream mining algorithm: the capability to predict anytime. Also, it assures that the output does not reveal any sensitive information about the individuals referred in the data.

### 5.3.2 Discretization of data streams

Strategies for handling continuous attribute values have been extensively studied in the batch setting. The discretization methods transform the continuous data to discrete values

as a pre-processing step that is independent from the learning algorithm. Hence, any data mining technique that accepts only discrete attributes can mine data that originally contained continuous numeric attributes, by utilizing transformed version of the data.

To replicate the same in case of data streams is difficult as it is inefficient and at times impossible to load the entire stream into the memory and discretize it. The major reasons for this difficulty, is the large size of data streams as well as its nature of arriving continuously and not all together. Hence, the work considers a small but sufficient part of initial data stream (labeled) and uses it to find suitable discrete value substitutes for the continuous data. After this pre-processing step, any incoming data instance with numeric attribute is substituted to relevant discretized values and then used. The popular discretization methods for data streams are:

- a. Exhaustive Binary Tree based discretization (Gama, Rocha, and Medas)
- b. VFML method of discretization (Hulten and Domingos)
- c. Greenwald and Khanna's discretization method (Greenwald and Khanna)
- d. k-means clustering (Hartigan and Wong)
- e. Partition Incremental Discretization (PID) (Gama and Pinto)

The work utilizes a variant of PID to discretize the numeric attributes. The method is composed by two layers: The first layer divides the continuous numeric range into  $b$  bins of equal width. There is no overlapping between the bins, so that any given value will lie in exactly one bin. The second layer counts the number of instances in each bin and splits or merges bin as per the following criteria based on  $\text{count}_{avg}$  the average number of instances in each bin:

- If the number of instances in any bin is a threshold times less than count: the elements of the bin are merged with neighboring bins.

- If the number of instances in any bin is a threshold times greater than  $\text{count}_{avg}$ : the bin is split into two parts from the middle-point and elements are divided into respective bins.

The above procedure is repeated until either of the above two conditions exist. Finally, all other instances are sorted as per the derived discretization.

## 5.4 Experimental details

To demonstrate the effectiveness of the proposed algorithm, experiments are performed with different datasets. This section presents the details of datasets used as well as implementation details and reports the evaluation results.

### 5.4.1 Data streams

Since the scope of the work considers addressing the issue of privacy-preserving data stream classification in banking application described in the Chapter 4, experiments are performed on data streams in the banking sector. As mentioned in the scope of the work in Chapter 1, the class-label is considered as a SA and all other attributes are considered to be QIDs.

The banking datasets used in most of the existing work on privacy-preserving data mining have a small number of instances. This is mainly due to unavailability of real financial data from banks due to privacy constraints. But these days several synthetic data stream generators and large datasets depicting real-world problems are available. Four such data streams from banking sector are utilized in the experiments and their details are as follows:

- (1) Default of credit card clients: The dataset represents a case of defaulting on credit card payments by customers in Taiwan and has been taken from machine learning repository of the University of California at Irvine (Lichman). The data contains



23 attributes with 30,000 instances. The SA class-label has 2 values, depicting the client’s behavior to ‘default’ or ‘not default’.

- (2) Give me some credit: This dataset is provided by a financial institution for Kaggle competition (Kaggle) named “Give me some credit”. It contains 10 attributes and 150,000 instances divided into 2 classes representing whether somebody will experience financial distress in the next two years: ‘yes’ or ‘no’.
- (3) German Credit: The data stream is synthetically generated using the publicly available real-life data set provided by a German financial institution and available at machine learning repository of the University of California at Irvine (Lichman). This synthetic data stream contains 20 risk driver attributes with 300,000 instances. The sensitive class attribute has 2 values: ‘good payers’ and ‘bad payers’.
- (4) Loan Approval: This dataset is synthetically generated using Massive Online Analysis framework (Bifet et al.). The data generation function is designed to aid determining whether the loan should be approved. It has 9 attributes and 1,000,000 instances falling into 2 classes: ‘approve’ or ‘disapprove’ loan applications.

Although each of these data streams has only 2 classes, the proposed approach is applicable to multi-class data streams too. Since the data streams used for the application targeted in this work have only two classes, results on multi-class data streams are not shown.

### 5.4.2 Baseline methods for comparison

To demonstrate the effectiveness of the proposed algorithm, its performance is compared with 3 other methods. Details of these methods are described in the following:

- (1) **Sanitized Decision Tree Classifier on Partial Data Stream (SDTP)**: The method is based on sampling and has been adopted from approach by Friedman, Wolff, and Schuster; Aggarawal; Golab and Ozsú. It utilizes only partial (sample of) labeled instances to induce a classifier. From this partial data stream, the traditional decision

tree classifier using the algorithm CART is created and pruned until it satisfies the required anonymity and diversity constraints (i.e. pruned for sanitization).

- (2) **Ensemble of Sanitized Decision Tree Classifiers (ESDT)**: This technique is based on creating an ensemble of  $N$  classifiers using the Bagging approach (Breiman) and is an adaptation of method by Abdulsalam, Skillicorn, and Martin and Wang et al. Whenever a predefined number of labeled data instances are observed, a decision tree classifier (using CART) is created, sanitized (by pruning) and added to the ensemble. The number of instances to observe before inducing a base classifier is dependent on the type of application and preference of the user.
- (3) **Hoeffding Tree (HT) classifier**: Since the proposed algorithm is an enhancement of Hoeffding tree classifier (Domingos and Hulten; Kirkby), an efficient data stream classifier, it is used as a reference for comparison with the proposed method and analysis of the information loss and difference in accuracy due to sanitization.

### 5.4.3 Evaluation criteria

Considering the characteristics of an ideal privacy-preserving data stream classifier, the evaluation is done using the following parameters:

- (1) **Training and Predictive Accuracy**: Training accuracy is defined as the percentage of training instances that are correctly classified by the classifier whereas predictive accuracy is defined as the percentage of testing (previously unseen) instances that are correctly classified by the classifier.
- (2) **Training and Prediction Time**: Training time refers to the time taken to construct the model whereas prediction time is the time required for classifying previously unseen instances.
- (3) **Classifier Interpretability**: Interpretability refers to the level of understanding provided by the classifier. Since the work uses decision tree as a base classifier, inter-

pretability is considered based on number of nodes in the classifier. Large trees may not be interpretable and thus trees with smaller number of nodes are preferred.

- (4) Information Loss: The literature presents a variety of information loss metrics. In this work, the notion of information loss is based on the Classification Metric (CM) by Iyengar which is specifically applicable for anonymization of data concerned with the classification task. CM is computed based on an instance's adherence to the majority class of the classification tree node it is filtered to. It is defined in equation 5.1 as the difference between the sum of penalties of each instance in the data stream post and pre-sanitization, normalized by the total number of instances in the stream  $D$ :

$$Information\ Loss = \frac{(\sum_{instance\ r} penalty(r)_{post-sanitization} - \sum_{instance\ r} penalty(r)_{pre-sanitization})}{|D|} \quad (5.1)$$

An instance  $r$  is penalized if its class-label is not the majority class  $majority(N)$  of the node  $N$  it belongs to and is defined as in equation 5.2:

$$penalty(r) = \begin{cases} 1, & \text{if } class(r) \neq majority(N(r)) \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

#### 5.4.4 Evaluation methods

Two different evaluation methods are used in this work:

- (1) Periodic hold-out evaluation:

For performance assessment and comparison, a periodic hold-out method of evaluation is used where the classifiers are periodically tested on previously unseen instances.

This can be adjoined to a scenario of creating a classifier on a data stream whose initial instances are labeled and remaining stream consists of a mixture of labeled and unlabeled instances. Herein, whenever the stream encounters a labeled instance,

it is used for training the classifier; whereas when an unlabeled instance arrives, the classifier being trained is used to find the class label of that instance (that is, deployment of the classifier). While linking periodic hold-out with this scenario in the experimentation, instead of deployment, testing is conducted.

The proportion of training and testing instances during the entire stream is demonstrated in Figure 5.2 where the shaded area represents labeled, training instances. The commonly used approach in data mining is followed where 66% of the total data is used for training the classifier (Bifet et al.; Kohavi; Han, Kamber, and Pei).

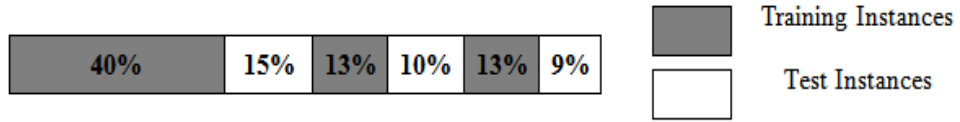


Figure 5.2: Scenario of interleaved training and test instances in data stream

The first testing is carried out after observing a large number of training instances. Hence, 60% of the training data (i.e. 40% of the total data stream) is devoted for the initial training part. The remaining 60% of the stream has mixed training (26%) and test (44%) instances. Experiments were also conducted using 50% and 70% of the training data for the initial part of the stream. The classification accuracy considerably drops on using only 50% of the training data initially whereas a little increase in accuracy is observed while using 70% of the training data for the same. The reported results use 60% of the training data before the first testing as it represents an average case scenario and seems to be more realistic as compared to the rest.

(2) Traditional hold-out evaluation:

Experiments have also been conducted using the traditional hold-out of method of evaluation as it is a standard method of evaluating data mining algorithms. The experiments use the initial 66% of the data stream for training the classifiers and rest for testing.

## 5.5 Results and Discussion

In this section, different discretization methods are evaluated and compared to identify a suitable one. Further, the experimental results of the proposed DAHOT with SDTP, ESDT, and HT are compared and analyzed on the data streams stated in Section 5.1.

### 5.5.1 Empirical comparison of discretization methods

The popular methods for data stream discretization have been empirically compared and the results of accuracy of Hoeffding classifier with attributes discretized using each of these techniques is depicted in Figure 5.3. The number of bins is set to 10 and the discretization threshold for PID is set to 50%. Since the purpose of this experiments is to identify a suitable discretization method for the DAHOT, experiments are conducted using DAHOT's base data stream classifier Hoeffding tree only.

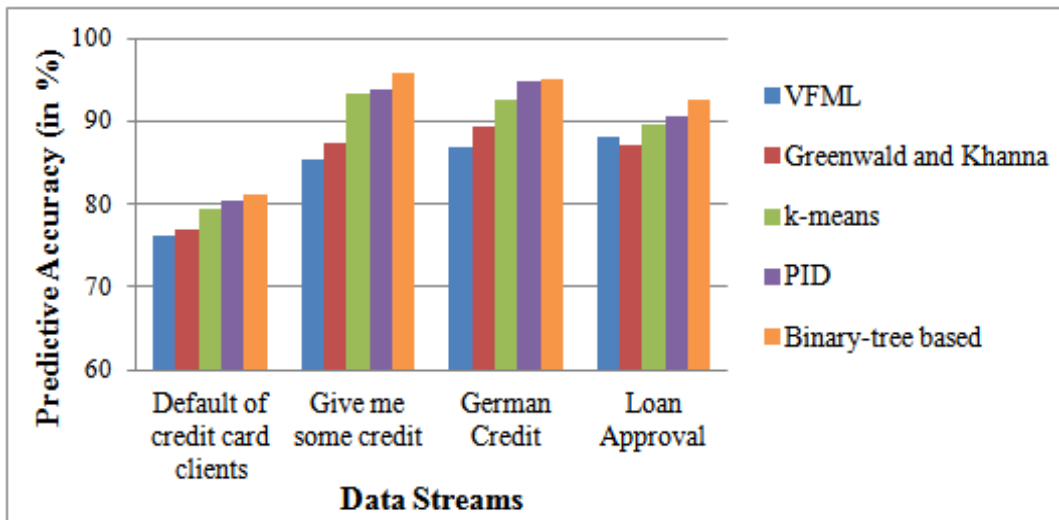


Figure 5.3: Predictive accuracy (in %) using different discretization methods

Although Binary-tree based discretization provides the highest result, the method ultimately stores all the values in the tree and hence occupying a large amount of space. Comparatively, PID gives good results and does not require excess storage post-discretization.

### 5.5.2 Evaluation using periodic hold-out method

This section discusses the experimental results of the periodic hold-out method of evaluation. SDTP utilizes only the initial (labeled) chunk of instances to induce a classifier. Any labeled instance arriving later in the stream is not used in creating or updating the classifier. For experiments using ESDT, an ensemble of  $N = 5$  classifier is formed with 3 decision tree classifiers are created from the initial 40% of data stream and 2 from the remaining labeled instances arriving later in the stream. Since the output of the algorithm is sensitive to the chosen values of privacy parameter  $k$  and  $d$ , a range of values for  $k$  and  $d$  are tested and its results are reported in this section.

#### 5.5.2.1 Effect of privacy parameter $k$

The amount of anonymity required, i.e. the anonymity parameter  $k$  is specified as the percentage of training data. For example,  $k = 0.2\%$  indicates that the number of instances at a node in the classification tree should be at least 0.2% of the training data. The effect of  $k$  in  $k$ -anonymity on all four data streams is tested and results of prediction accuracy by varying the value of  $k$  from 0.2 to 0.5 are reported in Figure 5.4 to Figure 5.7. For this set of experiments, the value of diversity parameter is set as  $d = 0.3\%$ . As the value of  $k$  increases, a greater amount of privacy is assured but more data instances are required to form an equivalence class at the leaf nodes. Also, a higher number of nodes may fail to be validated and hence be pruned. As a result, the predictive accuracy decreases, which can be observed from Figure 5.4 to Figure 5.7. Since HT represents a data stream classifier without privacy constraints, its performance is not affected by the anonymity parameter  $k$ . On the other hand, for all four data streams, a greater amount of reduction in predictive accuracy of STDP, ESDT, and DAHOT is seen for the value of  $k > 0.3\%$ . Hence, in all the remaining experiments conducted and reported in this paper, the value of  $k$  is constrained to be 0.3%.

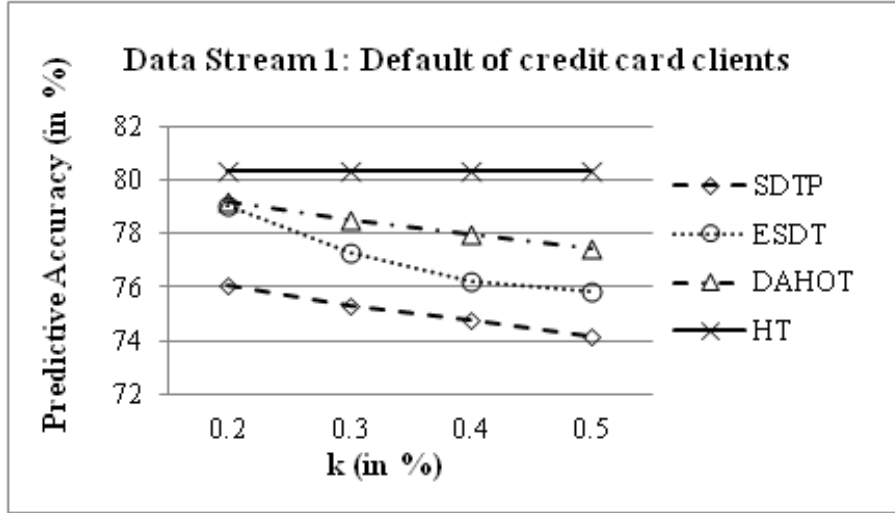


Figure 5.4: Predictive accuracy vs Privacy level ( $k$ ) on data stream 1

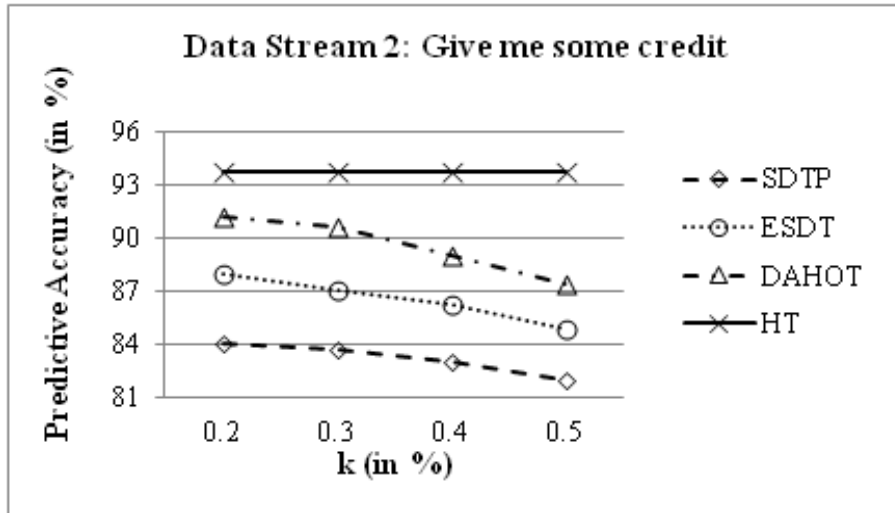


Figure 5.5: Predictive accuracy vs Privacy level ( $k$ ) on data stream 2

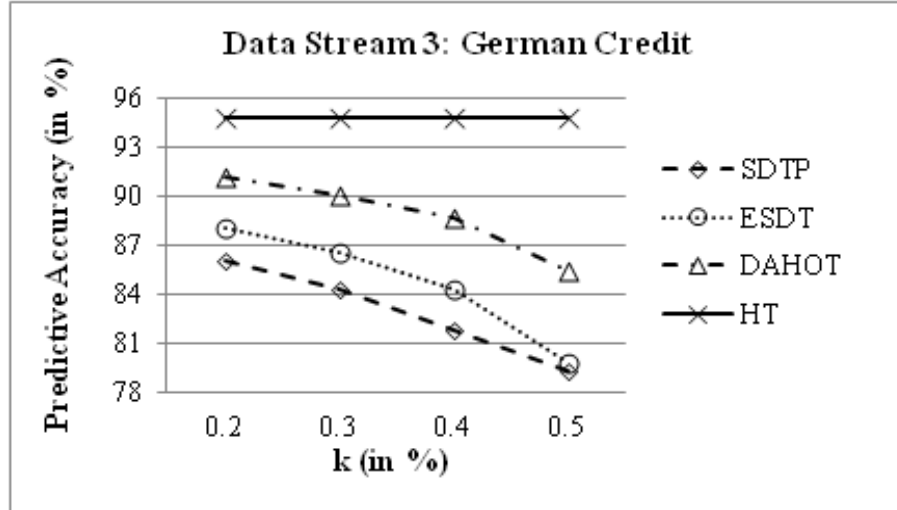


Figure 5.6: Predictive accuracy vs Privacy level ( $k$ ) on data stream 3

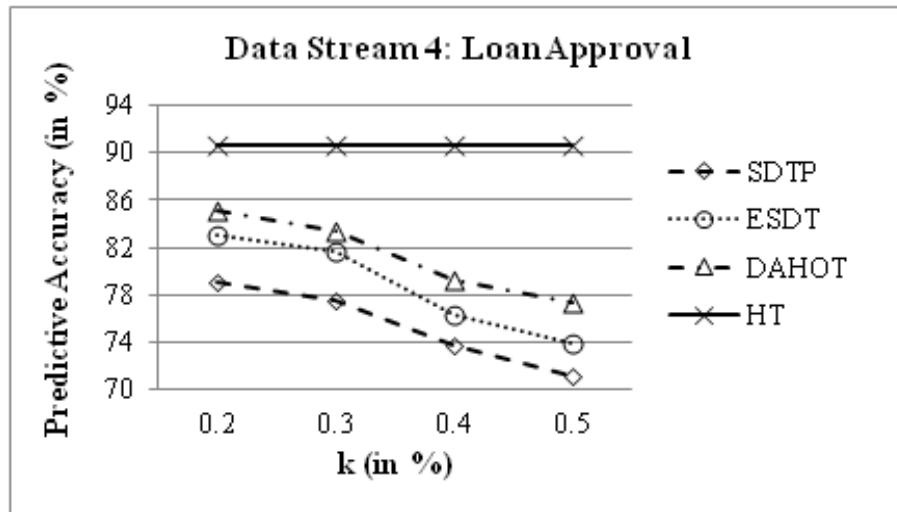


Figure 5.7: Predictive accuracy vs Privacy level ( $k$ ) on data stream 4



### 5.5.2.2 Effect of diversity parameter $d$

Further, the effect of diversity parameter  $d$  is examined by varying its value from 0.1% to 0.4%. The value of diversity parameter  $d = 0.1\%$  (for example) implies that at any leaf node, the proportion of instances of each class should be at least 0.1% of the total instances at that leaf node. The larger the value of  $d$ , the more privacy is preserved but the accuracy degrades. The results in Figure 5.8 to Figure 5.11 confirm the effect.

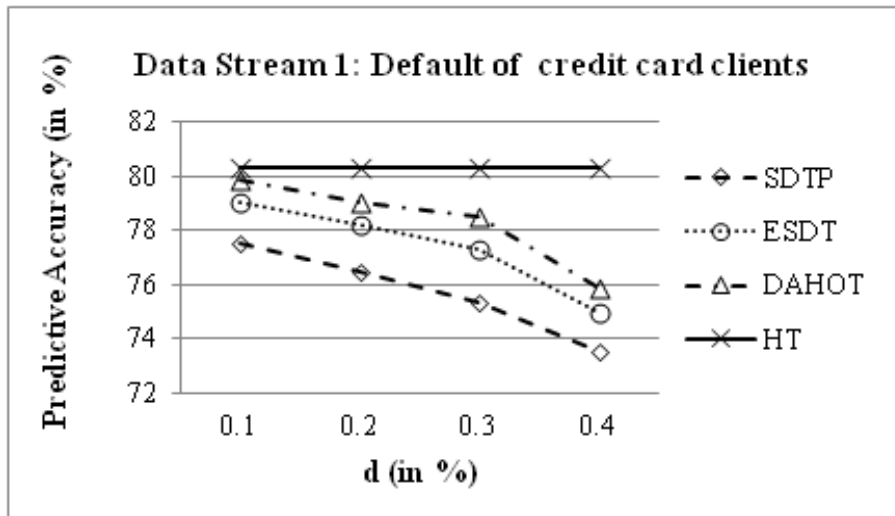


Figure 5.8: Predictive accuracy vs Diversity level ( $k$ ) on data stream 1

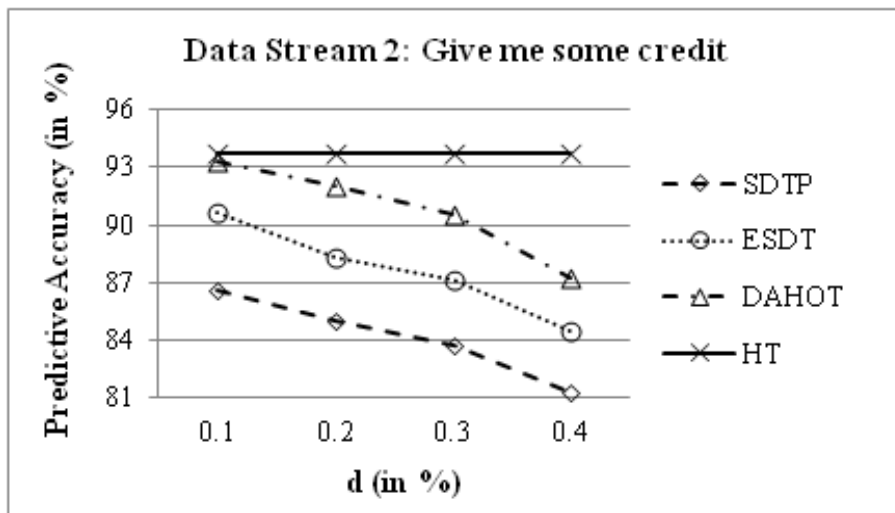


Figure 5.9: Predictive accuracy vs Diversity level ( $k$ ) on data stream 2

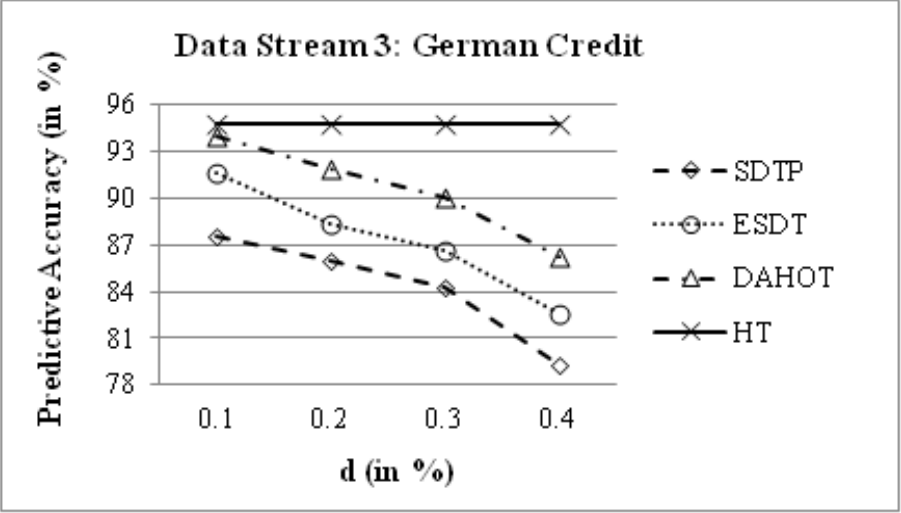


Figure 5.10: Predictive accuracy vs Diversity level ( $k$ ) on data stream 3

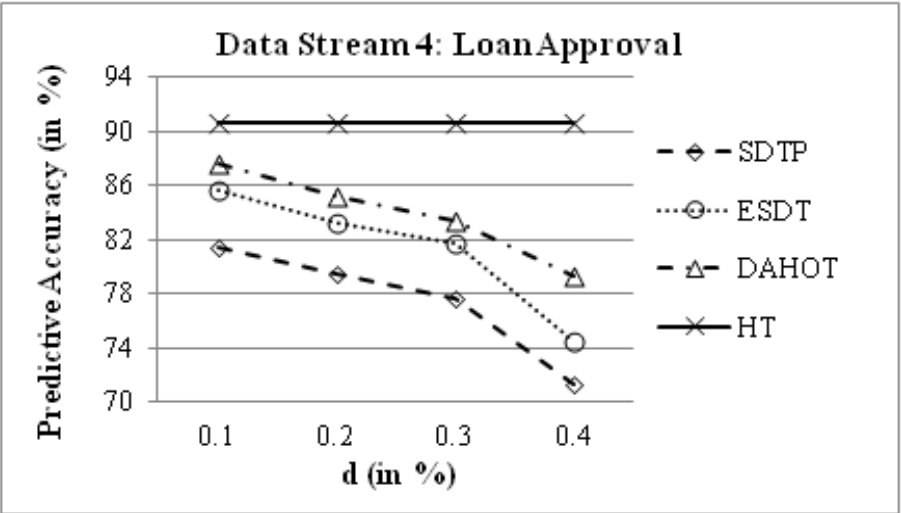


Figure 5.11: Predictive accuracy vs Diversity level ( $k$ ) on data stream 4

The results indicate that a tradeoff between accuracy and diversity is achieved by setting the value of diversity parameter  $d$  in between 0.2 to 0.3. In the rest of the experiments, by default, the value of  $d$  is set to 0.3%. A smaller value of  $d$  may make the classifier vulnerable to homogeneity attacks whereas for  $d > 0.3$ , the predictive accuracy falls considerably. Hence, the value  $d = 0.3\%$  that provides a reasonable performance is used.

### 5.5.2.3 Key Experimental Results

The training accuracy, predictive accuracy, training time, prediction time, information loss and interpretability of all four classifiers are evaluated on stated data streams and the results of the same are reported in Figure 5.12 to Figure 5.17 respectively. As shown

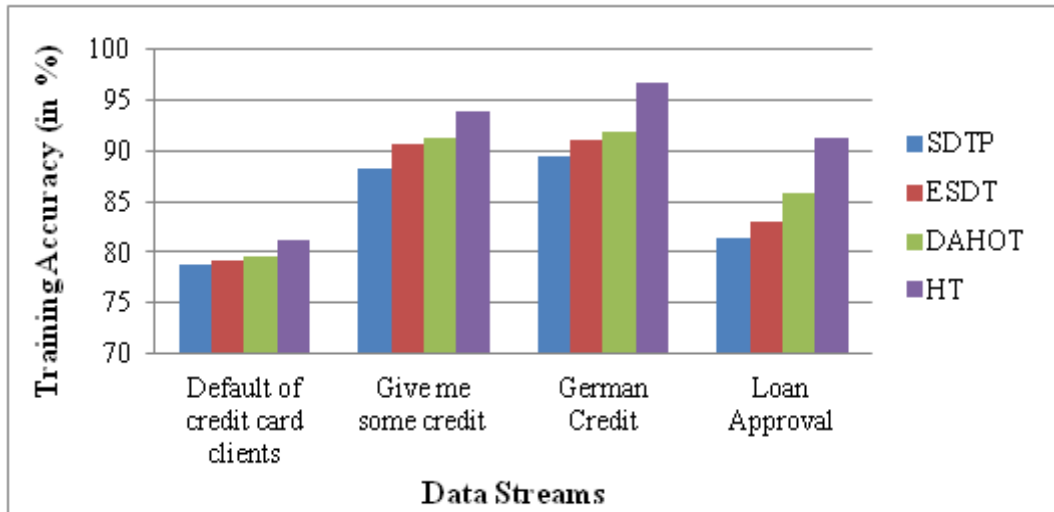


Figure 5.12: Training accuracy of classifiers (in %) using periodic hold-out evaluation

in Figure 5.12, the training accuracy of SDTP classifier is the low because the classifier is not updated with the new labeled instances that arrive after the classifier has been published once. Unlike SDTP, since ESDT does not ignore any of the labeled instances, its training accuracy is higher than STDP. Specifically, when enough instances are accumulated, ESDT simply induces a new decision tree from them and adds the tree to the ensemble classifier. ESDT waits until it receives at least 30% of the total number of

instances received earlier and then forms a new decision tree. ESDT is expected to be well accurate since it is based on the popular and successful ensemble classification technique. But, since each of the ensemble members is sanitized to achieve the required privacy, the overall accuracy of ESDT falls.

Further, like ensemble classifiers, HT has also gained popularity in recent literature and is a widely used technique for classifying data streams efficiently. Since DAHOT is based on HT, its training accuracy is higher than STDP and ESDT on all four data streams. However, a major reduction in accuracy of DAHOT is seen as compared to its ancestor HT as DAHOT is pruned to satisfy anonymity and diversity constraints. A small amount of pruning is expected to increase the accuracy of a classifier (in case of overfitting), but when several nodes are pruned (as in this case) a reduction in accuracy is seen.

Further, as shall be described towards the end of this sub-section, the size of DAHOT induced from ‘Default of credit card clients’ and ‘Give me some credit’ data streams is small. This implies that a larger number of instances belong to the leaves of classification tree and hence lesser nodes are to be pruned (to satisfy privacy constraints). As a result, there is a small difference in the training accuracy of DAHOT and HT on these data streams as compared to the remaining two as shown in Figure 5.12.

From Figure 5.13, it can be seen that the predictive accuracy of SDTP is again the lowest among all as it has not been trained with excess examples like the remaining three classifiers that are compared. However, when memory requirements do not permit all the training instances to be utilized for learning, SDTP can be considered as a choice. Further, an ensemble of SDTP classifiers can prove to be effective and is explored in the next chapter. As the sanitization process of the trained ensemble classifier removes some nodes and hence the learning by those nodes, the predictive accuracy of ESDT is lower.

Similarly, during the sanitization process of DAHOT, certain concepts are unlearned (tree pruning) and hence its predictive accuracy is lower than HT. As shown in Figure 5.13, since HT performs efficiently on all data streams and DAHOT is derived from HT, the predictive accuracy of DAHOT is higher than both SDTP and ESDT. SDTP

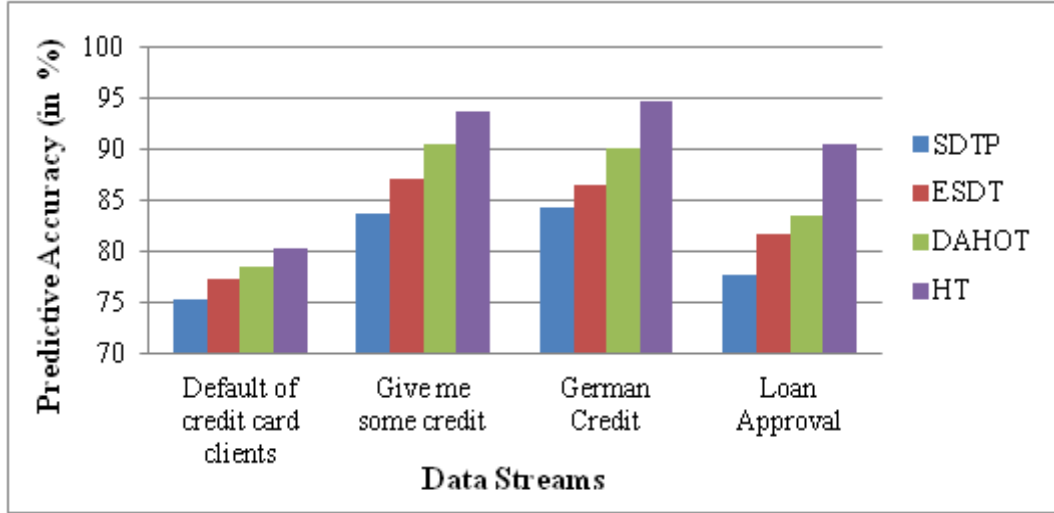


Figure 5.13: Predictive accuracy of classifiers (in %) using periodic hold-out evaluation

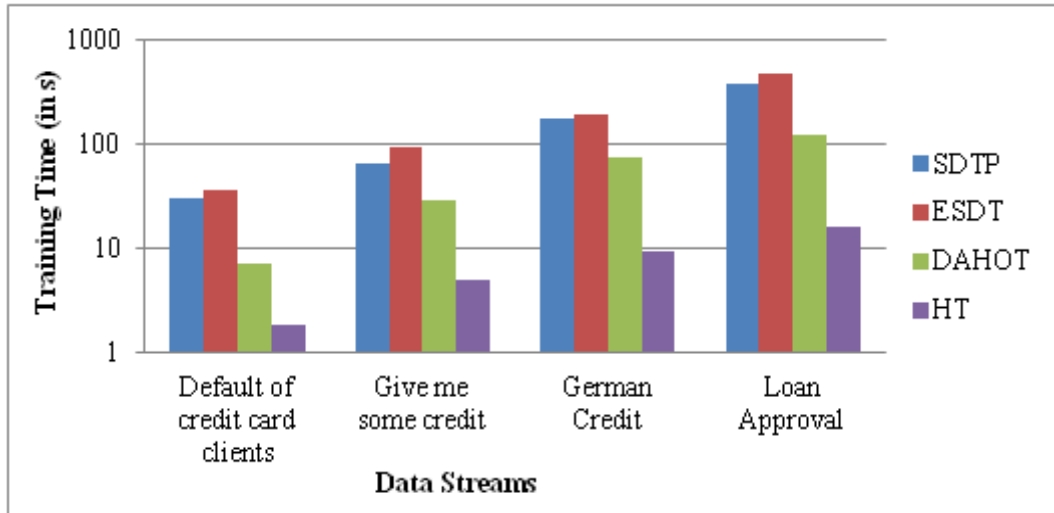


Figure 5.14: Training time of classifiers (in s) using periodic hold-out evaluation

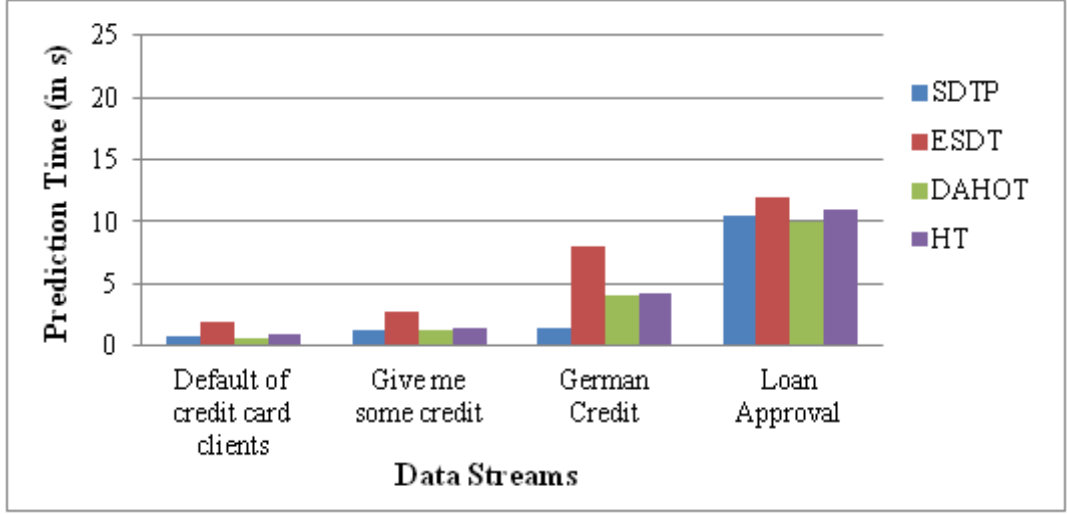


Figure 5.15: Prediction time of classifiers (in s) using periodic hold-out evaluation

and ESDT induce traditional decision tree classifiers and hence their training time is high and increases with the size of the data, which can be seen from Figure 5.14. Since HT is a classifier specifically designed for data streams, the time required for training HT is lower as compared to the traditional decision tree classifier. Since DAHOT is based on HT, training time of DAHOT is also less than SDTP and ESDT. But, along with HT-based tree induction, as DAHOT also performs the privacy-preservation process, its training time is higher than HT as shown in Figure 5.14.

Prediction time of classifiers is shown in Figure 5.15. Since SDTP is induced from lesser instances and size of the classifier is small, the prediction time is also less. ESDT requires maximum time in prediction since each testing instance needs to be evaluated against all the ensemble members prior declaring a class label. As the size of the HT classifiers is small (as shown in Figure 5.16 and described later), the time required in classifying new instances is proportionally small. Since DAHOT is a pruned version of HT, its size is smaller than HT. As a result, the time required to classify an instance using DAHOT is even less than HT, which is one of the advantages of our algorithm. It may be required to clarify that the privacy-preservation process in DAHOT is carried out during the training phase only and since the classifier published is already sanitized, no extra time for privacy-preservation is required during prediction.

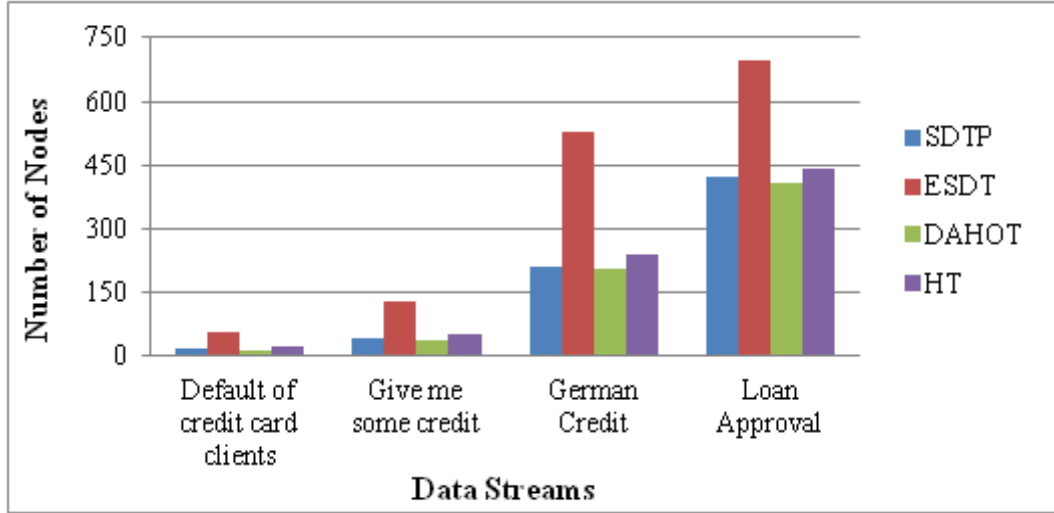


Figure 5.16: Number of nodes in classifiers using periodic hold-out evaluation

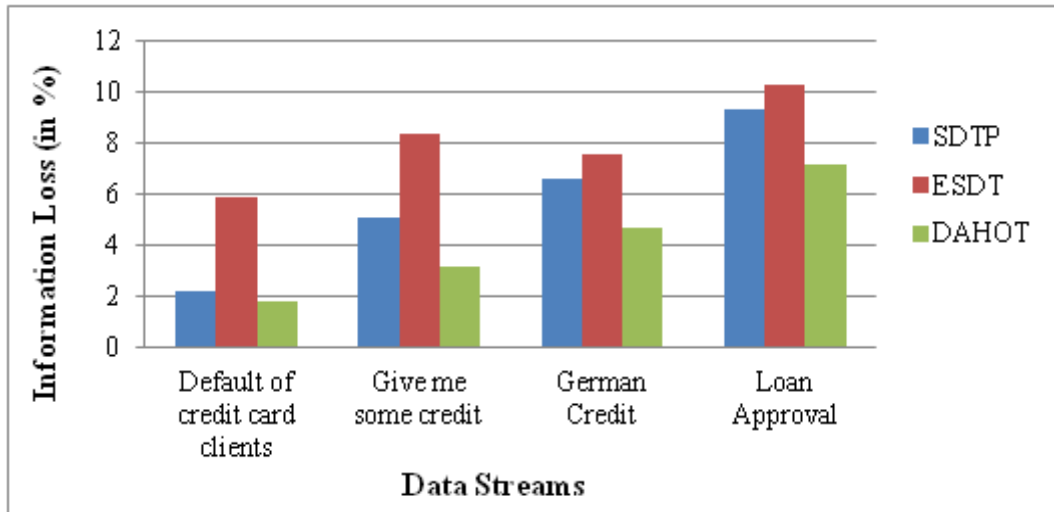


Figure 5.17: Information loss of classifiers (in %) using periodic hold-out evaluation

As mentioned in Section 5.4.3, interpretability is considered based on the number of nodes in the classifier which is depicted in Figure 5.16. Since SDTP induces a classifier from only partial data stream, the number of nodes in the classification tree is small, making SDTP very interpretable. On the other hand, ESDT does not ignore any of the training instances and ensembles 5 traditional decision tree classifiers induced from the entire data stream. As a result, the total number of nodes in classifier induced using ESDT is large making it less interpretable.

HT classifiers are smaller due to the Hoeffding bound as well as its characteristic of accumulating  $n_{min}$  instances prior splitting the nodes. As a result, HT and the proposed HT-based classification tree DAHOT has lesser number of nodes. Further, since DAHOT is a result of pruning HT for anonymity and diversity, the number of nodes in DAHOT is lesser than HT too.

Further, the information loss incurred by the targeted classifiers is shown in Figure 5.17. As the size of the classifier induced using SDTP is small, the leaf nodes contain a large number of instances. Hence, only a small number of nodes are to be pruned to satisfy the privacy constraints. As a result, the information loss using SDTP is lower than ESDT. On the other hand, since each of the ensemble members of ESDT is pruned till privacy is preserved, the resultant number of nodes pruned is large and hence the information loss of ESDT is the highest. Like SDTP, the size of classifier induced using the proposed DAHOT is small and unlike ESDT all the instances are sorted to a single tree of DAHOT. Also, since DAHOT is based on HT, it is accurate enough as it uses all the training instances for classifier induction. Thus, DAHOT requires a minimum amount of pruning which leads to minimum information loss as compared to all other classifiers.

In a nutshell, from the results of Figure 5.12 to Figure 5.17, it can be verified that DAHOT is an efficient and effective technique for preserving the privacy of data stream classification output.



### 5.5.3 Evaluation using traditional hold-out method

In this section, the results of traditional hold-out method of evaluation are demonstrated for the same algorithms and data streams used in Section 5.5.2. The periodic hold-out evaluation method and the traditional hold-out evaluation method produce nearly similar results but since hold-out method is considered as a standard method of evaluating data mining algorithms, experiments have been conducted using the former also and its results are presented here. The results of anonymity and diversity parameter tuning are not reported as they yield same conclusions as in periodic hold-out evaluation. Hence, again  $k = 0.3\%$  and  $d = 0.3\%$  is used throughout the experiments. Furthermore, again as per the commonly used approach of data mining Han, Kamber, and Pei, 66% of data is used for training and remaining is used for testing.

Figure 5.18 to Figure 5.23 demonstrate the results of training accuracy, predictive accuracy, training time, prediction time, interpretability, and information loss of SDTP, ESDT, DAHOT and HT using traditional hold-out evaluation on all data streams stated in Section 5.4.1.

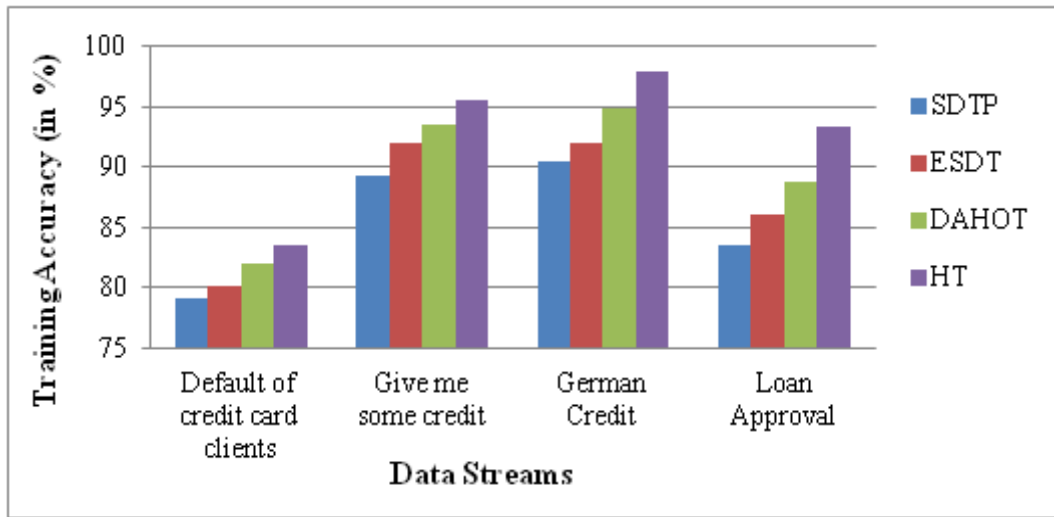


Figure 5.18: Training accuracy of classifiers (in %) using traditional hold-out evaluation

The training and predictive accuracies of classifiers are shown in Figure 5.18 and Figure 5.19 respectively. Similar to the results of periodic hold-out method, DAHOT

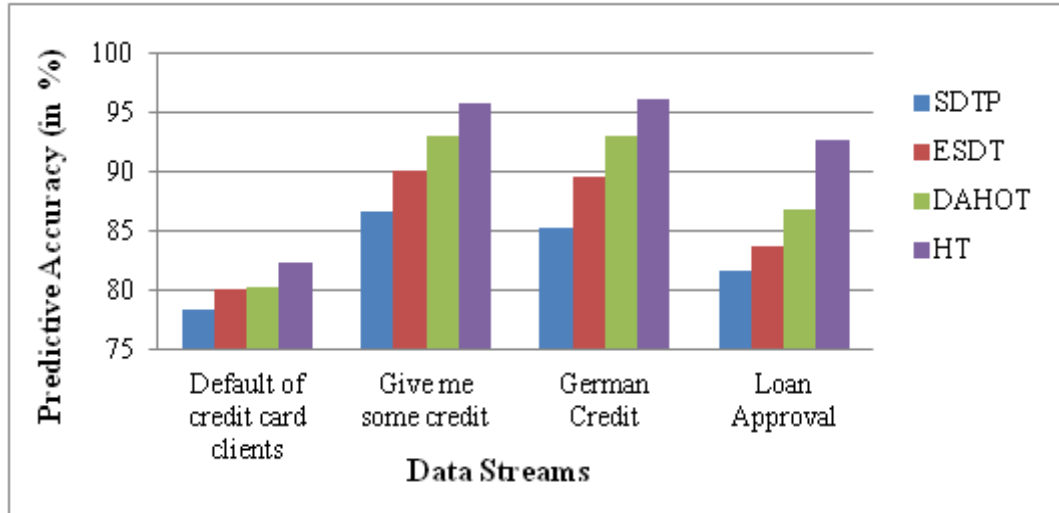


Figure 5.19: Predictive accuracy of classifiers (in %) using traditional hold-out evaluation

performs better than SDTP and ESDT and there is a small reduction in its accuracy as compared to HT due to the privacy-preservation process carried out by DAHOT. Since in the traditional hold-out method of evaluation, all the data stream instances are present beforehand, STDP has a wide range of instances to sample from and instances are sampled randomly without replacement. As a result, its predictive accuracy increases as compared to periodic hold-out evaluation method presented in Section 5.5.1.

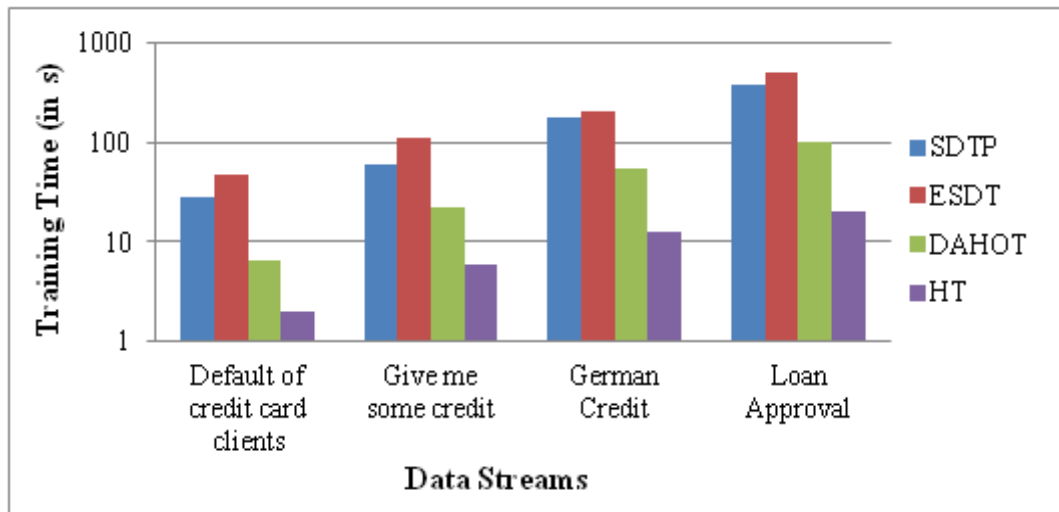


Figure 5.20: Training time of classifiers (in s) using traditional hold-out evaluation

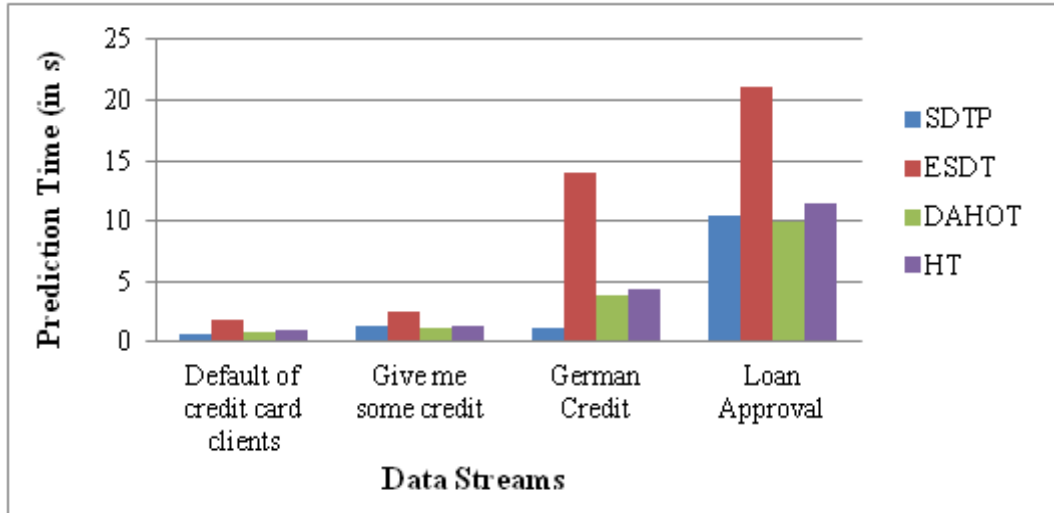


Figure 5.21: Prediction time of classifiers (in s) using traditional hold-out evaluation

Figure 5.20 and Figure 5.21 show the training and prediction time required by the classifiers which are almost equivalent to their versions induced using periodic hold-out evaluation. ESDT again consumes maximum time for training as well as prediction whereas the prediction time of DAHOT is the lowest among the rest as it is a pruned version of the small sized HT.

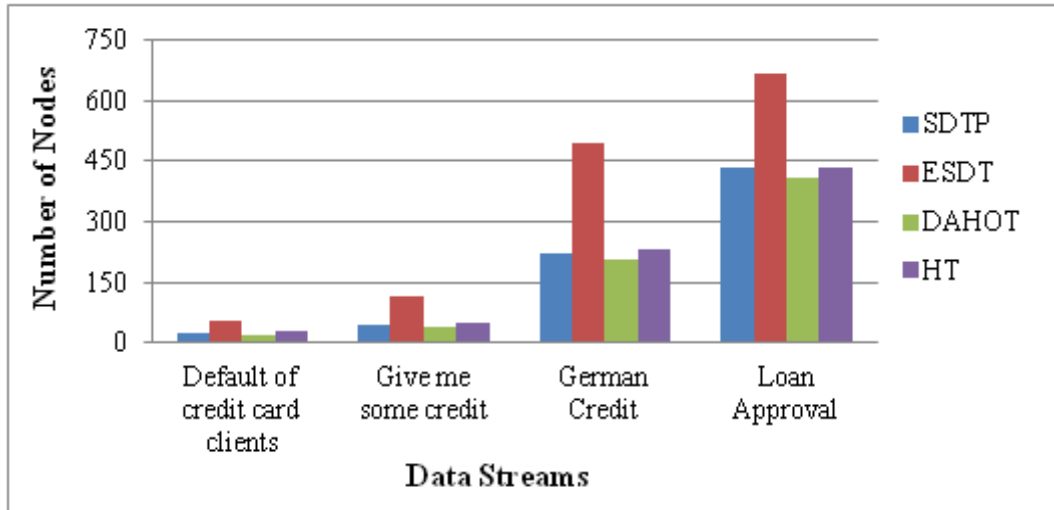


Figure 5.22: Number of nodes in classifiers using traditional hold-out evaluation

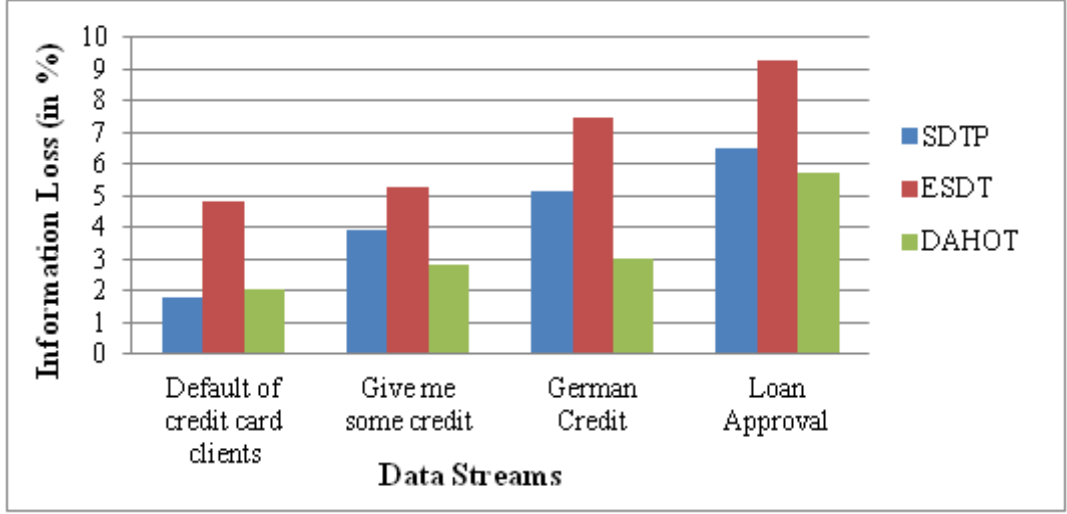


Figure 5.23: Information loss of classifiers (in %) using traditional hold-out evaluation

The observation for classifier interpretability (Figure 5.22) is same as that derived using evaluation of periodic hold-out method. Further, since the predictive accuracies of classifiers when they are supplied instances using traditional hold-out method are high, the information loss is little less as compared to the respective periodic hold-out version. The small information loss using each of the classifiers as shown in Figure 5.23 is acceptable due to the privacy-preservation guaranteed.

From the results of Figure 5.18 to Figure 5.23, the efficacy of DAHOT in preserving the privacy of data stream classification output is reassured.

## 5.6 Summary

Based on all the experiments conducted, it is observed that the major advantages of the proposed DAHOT, which is an enhancement of HT, include high predictive accuracy, less prediction time and lower information loss. The increase in predictive accuracy is only over SDTP and ESDT but not on HT since it is not privacy compliant. On using DAHOT, the minimum improvement in training accuracy is 0.76% whereas the maximum improvement is 5.27%. The minimum and maximum improvement in predictive accuracy (using DAHOT) is 1.96% and 5.82% respectively. Also, DAHOT preserves a minimum

of 2.47% and a maximum of 5.24% information as compared to SDTP and ESDT. Also, the advantages of SDTP indicate that it can be extended and enhanced to be utilized as an effective privacy-preserving data stream classifier. Techniques like ensemble learning, genetic programming, etc. have been applied in the next chapter to enhance SDTP. These techniques have also been applied to DAHOT to build an even improved classifier. Specifically, the next chapter explores these hybrids for an environment where the data is distributed between multiple parties.

# Chapter 6

## Genetic Programming-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams

### 6.1 Introduction

Data mining has emerged as a powerful tool for extracting patterns from the data streams arriving at various organizations. In today's technological era, success in a business isn't achievable single-handedly. Rather, success is based upon collaboration between parties because of the mutual advantage it brings. That is, multiple data streams need to be aggregated and mined to learn global patterns existing in the market and obtain accurate results for effective business decision-making.

An easy way is to gather the data streams from all the participating parties into a central site and run a data mining algorithm at this central site. But, this centralized

---

Part of this chapter appears in: Radhika Kotecha and Sanjay Garg, "Genetic Programming based Evolution of Classification Trees for Decision Support in Banking Sector", International Journal of Knowledge Engineering and Soft Data Paradigms, vol. 5, nos. 3/4, pp. 186-204, Inderscience Publishers (2016)

## *CHAPTER 6. GENETIC PROGRAMMING-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS*

approach is infeasible for a large number of applications due to several factors. One of major factors is that transmitting such large data streams to a central site is cumbersome and inefficient. Secondly, in today's malicious environment, it is difficult to assume the commonly used stratagem of the existence of a trusted third party for conducting the overall mining process. Specifically, collaboration may occur between competitors who are unwilling to share the data with each other or anyone due to privacy and confidentiality issues. However, these competitors are aware that the mutually beneficial collaboration is extremely important for business escalation. This issue has led to the concept of distributed data streams mining (DDSM) which involves mining the patterns from data streams that are distributed between multiple sites. DDSM addresses this issue without requiring the sharing of data and without revealing any information except the final data mining output. But as described in Chapter 5 of this thesis, since the data mining output can also breach the privacy, DDSM cannot be accomplished without application of privacy-preserving techniques. Privacy-preserving distributed data stream mining has emerged to address this issue.

As stated in Chapter 2, a vast amount of literature presents different methods to perform privacy-preserving data mining without using a trusted third party. These methods incorporate cryptographic techniques such as secure multi-party computation. However, due to high computation complexity of such methods, they become impractical when the input is large. Especially for the case of potentially infinite data streams targeted in this work, cryptographic techniques appear deficient to provide the required solution.

The goal of this chapter is to develop methods for output-privacy-preserving distributed data stream mining. Particularly, the focus is on output-privacy-preserving distributed data stream classification since the task of classification is the core subject of this thesis.

Organizations may collect the same set of data about different entities, i.e. all the parties have the same schema. Such a data model is referred as homogeneously distributed data or horizontally partitioned data. Numerous applications fall under this category. For example, customers' credit card transaction data streams where the features collected, like

income, age, gender, account balance, average monthly deposit, etc. are similar for all banks. Other examples include transaction information of clients of supermarkets, data streams arriving at hospitals, etc. Alternatively, the participating parties may gather different information about the same entities, i.e. all the parties have different attributes. This data model is referred as heterogeneously distributed data or vertically partitioned data. Examples in this category include trading transactions and phone call data, traffic report streams and road-accident streams, etc. As mentioned in the scope of the work, this thesis considers only horizontally partitioned data.

This chapter extends the work of Chapter 5 to privacy-preserving classification of horizontally partitioned data streams. Specifically, a genetic programming-based approach has been proposed that utilizes the output-privacy-preserving classifiers trained from data streams at individual sites to develop a global classifier at a central site. Since the classifier output released by individual sites does not disclose any private information, the global classifier formed using these local classifiers also prevents the privacy breach. The key dimensions considered while designing the global classifier are accuracy, privacy and interpretability.

## **6.2 Related work**

The participating sites send output-privacy-preserving decision tree classifiers to the central site. These classification trees accumulated at the central site are combined, evolved and optimized to form a global decision tree classifier. Decision trees (Mitchell 1997; Rokach and Maimon 2005) provide very accurate results and are symbolic (i.e. interpretable) classifiers. But, if the evolved global classification tree has very large number of nodes, it may not be interpretable. Thus, one should seek for the smallest and accurate classification tree.

Evolutionary algorithms (EA) (Back) like Genetic Algorithms (GA) (Goldberg) and Genetic Programming (GP) (Koza) have been found successful in solving numerous classification problems (Jabeen and Baig; Lee; Shali, Kangavari, and Bina). They are used



either directly as classifiers or to optimize the classification techniques (Oka and Zhao; Muni, Pal, and Das; Riekert, Malan, and Engelbrecht; Bot and Langdon; Davis et al.; Saraee and Sadjady; Khoshgoftaar, Liu, and Seliya). For example, GA is combined with kNN to improve the classification performance (Suguna and Thanushkodi). Kotecha, Ukani, and Garg have directly used GP as a classification technique. The results show that the classification tree obtained using GP has a smaller number of nodes. This is because GP allows controlling the size of the trees by depth-limiting the trees generated in order to form the initial population. Also, tree size can be used as a fitness measure while evolving the trees using GP and hence preferring trees with a smaller size. Hence, by combining decision trees and GP, a symbolic classifier that presents a good trade-off between accuracy and compactness can be produced. Further, tree structures are an encoding scheme for GP individuals (Espejo, Ventura, and Herrera), which makes merging GP and decision trees a preferable approach.

The remainder of this section presents in a nutshell, the existing literature related to the application of genetic programming to classification (and specifically decision trees), to show the different ways in which this evolutionary algorithm can help in the construction of accurate and reliable classifiers.

### **6.2.1 Application of genetic programming for classifier optimization**

The approach proposed by Enodu and Zhao evolves a data set rather than trees. The data set being evolved is targeted to be small in size while including maximum domain knowledge. The fitness of each evolving data set in the population is measured by creating a decision tree from it and verifying its generalization ability on a validation set. The approach works well (as compared to C4.5 (Han, Kamber, and Pei) on very small datasets and redundant datasets but fails on large datasets.

In the method proposed by Oka and Zhao, the initial population is created by randomly selecting parts of the training examples from the entire training set and applying C4.5

## *CHAPTER 6. GENETIC PROGRAMMING-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS*

on these examples to create Binary Decision Trees. The trees are further evolved using GP. The method does not show any improvement over C4.5. This may be because while randomly selecting training examples for creating the initial population, some significant knowledge may be missed. Also, the trees evolved are large as no size penalty is issued.

An approach that evolves a multi-tree classifier is proposed by Muni, Pal, and Das. For a classification problem with  $n$  classes, each individual in the population consists of  $n$  trees; one for each class. GP evolves by considering a joint view of each tree. Unfit trees are given more chances to evolve. The approach requires only a single GP run to evolve an optimal classifier for a multi-class problem. However, the size of the trees needs to be reduced, which is an open issue.

A new adaptive feature called culling is introduced by Riekert, Malan, and Engelbrecht. Culling adds new randomly created trees to the existing population after some generations of evolution. This feature adds diversity and hence provides little improvement over GP.

In the approach proposed by Saraee and Sadjady, values of attributes of the same type can be compared at the nodes of trees. For example, a test at the node of a decision tree can be  $\text{Attribute1} = \text{Attribute2}$ ? By comparing attribute values, the number of classification rules or the size of the tree can be reduced significantly because the number of comparisons is reduced. But, the approach fails when there are no dependent attributes.

A survey of existing literature on how GP can be applied to obtain efficient classifiers is conducted by Espejo, Ventura, and Herrera. The survey discusses how GP has been applied for feature selection, feature construction, extracting classification rules, learning ensemble classifiers, etc. Some important issues that can be the subject of future research in application of GP for classification are also addressed.

A method that constructs initial population using random decision trees is proposed by Rouwhorst and Engelbrecht. Also, mutations on relational operators in attribute tests, pruning, etc. are used as genetic operators. Finally, the final classification tree is converted into rules and the number of classification rules is decreased significantly as compared to C4.5 and CN2. But, no major increase in test accuracy is found. Two new genetic operators named elimination and merge are introduced by Kuo, Hong, and Chen.

Elimination operator eliminates redundant rules from the classification tree whereas the merge operator removes any subsumed rule from the classification tree. The usage of the new operators increases the accuracy of the classification tree while reducing the number of nodes.

Taking motivation and lessons from this study, the next section proposes a new algorithm that merges GP to Decision Trees and attempts to produce a near-optimal classifier in terms of accuracy and interpretability.

## 6.3 Proposed Approach

This section proposes an approach named **Genetic Programming based Evolution of Classification Trees (GPeCT)** that uses the output-privacy-preserving data stream DAHOT proposed in the previous chapter. This hybridized approach of DAHOT and GPeCT named DAHOT-GPeCT aims to evolve an optimized privacy-preserving classifier from horizontally partitioned data streams using Genetic Programming. The algorithm is run at the merger site and works on the output-privacy-preserving data stream classifiers (DAHOTs) received from participating sites. An optimal classifier needs to be accurate and comprehensible. Thus, the proposed approach gives importance to the size of the tree as well as its accuracy, by designing the fitness function accordingly.

### 6.3.1 Proposed algorithm

The proposed algorithm is shown in algorithm 4.

The input to the algorithm is a set of artificially generated data instances. Each participating party sends some output-privacy-preserving DAHOT classifiers to the global site to form the initial population. Further, two validation factors  $c_1$  and  $c_2$  that improve the efficiency of the algorithm are provided as an input. As an output, the algorithm gives a classification tree *best\_so\_far*, which is a near-optimal classifier in terms of accuracy and comprehensibility. Initially, *pop\_size* is set equal to the desired size of the initial population

---

**Algorithm 4** DAHOT-GPeCT. Evolves a classification tree.

---

**Input:**  $D_s$ , Data instances; DAHOT classifiers received from  $n$  participating parties; validation factors  $c_1, c_2$

**Output:**  $best\_so\_far$ , a near-optimal classification tree; Population of last generation

1. **for**  $i = 1$  to  $n$  **do** //Population initialization
2.   **for**  $i = 1$  to  $pop\_size/n$  **do**
3.     Add a DAHOT classifier  $C_{ij}$  to the population
4.   **end for**
5. **end for**
6. Evaluate fitness of each individual in initial population using fitness measure of Equation 6.1 and  $D_s$
7. **while**  $generation < max\_generations$  **or** Termination criterion is not satisfied **do**
8.   **repeat**
9.     Perform genetic operations on parent trees to generate offspring trees for the new population
10.    **if** (Size of individual  $< c_1 * \text{Size of } best\_so\_far$ ) **then**
11.     Evaluate its fitness using Equation 6.1 and increment  $new\_pop\_size$
12.    **else if** (Accuracy of individual  $> c_2 * \text{Accuracy of } best\_so\_far$ ) **then**
13.     Evaluate its fitness using Equation 6.1 and increment  $new\_pop\_size$
14.    **else**
15.     Discard the individual
16.    **end if**
17.   **until**  $pop\_size$  individuals are produced
18.    $generation \leftarrow generation + 1$
19.    $pop\_size \leftarrow new\_pop\_size$
20. **end while**
21. Save the population of last generation for classifier update
22. Designate best individual found so far as  $best\_so\_far$

---

and as the name suggests, the parameter  $max\_generations$  is set equal to the maximum number of generations the algorithm is desired to run.

The following sub-sections provide in-depth details of the algorithm and GP steps that are executed within the algorithm.

### 6.3.1.1 Initializing Genetic Programming Population

The algorithm begins by creating the initial GP population (of size  $pop\_size$ ) using DAHOT classifiers (decision trees). With  $n$  participating parties, each party  $i$  sends ( $pop\_size / n$ ) DAHOT classifiers. At each site  $i$ , a classifier is induced after a random instances

using random subset of attributes (Breimann). The classifier induction continues until  $(pop\_size / n)$  individuals are produced which are together sent to the global site.

As detailed in Chapter 5, DAHOT is composed of Hoeffding tree that is a decision tree classifier for data streams. Initializing the population using these decision trees would give us good trees (in terms of accuracy) from the beginning itself. This will give way to an advantage over the traditional GP way of initializing population randomly with terminals and functions (Oka and Zhao; Bot and Langdon; Rouwhorst and Engelbrecht; Kuo, Hong, and Chen).

As mentioned in section 6.2.1, Oka and Zhao use decision tree algorithm C4.5 to create an initial population, but not including every training tuple while creating the initial population results in loss of accuracy. In the proposed algorithm, the initial population produced using DAHOT considers all the data stream instances. Further, since multiple DAHOT trees are used from each site, instances are used multiple times for identifying patterns.

The Hoeffding trees (Domingos and Hulten) use Gini index (employed by the popular algorithm CART (Breiman et al.; Han, Kamber, and Pei)) as an attribute selection measure. With Gini index, binary trees are formed if the Hoeffding bound is satisfied. From the literature survey conducted, it is observed that binary trees are preferable for GP as most of the researchers have applied evolutionary algorithms on binary decision trees only.

The fitness of each individual (tree) is evaluated using the fitness measure as proposed in Equation 6.1:

$$Fitness = \frac{Accuracy\ of\ the\ Decision\ Tree}{[No.\ of\ Nodes\ in\ the\ Tree]^\lambda} \quad (6.1)$$

where the parameter  $\lambda$  represents a trade-off between accuracy and size (i.e. number of nodes) of the classification tree. Its value can range from 0.01 to 0.1.

The fitness function is developed such that a trade-off between accuracy and size of the tree can be provided. As the value of  $\lambda$  increases, the value of the denominator increases,

i.e. large trees are assigned lower fitness. The impact of trade-off factor  $\lambda$  on fitness is shown in Table 6.1.

Table 6.1: Impact of  $\lambda$  on fitness measure

Classifier No.	Accuracy	No. of Nodes	$\lambda$	Fitness
1	80	100	0.01	76.40
2	90	100	0.01	85.95
3	90	200	0.01	85.36
4	80	100	0.05	63.55
5	90	100	0.05	71.49
6	90	200	0.05	69.05
7	80	100	0.1	50.48
8	90	100	0.1	56.79
9	90	200	0.1	52.98

From Table 6.1, it can be seen that when two trees are of the same size: the tree with higher accuracy is considered fitter; when accuracies of two trees are same: the tree with a smaller size is considered fitter.

### 6.3.1.2 Genetic Operations

The algorithm performs the basic three genetic operations on the trees of existing population to generate the new population. They are: Reproduction, Crossover, and Mutation. The method of selecting individuals for genetic operation varies according to the nature of the operation.

For reproduction, the best individuals are selected using the Rank Selection Method. This is to make sure that the fittest members are more likely to be passed on to the next generation, and past optimal solutions are not lost. For crossover operation, tournament selection method is used to select individuals for this operation. For mutation, a method

called reverse rank selection is proposed to select individuals for mutation. That is, this method selects unfit individuals over the fit individuals and mutates them. Using this method saves good trees from undergoing the random changes of mutation, because as per (Muni, Pal, and Das), the nature of mutation is destructive at times. This new selection method has three benefits. One is that the good trees are not damaged, secondly, very bad trees are removed and replaced with better trees and lastly, diversity is also introduced.

### 6.3.1.3 Validation of Individuals

The work introduces a new process named “validation process” for the individuals produced through the genetic operations. For any GP individual  $i$ , let  $S_i$  denote the size of individual and  $A_i$  denote individual’s accuracy. Further,  $best\_so\_far$  indicates the best individual obtained by far.

For validation factors  $c_1, c_2 \in R^+$ , the validation, i.e. acceptance or rejection occurs as per Equation 6.2:

$$Validate_i = \begin{cases} 1, & S_i < c_1 \cdot S_{best\_so\_far} \\ 1, & A_i > c_2 \cdot A_{best\_so\_far} \\ 0, & otherwise \end{cases} \quad (6.2)$$

Within this process, the fitness of only validated individuals is calculated. If any individual  $i$  is rejected (i.e.  $Validate_i = 0$ ), its fitness is not calculated.

The individuals who pass the above test and get validated are added to the new population and the counter of *new\_pop\_size* is increased by one per individual validated. This parameter at the end of the generation gives us our new population size which is to be used for the next generation. This way, the algorithm uses variable-size population. Using variable-size population allows saving the time and computation effort required in calculating the fitness of already known bad individuals.

However, a generation is incremented only if *pop\_size* individuals are produced, irrespective to those individuals being accepted or rejected. Hence, to check if the generation should be incremented, the count of number of individuals produced is increased with ev-

ery genetic operation (in accordance with the number of individuals produced, i.e. count of individuals produced is increased by 1 for reproduction and mutation and by 2 for crossover). Once the required amount of individuals i.e. *pop\_size* individuals are produced, the next generation is evolved. For this new generation, *new\_pop\_size* becomes *pop\_size*.

#### 6.3.1.4 Termination Criterion

The algorithm stops when the termination criterion is met or the maximum number of generations (*max\_generations*) has been reached. Another termination criterion here is that for  $p$  consecutive generations, the best  $q$  fitness values ( $F$ ) achieved should be the same, which is depicted using Equation 6.3:

$$\forall i \in [1, p], \forall j \in [1, q] : F_{i,j} = F_{i+1,j} \quad (6.3)$$

Once the termination criterion is met or the maximum number of generations has evolved, the best classifier obtained by now, that is, *best\_so\_far* is designated as the output classifier.

Since the data streams are continuously arriving, formation of new classification trees at local sites may be required and the global classifier update may also be needed. The proposed algorithm has the provision to handle such classifier update. Here, the population of the last generation produced while evolving the classifier is saved. Later, when new training instances come at hand, trees induced from this data are sent to the merger site and added into the current population and GP is run for one more generation using the said population. The *best\_so\_far* obtained at the end of this run becomes the final classifier. This classifier would incorporate training instances available earlier as well as the newly arrived instances. This method of classifier update is inadequate, especially when there is a concept-drift in the data. A systematic method of classifier update is presented in Chapter 7.

A framework of the proposed approach is shown in Figure 6.1.



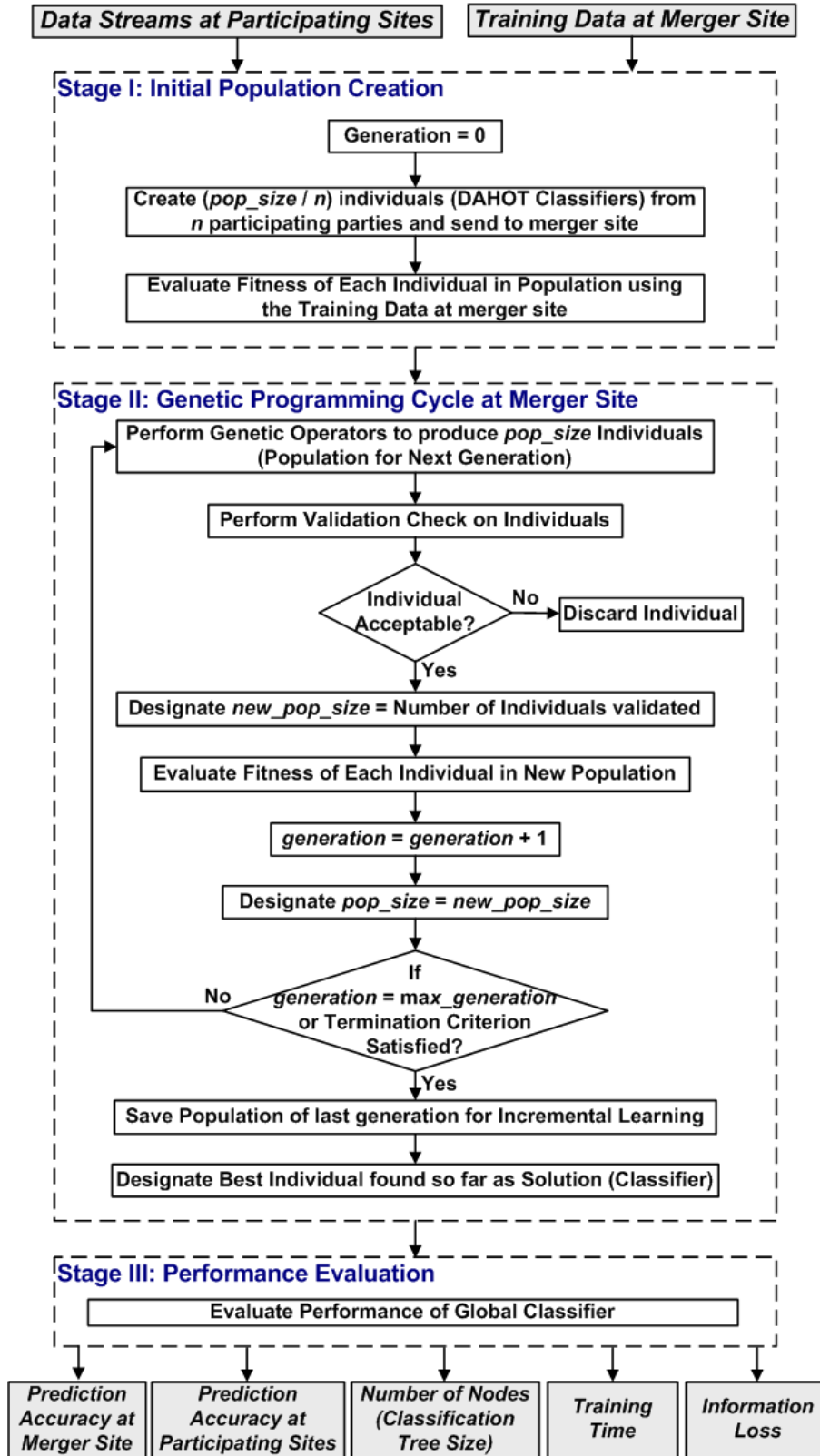


Figure 6.1: Framework of the proposed approach

The entire process is divided into 3 stages. **Stage I** includes creating initial population of DAHOT classifiers from the incoming data streams at participating sites and sending these local classifiers to the merger site. The training data at merger site is used to evaluate the fitness of these local classifiers. The Genetic Programming cycle for inducing the decision tree classifier is executed at the merger site in **stage III**. The global privacy-preserving classifier induced from horizontally partitioned data streams is evaluated in **stage III** and is analyzed for prediction accuracy (at the merger site as well as after sending them to the participating sites), classifier size, training time and information loss.

## 6.4 Experimental Evaluation and Analysis

In order to verify the effectiveness of the proposed algorithm and address the problem of privacy-preserving classification of horizontally partitioned data streams, several experiments are conducted with different real-world and synthetic data streams. The following sub-sections describe the details of the data streams used as well as implementation details and present the evaluation results.

### 6.4.1 Data streams at participating sites

Since decision making in banking sector is the target application throughout the work, the same data streams described in Chapter 5 are used in this chapter too. But, since the work in this chapter works on horizontally partitioned data streams, the composition of the data at each site is as shown in Table 6.2:

The number of instances in the data streams at each site ranges from thousands to a million, verifying the performance of the proposed algorithm on data streams with different sizes. The types of attributes in these datasets are numeric, categorical, or mixed. Some instances of data streams ‘Default of credit card clients’ and ‘Give me some credit’ have been replicated at the other sites. Further, as stated in Chapter 5, because of the lack of availability of large datasets for banking as well as for validation of the

Table 6.2: Composition of data streams

<b>Data stream</b>	<b>No. of Attributes</b>	<b>No. of Classes</b>	<b>Instances at Site 1</b>	<b>Instances at Site 2</b>	<b>Instances at Site 3</b>
<b>Default of credit card clients</b>	23	2	20,000	20,000	20,000
<b>Give me some credit</b>	10	2	100,000	100,000	100,000
<b>German credit</b>	20	2	300,000	300,000	300,000
<b>Loan approval</b>	9	2	1,000,000	1,000,000	1,000,000

proposed algorithm, synthetic data streams are used. Thus, large number of instances has been generated at each site for synthetic datasets ‘German credit’ and ‘Loan approval’. Furthermore, as described in Chapter 5, although each of these data streams has only 2 classes, the proposed approach is applicable to multi-class data streams too. Since the data streams used for the application targeted in the work have only two classes, results on multi-class data streams are not shown.

#### 6.4.2 Data streams at merger site

At the merger site, several DAHOT classifiers are received from the local participating sites. The proposed algorithm GPeCT is run on these DAHOT classifiers with a goal to induce an optimal global classifier. For the algorithm to run, some data instances are required as training data. The data instances are also required to predict the performance of the induced global classifier. In order to meet this demand of the algorithm, the participating sites need to send a sample of their data streams to the merger site. But, since the privacy of the data is of utmost importance, it is not acceptable for the local sites to disclose or send the original data in raw form. The original global data cannot exist physically but is required to exist, at least conceptually. To address this issue, the

framework proposes to produce artificial data at the merger site. The instances in the artificial data have the same attributes as in the data streams of all the local sites. Such artificial data is generated using two different ways as follows:

- (1) Artificial Data Generation through Anonymization: An approach to preserve the privacy of the micro-data is to transform it using anonymization methods like generalization and suppression. As described in Chapter 2,  $k$ -anonymity principle necessitates that each record in the released dataset is indistinguishable from at least other  $k - 1$  records appearing in the dataset. This principle assures that the probability of discovering any individual's information based on the published dataset does not exceed the threshold  $1/k$ .

In the framework proposed by this work, each local site applies this anonymization approach and sends a sample of anonymized data to the merger site. For this sample data (and not the data stream), the anonymity parameter is set as  $k = 3$  and privacy leakage is obstructed using cell suppression method. Underneath this method, values of some attributes in the instances may be missing due to suppression and thus imputation methods should be deployed to determine such values.

- (2) Artificial Data Generation through Decision Tree Paths: The disjoint classification rules formed paths of decision tree classifiers received from the local parties can be used as a template by the merger site to generate pseudo-data. The data generated in this manner is efficient as paths of a decision tree accentuate the most significant patterns of the original data. Moreover, this method can successfully generate high-quality data without violating the privacy of individuals as the classification trees published by the local sites are output-privacy-preserving.

The values of the attributes for this artificial data are assigned using the following approach: For attributes appearing in the path of the decision tree, the values are assigned using the respective path labels. But, if the attribute is not present in any decision tree path, its value in the resultant tuple is determined using certain heuristics or imputation methods.

The imputation method used in this work is as follows:

- (i) Each party should send a sample of instances without the sensitive attribute class-label. Let  $D_a$  be this sample data.
- (ii) Let the artificial data generated be  $D_m$ .  $D_m$  will be having instances in which at least one of the features is missing.
- (iii) For each instance  $x$  in  $D_m$ :
  - (a) Divide the instance into observed and missing parts as  $x = [x_o; x_m]$
  - (b) Calculate the distance between the  $x_o$  and all the instances from the set  $D_a$  (using only those features in the instance vectors from  $D_a$  which are in  $x_o$ ).
- (iv) Use the  $K$  nearest instances ( $K$ -nearest neighbors) and perform a majority voting to derive the missing values in  $x_m$ .

All the attributes are categorical since the data is discretized a priori. Sending a sample of instances without the concerned class-label is acceptable since class-label is the only sensitive attribute.

### 6.4.3 Baseline methods for comparison

To demonstrate the effectiveness of the proposed algorithm, its performance is compared with 3 other methods. Since the literature does not have exact methods of privacy-preserving classification of horizontally partitioned data streams, the privacy-preserving data stream classification methods are adapted to the distributed data environment and utilized for comparative analysis with the proposed approach.

Details of these methods are described in the following:

- (1) Ensemble of SDTP Classifiers (SDTP Ensemble): The experimental results presented in the previous chapter suggest that extended versions of SDTP classifier (Friedman, Wolff, and Schuster; Aggarawal; Golab and Ozsu) can prove to be effective. As stated in Chapter 2, ensembles are popular methods that combine base learners to form an

efficient classifier. SDTP Ensemble uses the Bagging approach (Breiman) to combine the STDP classifiers received from the participating sites. The ensemble combines best N classifiers based on their predictive accuracy. Since each of these members are output-privacy-preserving classifiers, the resultant ensemble turns to be an output-privacy-preserving classifier built from horizontally partitioned data streams.

- (2) Ensemble of DAHOT Classifiers (DAHOT Ensemble): The approach creates an ensemble of DAHOT classifiers using the Bagging approach (Breiman). The ensemble combines best N classifiers based on their predictive accuracy. Since DAHOT is proved as an efficient output-privacy-preserving data stream classifier in Chapter 5 and ensembles have efficacy in combining base classifiers, DAHOT ensemble seems comparable to our proposed approach.
- (3) Hybrid of SDTP and GPeCT (SDTP-GPeCT): Since preserving the privacy is a chief goal of the work, the privacy-preserving SDTP classifiers (Friedman, Wolff, and Schuster; Aggarawal; Golab and Ozsu) obtained from the participating parties are hybridized with GPeCT to produce a global privacy-preserving classifier on horizontally partitioned data streams. These SDTP classifiers form the initial population of GP. That is, rather than using DAHOT classifiers, SDTP classifiers are deployed as GP individuals and the same approach described in Section 6.4 is utilized. The final classifier thus becomes a suitable candidate for comparison with the proposed approach.

#### 6.4.4 Implementation details

Some initial runs were performed to compare various values of tournament size, trade-off factor, and validation factors. Table 6.3 summarizes the values of all the GPeCT algorithm parameters used for experiments. Since reverse rank selection method is used for mutation, assigning the rate of selection of mutation operator equivalent to (i.e. as high as) reproduction operator is acceptable. Since the targeted application for the proposed algorithm is for the banking sector, all the parameters of GPeCT (and hence for SDTP-

Table 6.3: GPeCT algorithm parameters

Parameter	Value
Initial population size	30
Maximum number of generations	100
Tournament size	7
Trade-off factor $\lambda$	0.01
Validation factor $c_1$	5
Validation factor $c_2$	0.5
Termination criterion	5,000 Fitness Evaluation OR 100% training accuracy OR the best $p = 3$ fitness for $q = 3$ consecutive generations remains same
Probability of Reproduction ( $P_r$ )	20%
Probability of Crossover ( $P_c$ )	60%
Probability of Mutation ( $P_m$ )	20%

GPeCT and DAHOT-GPeCT) have been tuned on the basis of data streams depicted in Table 6.2. Further, 10 different executions of GP are conducted and an average case is reported here. For SDTP Ensemble and DAHOT Ensemble,  $N = 5$  is used, that is, best 5 classifiers are combined to form the Ensemble.

## 6.5 Results and Discussion

The resultant global privacy-preserving classifiers (created from local classifiers built on horizontally partitioned data streams) are compared for their predictive accuracy at each site, the number of nodes in the global classifiers, their training time and the information loss using these global classifiers.

Figure 6.2 shows the results of predictive accuracy (in %) of the global classifiers at the merger site: SDTP Ensemble, DAHOT Ensemble and best three solutions (BSF – Best So Far) obtained using SDTP-GPeCT and DAHOT-GPeCT. Figure 6.3 to Figure 6.5 show the results of predictive accuracy (in %) of the stated global classifiers on data streams at the three local sites individually. The number of nodes in each of these six classifiers is shown in Figure 6.6. In case of ensemble classifiers, the number of nodes represents a total of number of nodes combined from each base classifier. Figure 6.7 shows the information loss using the four global classifiers whereas Figure 6.8 shows the training time of each.

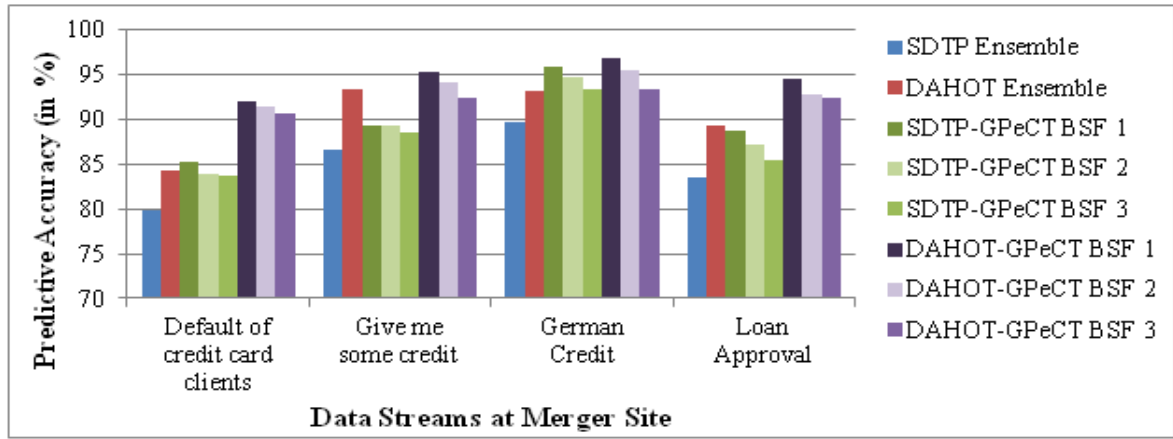


Figure 6.2: Predictive accuracy (in %) of classifiers on data streams at merger site

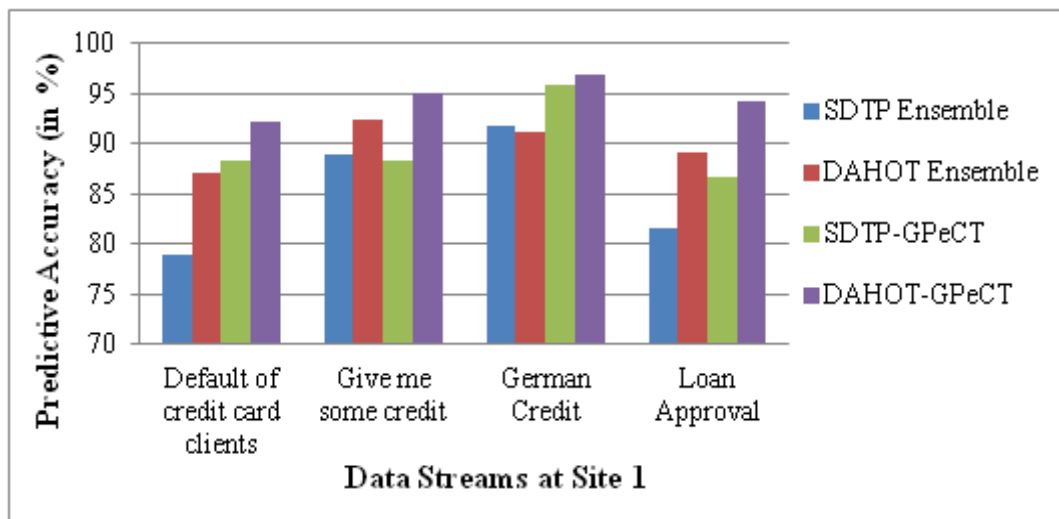


Figure 6.3: Predictive accuracy (in %) of classifiers on data streams at site 1



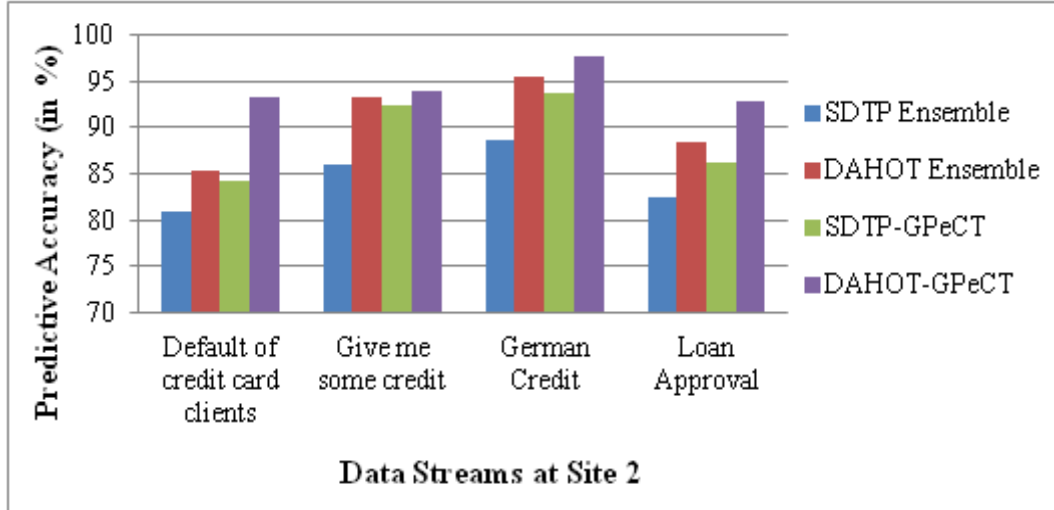


Figure 6.4: Predictive accuracy (in %) of classifiers on data streams at site 2

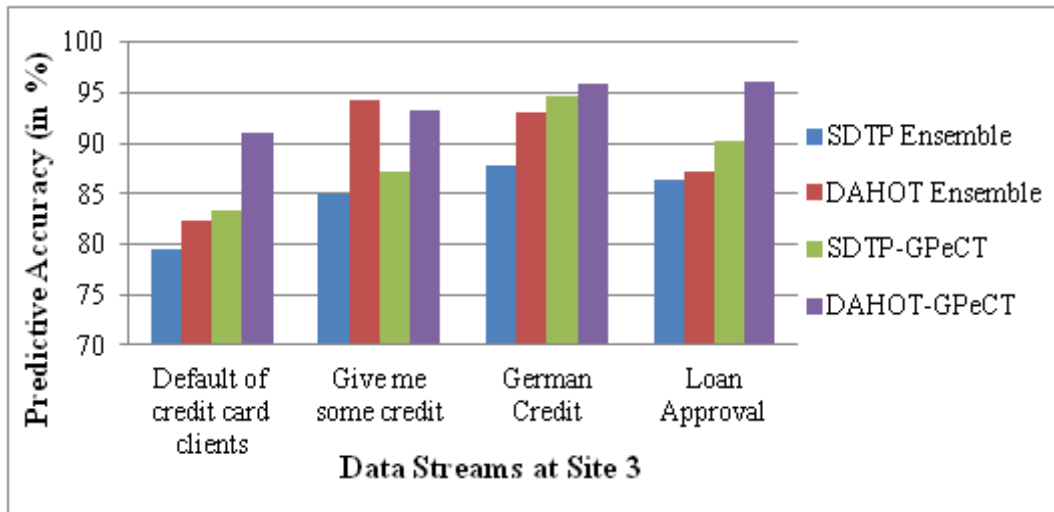


Figure 6.5: Predictive accuracy (in %) of classifiers on data streams at site 3

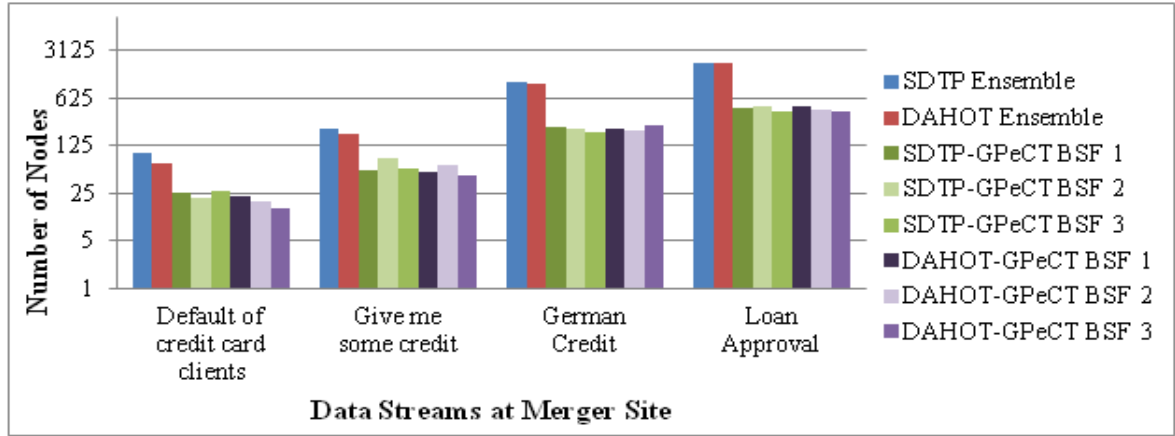


Figure 6.6: Number of nodes in classifiers on data streams at merger site

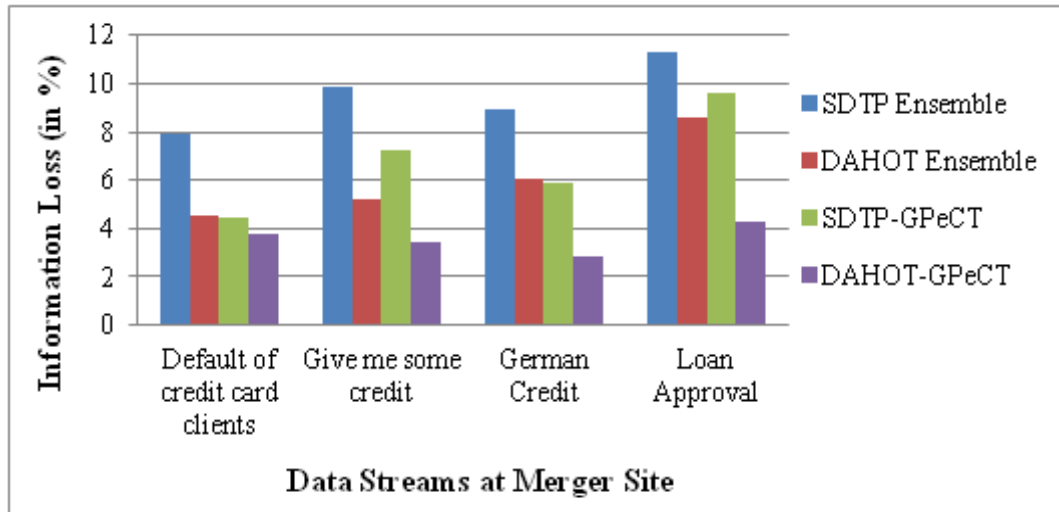


Figure 6.7: Information loss (in %) of classifiers on data streams at merger Site

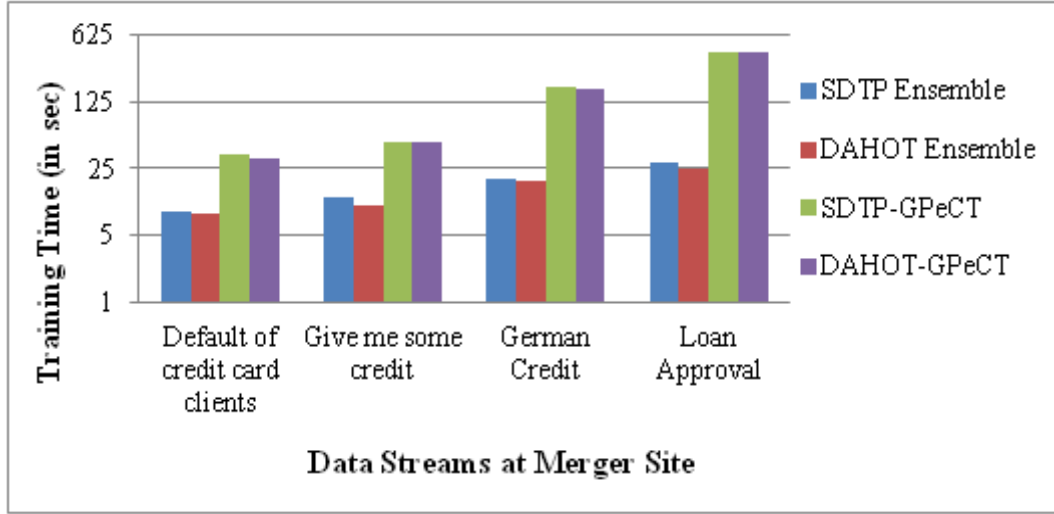


Figure 6.8: Training time (in s) of classifiers on data streams at merger site

The effect of using ensemble methods on the predictive accuracy of a classifier is shown in Figure 6.2 to Figure 6.5. That is, the performance of the global classifier SDTP Ensemble has improved as compared to the single classifier (as compared to SDTP used at local sites). Undoubtedly, the testing instances are different at both the ends, but even on periodic evaluation, SDTP could not reach the accuracy level achieved by SDTP Ensemble. Similarly, the accuracy of DAHOT Ensemble has also increased.

Further, discussing about the comparative performance of the classifiers on the same testing data at the merger site, the predictive accuracy of the SDTP Ensemble is lowest while DAHOT Ensemble ranks high in predictive accuracy, especially on ‘Give me some credit’ and ‘German Credit’ data streams. In fact, as can be seen in Figure 6.2, on all the data streams, the predictive accuracy of DAHOT Ensemble has been comparable to SDTP-GPeCT classifier. Rather than SDTP ensemble, the SDTP-GPeCT has shown an improved performance and all of the three fittest classifiers (BSF1, BSF2 and BSF3) obtained using the later have higher predictive accuracy as compared to the former. The improved performance is due to GP’s ability to produce optimized classifiers from the population at hand. DAHOT-GPeCT seems to be the most successful classifier among all as it inherits the benefits of the efficient DAHOT classifier as well as the optimization proficiency of GPeCT.

## CHAPTER 6. GENETIC PROGRAMMING-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS

As shown in Figure 6.3 to Figure 6.5, similar observations are derived using the experimental evaluation of classifiers at all the local sites. DAHOT-GPeCT gives the best performance in terms of predictive accuracy followed by SDTP-GPeCT.

In the application targeted in this thesis, the induced classifier is utilized to make financial decisions and hence the conclusions of the classifier should be easily interpretable. Classification trees with large number of nodes reduce the interpretability. Thus the fitness function designed for GPeCT prefers a classification tree with lesser number of nodes. As a result, SDPT-GPeCT classifiers BSF1, BSF2 and BSF3 as well as DAHOT-GPeCT classifiers BSF1, BSF2 and BSF3 have lesser nodes which is depicted in Figure 6.6. SDTP Ensemble and DAHOT Ensemble combine  $N = 5$  decision trees, and additionally, selection of ensemble members is made based on the predictive accuracy of classifiers without favoring small classification trees. Hence, the number of nodes in the classifiers induced using these methods are higher. This loss in interpretability can be ignored if the predictive accuracy is remarkably high.

At the merger site, the artificially generated training instances are utilized only by algorithms hybridized using GPeCT wherein their role is to evaluate the classifiers and select from them the ones on which genetic operators will be applied in every generation. The classifiers worked upon at the merger site are already privacy-preserving and no explicit privacy-preserving technique needs to be applied. As a result, once the local classifiers are induced, no more information loss occurs in formation of the global classifier. However, one may consider that the information loss using any global classifier is eventually due to training instances at the merger site which are misclassified by the classifier. To be precise, the information loss may be computed as the difference between the training error due to the privacy-preserving global classifier and its non-privacy-preserving counterpart. Such information loss can be easily ignored as no information from the arriving data streams is lost.

However, taking such information loss into consideration, Figure 6.7 shows the information loss incurred by the global classifiers targeted in this chapter. SDTP Ensemble has high information loss whereas the information loss using DAHOT Ensemble and

DAHOT-GPeCT is almost equivalent, showing the efficacy of these global classifiers in preserving privacy while being accurate in classification. The DAHOT-GPeCT classifier has minimum information loss as compared to all other classifiers, again owing credit to the adeptness of its constituents GPeCT and DAHOT.

Figure 6.8 shows the time required to induce the global classifiers at the merger site. Classifiers evolved using GP, which are, SDTP-GPeCT and DAHOT-GPeCT take more time than SDTP Ensemble and DAHOT Ensemble as they run for a number of generations with each run consuming little time. However, the construction of classifier is a one-time cost and once generated, it can be used to classify the newly arriving data stream instances for long. Hence, the time required in training SDTP-GPeCT and DAHOT-GPeCT is acceptable. The prediction time required by each of these classifiers isn't shown as it is very small and almost similar.

## 6.6 Summary

This chapter proposes an algorithm named GPeCT that merges Genetic Programming and Classification Trees considering accuracy and interpretability as optimization parameters. The fitness function is designed accordingly and the results of experimental evaluation show that the classifiers hybridized with GPeCT have improved efficacy. The goal of privacy-preserving classification of horizontally partitioned data streams is satisfactorily achieved because the base classifiers are output-privacy-preserving and are built from continuously arriving data streams.

The proposed DAHOT-GPeCT classifier outperforms all other classifiers on the four data streams concerning decision making in banking sector. The minimum and maximum improvement in predictive accuracy using DAHOT-GPeCT classifier at the merger site as compared to other classifiers is 7.1% and 12.22% respectively.

Further, the classifiers hybridized using GPeCT have high interpretability and the results of best three classifiers BSF1, BSF2 and BSF3 obtained at the termination of GP prove the same. Additionally, the comparable performance of DAHOT Ensemble with

*CHAPTER 6. GENETIC PROGRAMMING-BASED PRIVACY-PRESERVING  
CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS*

SDTP-GPeCT suggests that the advantages of ensemble learning can be utilized with GP to further optimize the performance of the classifier.

Thus, taking inspiration from the experimental conclusions derived in this chapter, a novel technique for privacy-preserving classification of horizontally partitioned data streams using an amalgamation of ensemble learning and GP is proposed and implemented in the next chapter.



# Chapter 7

## Ensemble-Based Privacy-Preserving Classification of Horizontally Partitioned Data Streams

### 7.1 Introduction

Formally as well as empirically, ensembles have proved to be one of the most efficient techniques in the fields of data mining and machine learning. As a result, ensemble learning has achieved wide popularity and the area has seen an active involvement of researchers proposing many algorithms to enhance the prediction ability of such ensemble systems. The efficacy of ensemble learning in producing privacy-preserving classifier from horizontally partitioned data streams has been witnessed from the experiments in Chapter 6. However, there is a scope of improvement in these methods which is addressed in this chapter.

One of the major concerns in machine learning and data mining research is that of generalization. Formally, generalization can be defined as the prediction ability of a

---

Part of this chapter appears in: Sanjay Garg and Radhika Kotecha, “DAHOT-GPeCT Ensemble for Output-Privacy-Preserving Classification of Horizontally Partitioned Data Streams”, in communication with IEEE Transactions on Cybernetics (2017).



base learner or a data mining model. Specifically targeting the classification task, the better a classifier performs on the previously unseen data; the better it is considered to have the generalization ability. The ‘bias-variance’ quandary depicts the significance of generalization ability of a learner.

The bias term refers to the measure of how much the best classifier  $h^*$  in the classifier search space  $H$  deviates from the target concept. The variance term refers to the measure of how much different realizations of a model can vary from  $h^*$ , selecting wrong classifiers in  $H$ .

Figure 7.1 illustrates the graphical visualization of bias-variance quandary using a bulls-eye diagram (Fortmann). The center circles (bulls-eye) depict the target concept. The accuracy of classification degrades while shifting away from the bulls-eye. Several different hits on the target are obtained by reiterating the complete classifier induction procedure. Every hit symbolizes a specific realization of the classifier. If the bias is high,

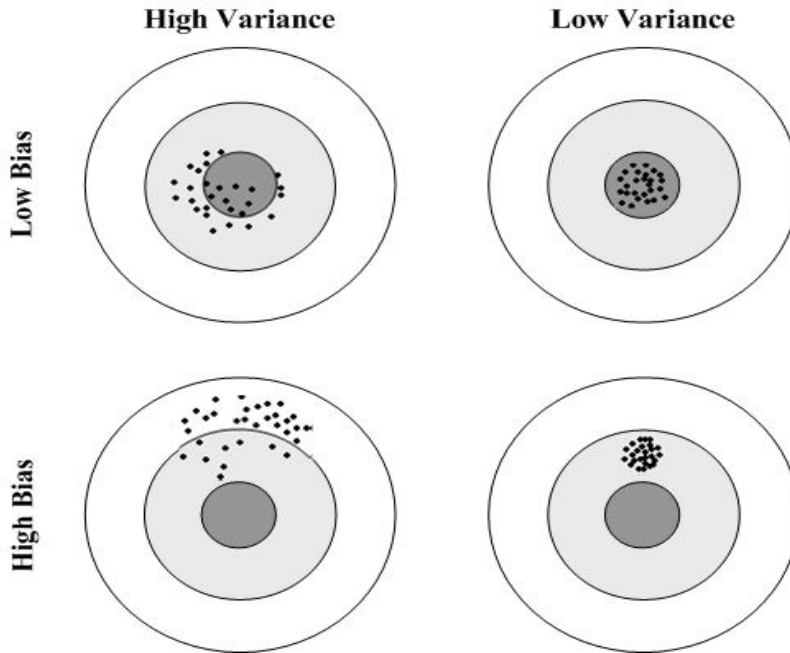


Figure 7.1: Bull's eye diagram for bias-variance quandary

both training as well as test error will be high. Whereas, if the variance is high, training error will be low, and test error will be high.

## *CHAPTER 7. ENSEMBLE-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS*

The generalization ability of ensembles improves when the ensemble members are diverse and accurate (Chawla and Sylvester). That is, an important requirement for successful ensemble learning is to include base learners (classifiers) where errors on the training set are at least to some extent uncorrelated.

The empirical evidences demonstrated in Chapter 6 prove the potential of evolutionary algorithm Genetic Programming in giving accurate classification results. The evolutionary algorithms work on a population of individuals (classifiers in this case) and return the best-of-run classifier as the concluding output. But, along with the best-of-run classifier, the other classifiers in the population are also effective at classification. Such classifiers from the population can be utilized as members of the ensemble. By watchfully selecting diverse classifiers, an efficient ensemble classifier can be obtained and thus the research that combines ensemble learning and evolutionary computation is increasingly capturing attention.

The literature presents several ways to enforce diversity within an ensemble classifier setup. One such method focuses on co-evolution and creating an efficient evolutionary ensemble learning technique based on the theory of multi-objective evolutionary optimization.

Making use of this concept, a new fitness function that adopts diversification heuristics to extract diverse classifiers is presented in this chapter. Since the classifiers are adequately diverse and the errors made by them on the training data are independent, an ensemble of these classifiers that utilizes the majority vote technique will have good generalization ability.

Thus, an evolutionary ensemble classifier has proposed in this chapter and its performance has been verified using experiments. As expected, the proposed approach meets the goal of the work, which is, efficient privacy-preserving classification of horizontally partitioned data streams.

## 7.2 Related Work

Genetic Programming is gaining popularity for their contribution in building efficient homogeneous as well as heterogeneous ensemble classifiers. There are several ways in which Genetic Programming and ensembles are integrated, like application of Genetic Programming to select ensemble members, application of Genetic Programming to adjust weights of members of the ensemble, etc.

Research (Bhowan et al.) suggests combining Genetic Programming classifiers to form an ensemble where each individual classifier provides a vote on class membership and the aggregate knowledge of these classifiers can be utilized to obtain improved generalization as compared to other individual classifiers. Popular aggregating methods like Bagging, Boosting, etc. can be employed to obtain efficient classification.

Several researchers have proposed approaches to adjoin an additional penalty term in the fitness function of Genetic Programming. One such approach is Negative Correlation Learning (NCL) (Chen and Yao; Liu, Yao, and Higuchi) that encourages ensemble diversity by adding a penalty term in the error function. The goal of NCL is accentuate cooperation among the ensemble members and encourage biased members whose errors are negatively correlated.

NCL uses neural networks as base learners and for a training set  $\{x_n, y_n\}$ , with  $n = 1, 2, \dots, N$ , NCL combines  $M$  neural network classifiers, represented as  $f(x)$  in order to form an ensemble as shown in equation 7.1 (Chen and Yao):

$$f_{ens} = \frac{1}{M} \sum_{i=1}^M f_i(x_n) \quad (7.1)$$

In training a neural network  $f_i$ , the error function  $e_i$  for any network  $i$  is defined as in equation 7.2:

$$e_i = \sum_{n=1}^N (f_i(x_n) - y_n)^2 + \lambda p_i \quad (7.2)$$

where  $\lambda$  is a weighting parameter for the correlation penalty term  $p_i$  depicted in

equation 7.3 and Figure 7.4:

$$p_i = \sum_{n=1}^N \left\{ (f_i(x_n) - f_{ens}(x_n)) \sum_{j \neq i} (f_j(x_n) - f_{ens}(x_n)) \right\} \quad (7.3)$$

$$= - \sum_{n=1}^N (f_i(x_n) - f_{ens}(x_n))^2 \quad (7.4)$$

The aim is to minimize  $p_i$ , and negatively correlate each classifier's error with the error occurring through the remaining ensemble. The weighting parameter  $\lambda$  commands a trade-off between the training error and the correlation penalty. When  $\lambda = 0$ , each neural network classifier in the ensemble is trained independently. Whereas, with the increase in weighting parameter  $\lambda$ , the emphasis laid on minimizing the penalty term keeps increasing.

The error  $E$  of the ensemble classifier can be obtained by averaging the errors  $e_i$  of the individual classifiers. That is,  $E$  can be minimized by minimizing  $e_i$  individually.

The concept of NCL has been used in several ways to improve the performance of ensemble classifiers. (Chen and Yao) uses evolutionary algorithms and random subsets of features along with NCL to produce accurate and diverse ensembles by emphasizing cooperation in ensemble learning. Dam et al. have also shown improved performance and generalization ability of ensemble classifier on application of NCL.

Another important and popular diversity measure proposed in literature is Pairwise Failure Crediting (PFC) (Chandra and Yao) which is a population-level measure of diversity. Unlike NCL that compares an individual's output to the output of the ensemble, PFC measures the training error of an individual with all other individuals in the population.

Both NCL and PFC have proved to be efficient in producing diverse ensembles of classifiers due to their nature of promoting evolution of diverse individuals in the population. But, since these measures tend to be biased towards the majority class, Bhowan, Johnston, and Zhang have applied these measures to find the diversity of individuals separately for each class in order to address the problem of imbalanced class distributions. The final diversity-based fitness function is the average of the measure for minority as

well as majority classes so that the individuals in the ensemble are uniformly diverse with respect to all the classes.

A method named Boost Cellular Genetic Programming Classifier (*BoostCGPC*) (Folino, Pizzuti, and Spezzano) implements the AdaBoost.M1 boosting algorithm on a parallel computer by using the algorithm CGPC (Cellular Genetic Programming for data classification) as base classifiers. For training dataset  $S$  having  $N$  instances, if  $P$  number of processors are utilized to run the algorithm, *BoostCGPC* divides the entire population of classifiers in  $P$  subpopulations, applies uniform sampling with replacement on  $S$  to produce  $P$  subsets of instances of size  $n$  ( $< N$ ) and constructs an ensemble of classifiers by choosing the fittest individual from each subpopulation. Bag cellular genetic programming classifier (*BagCGPC*) (Folino, Pizzuti, and Spezzano) utilizes the same parallelization strategy of *BoostCGPC* to induce classification trees but performs Bagging.

A distributed Boosting Cellular Genetic Programming Classifier (Follino, Pizzuti, and Spezzano) that uses an algorithm named Clustering Boost Cellular Genetic Programming Classifier (*ClustBoostCGPC*) has proved to be effective in producing accurate ensemble classifier with diversification among the ensemble members. The algorithm is based on clustering technique to construct the ensemble of classifiers. Unlike *BoostCGPC* that selects the individual with highest fitness from each subpopulation; *ClustBoostCGPC* forms clusters of individuals that share a similarity measure and selects an individual with highest fitness from each cluster. As per the property of clustering technique, since the intra-cluster distance is lowest and inter-cluster distance is highest, a range of diverse classifiers can be obtained on selecting an individual from each cluster.

A new category of algorithms named Orthogonal Evolution of Teams (OET) proposed by Soule and Komireddy (Komireddy and Soule) are cooperative and co-evolutionary algorithms that focus on evolving ensemble components (classifiers) with diverse specializations. Two algorithms OET1 and OET2 are given by (Thomason and Soule) which consists of individual selection with team replacement (IT), and team selection with individual replacement (TI) respectively.

The OET1 algorithm performs selection on individuals and does replacement on teams.

## CHAPTER 7. ENSEMBLE-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS

Offspring creation is initialized with an empty team and fittest individuals are added using tournament selection. In order to be selected for a parent team, an individual must have high fitness and so as to avoid being replaced, any team in the population should have a high fitness.

Quite the reverse, the OET2 algorithm carries out selection on teams and does replacement on individuals. Two teams that are highly are selected by tournament selection to play as parents and crossover and mutation are performed on these teams in order to generate two new children teams. OET2 does replacement by comparing the fitness of team members in the children to the fitness of team members in the population. Individuals with lower fitness get selected to be replaced by individuals in the new children team. In order to be selected for a parent team, a team must have high fitness, that is, the members must cooperate better, and so as to avoid being replaced, any individual team member must have high fitness. OET1 and OET2 algorithms form evolutionary pressure for individuals to perform efficiently and for their teams to perform efficiently too.

Conventionally, the goal of any classification algorithm is to maximize the accuracy on training instances. But, not all instances are equally informative. An instance is considered hard if it is misclassified by a majority of the classifiers. Thus, any classifier that correctly classifies a hard example is considered more important than any classifier that correctly classifies the training instances rightly classified by nearly all classifiers. Evolutionary algorithms have been employed to address this issue and a new diversity-enforcing fitness function that enforces classifier diversity is proposed in the literature (Gagne et al.). The fitness of classifiers is characterized based on a group of reference classifiers, denoted  $Q$ . The hardness of any training instance  $x$  is computed based on the number of classifiers in  $Q$  that incorrectly classify  $x$ . Finally, the fitness of any classifier  $h$  is calculated by the cumulative hardness of the instances correctly classified by  $h$ . Since this approach seems appealing and has proven to be successful in building efficient classifiers, it is analyzed using an example in the next section. Further, a multi-objective fitness function that takes this approach as a building block is proposed in the next section.

## 7.3 An Improved Fitness Function

For an ensemble classifier to be efficient, its member classifiers should be efficient. Since the target is to form an evolutionary ensemble classifier that combines Genetic Programming individuals, the fitness of these individuals is of foremost importance. Hence, a detailed discussion regarding fitness functions is presented in this section. Particularly in sub-section 7.3.1, a popular fitness function from the literature (Gagne et al.) is presented. Followed by it, the concept of multi-objective optimization is discussed in sub-section 7.3.2 and based on that; a novel fitness function is proposed in sub-section 7.3.3.

### 7.3.1 Diversity-enforcing fitness function

A diversity-enforcing fitness function proposed by Gagne et al. works as follows:

Let  $D = \{(x_i, y_i), x_i \in X, y_i \in Y, i = 1 \dots n\}$  be the training data where  $X$  is the instance space and  $Y$  is the set of class labels. Let  $Q$  be the set of reference classifiers. The error function  $e$  is denoted as  $e(h(x_i), y_i)$  which is the (real valued) cost of misclassifying  $x_i$  (that is, assigning a class other than  $y_i$ ). That is,  $e(h(x_i), y_i) = 1$  if  $h(x_i) \neq y_i$ .

The hardness (or weight) of every training instance  $x_i$  is computed based on the number of classifiers in  $Q$  that incorrectly classify  $x_i$ . Formally, the hardness of training instances denoted as  $w_i$  is the average error caused by the reference classifiers, as shown in equation 7.5:

$$w_i = \frac{1}{|Q|} \sum_{h \in Q} e(h(x_i), y_i) \quad (7.5)$$

The fitness  $F$  of every classifier  $h$  is then measured by the aggregate hardness of the instances that are correctly classified by  $h$  as shown in 7.6:

$$F(h) = \sum_{\substack{i = 1 \dots n \\ h(x_i) = y_i}} w_i^\gamma \quad (7.6)$$

CHAPTER 7. ENSEMBLE-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS

That is,  $F(h)$  is the sum over all training instances correctly classified by  $h$ , of their hardness  $w_i$  raised to power  $\gamma$ . The term  $\gamma$  controls the significance of the cumulated hardness  $w_i$  and diversity pressure. That is, on setting  $\gamma = 0$ , fitness function happens to be equal to the number of correctly classified instances and the diversity pressure is eliminated.

The following example shows and justifies how this diversity-enforcing fitness function promotes diverse and accurate classifiers. The example considers  $\gamma = 1$  and for simplicity, only 7 instances are included in the training data. For demonstration, 5 different classifiers are evaluated (irrespective of type of classifiers).

Table 7.1 shows the error of each classifier on the 7 instances (1 indicates an error and 0 indicates a correct classification), the total error by classifiers on each instance, and hardness of each instance calculated using equation 7.5.

Table 7.1: Hardness of training instances based on classifiers							
	<b>Error Function on Instance <math>i</math></b>						
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
<b>Classifier 1</b>	0	1	1	0	0	1	0
<b>Classifier 2</b>	0	0	1	1	0	1	0
<b>Classifier 3</b>	1	1	<b>0</b>	1	0	1	0
<b>Classifier 4</b>	1	0	1	0	1	1	0
<b>Classifier 5</b>	0	0	1	0	0	1	0
<b>Total Error on Instance <math>i</math></b>	2	2	4	2	1	5	0
<b>Hardness of Instance, <math>w_i</math></b>	<b>0.4</b>	<b>0.4</b>	<b>0.8</b>	<b>0.4</b>	<b>0.2</b>	<b>1</b>	<b>0</b>

It can be seen that instance 3 is correctly classified by classifier 3 only hence its



hardness is high and instance 6 is misclassified by all the classifiers and has highest hardness. On the other hand, instance 7 is correctly classified by all classifiers and has hardness  $w_i = 0$ .

Table 7.2: Classifier fitness based on diversity-enforcing fitness measure

Classifier	Total number of instances classified correctly	Accuracy of Classifier	Classifier Fitness = Total Hardness of classifier = $\sum(w_i)$ of correctly classified instances, $\gamma = 2$
1	4	0.57	0.36
2	4	0.57	0.36
3	3	<b>0.43</b>	<b>0.68</b>
4	3	<b>0.43</b>	0.32
5	5	0.71	<b>0.52</b>

From Table 7.2, it can be seen that although the total number of instances correctly classified by classifiers 3 and 4 is same and thus the classification accuracy (defined as the amount of training instances that are correctly classified) is same, but fitness of classifier 3 is greater than classifier 4. This is because classifier 3 is more diverse and classifies hard examples. This justifies that the ensemble created using classifiers that dominantly qualify this fitness measure would be efficient.

However, this fitness function is restricted to a single objective that lays importance only on the hardness of examples. In order to include other measures like overall accuracy, etc. in the fitness functions, multi-objective optimization approach can to be utilized.

### 7.3.2 Multi-objective optimization

Multi-objective optimization (also known as Pareto optimization) is a field of multiple-criteria decision making for problems that require simultaneous optimization of more

than one objective. Such problems are widely found in real-world and are receiving high attention of researchers. One of the most common and straightforward method to solve multi-objective problems is to employ an a priori approach. This approach aggregates several objectives into a single objective function (a scalar value) in accordance with pre-defined preference information of objectives.

In order to indicate the relative preference of objectives, weighting coefficients are assigned to each objective. This can be characterized by a simple aggregation function given in equation 7.7:

$$F(h) = \sum_{i=1}^k \alpha_i f_i \quad (7.7)$$

where  $f_i$  symbolizes the performance of individual  $h$  on the  $i^{th}$  objective,  $\alpha_i$  denotes the weighted relative preference of the  $i^{th}$  objective ( $0 \leq \alpha_i \leq 1$ ), and  $k$  stands for the number of objectives.

### 7.3.3 Proposed multi-objective fitness function

The fitness function presented in Chapter 6 considers accuracy (that is, the proportion of training instances correctly classified) and classifier interpretability (number of nodes in the classification tree) combined and used as a single objective.

Since the ensemble members are required to be diverse in terms of errors made on training data, the diversity-enforcing fitness function described in sub-section 3.1 is required to be additionally considered.

Since these two major objectives are to be included, the fitness function of equation 7.7 becomes as shown in equation 7.8:

$$F(h) = \alpha_1 f_1 + \alpha_2 f_2 \quad (7.8)$$

The fitness function of chapter 6 becomes the first objective  $f_1$  whereas the diversity-enforcing fitness function presented in equation 7.6 becomes the second objective  $f_2$ . A

new fitness function that combines these two objectives is proposed and is depicted in equation 7.9:

Fitness  $F$  of every classifier  $h$  is measured using accuracy, interpretability and cumulated hardness of examples that are correctly classified by  $h$ :

$$F(h) = \alpha_1 \left( \frac{1}{n * (nodes)^\lambda} \sum_{\substack{i=1 \dots n \\ h(x_i) = y_i}} 1 \right) + \alpha_2 \left( \sum_{\substack{i=1 \dots n \\ h(x_i) = y_i}} w_i^\gamma \right) \quad (7.9)$$

where  $\alpha_1$  and  $\alpha_2$  denote the preferences of objectives 1 and 2 respectively,  $n$  represents the total number of training instances,  $nodes$  represent the number of nodes in the classification tree,  $\lambda$  is the accuracy and interpretability trade-off parameter,  $w_i$  is the cumulated hardness and  $\gamma$  denotes the diversity governing parameter.

Analysis of the proposed fitness function (Equation 7.9) is illustrated in Table 7.3 for the same set of instances and classifiers shown in Table 7.1. For easy understanding of the effect of the multi-objective fitness function, the number of nodes in each classifier (tree) is assumed to be 100. The effect of varying number of nodes on the fitness function is already studied and explained in Chapter 6.

It can be seen from Table 7.3 that although the classification accuracy and ultimately the value of objective 1 of classifier 2 is greater than classifier 3, but fitness of classifier 3 is greater than classifier 2 because classifier 3 is more diverse and classifies hard examples. But only when there is a vast difference in accuracies (much higher accuracy) and small difference in hardness (little lower hardness), the classifier with higher accuracy wins. This case can be understood using classifier 3 and 5. Hence, the proposed fitness function proves to be efficient in selecting accurate, interpretable and diverse classifiers. As a result, the ensemble created using the classifiers that dominantly qualify this fitness measure would be proficient.

Table 7.3: Classifier fitness based on the proposed fitness measure

Classifier	Accuracy	Objective $f_1$ with $\lambda = 0.01$	Objective $f_2$ with $\gamma = 2$	Classifier Fitness with $\alpha_1 = 0.4$ and $\alpha_2 = 0.6$
1	0.57	0.54	0.36	0.43
2	0.57	<b>0.54</b>	<b>0.36</b>	<b>0.43</b>
3	0.43	<b>0.41</b>	<b>0.68</b>	<b>0.57</b>
4	0.43	0.41	0.32	0.36
5	0.71	<b>0.68</b>	<b>0.52</b>	<b>0.58</b>

## 7.4 Proposed Approach

This section proposes genetic programming and ensemble-based approach for inducing a global privacy-preserving classifier. Figure 7.2 presents the proposed approach diagrammatically.

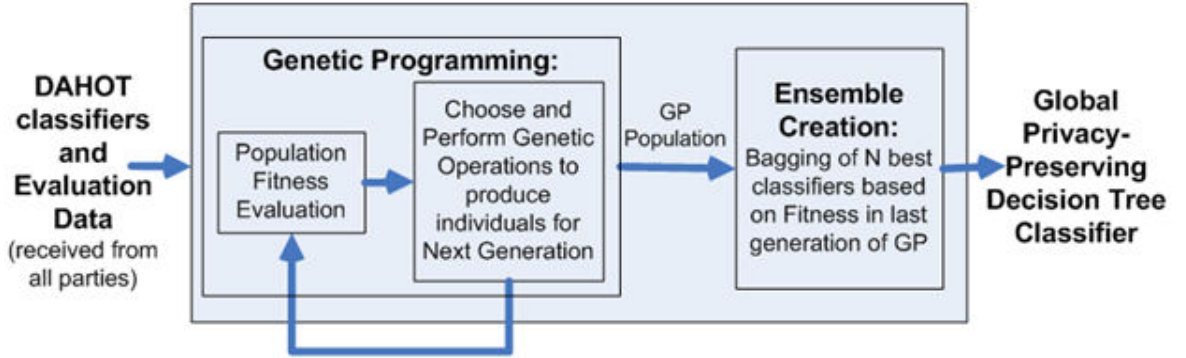


Figure 7.2: Genetic programming and ensemble learning for global privacy-preserving classifier induction

As shown in Figure 7.2, the merger site receives multiple output-privacy-preserving classification trees (DAHOTs) from each participating site where each classifier is built using random subset of attributes and at a gap of receiving  $t$  data stream instances. These DAHOTs undergo genetic programming runs to obtain accurate, interpretable and

diverse classifiers using the proposed fitness function depicted in equation 7.9. These accurate, interpretable and diverse classifiers are then combined to form an efficient ensemble classifier.

The proposed approach is an extension of Algorithm DAHOT-GPeCT presented in Chapter 6 and for completeness the entire approach is presented in section 7.5.1 as Algorithm 5.

### **7.4.1 Proposed algorithm**

The proposed algorithm is shown in algorithm 5.

The algorithm 5 named DAHOT-GPeCT-Ensemble is an extension of algorithm 6.1 proposed in Chapter 6. The difference lies in the final classifier formation (lines 21 and 22). At the termination of GP runs, the population of final generation is sorted in decreasing order of the fitness function values and stored for future usage. With this population (naming it Z), an ensemble classifier E using Bagging (Breiman) approach is created using the first N individuals from Z. The resultant DAHOT-GPeCT-Ensemble is the global classifier formed to address the issue of privacy-preserving classification of horizontally partitioned data streams.

## **7.5 Experimental Evaluation and Analysis**

In order to verify the effectiveness of the proposed approach DAHOT-GPeCT-Ensemble for privacy-preserving classification of horizontally partitioned data streams, several experiments are conducted with different real-world and synthetic data streams. The following sub-sections describe the details of the data streams used as well as implementation details and present the evaluation results.

---

**Algorithm 5** DAHOT-GPeCT-Ensemble. Evolves an ensemble classifier.

---

**Input:** Data instances  $D_s$ ; DAHOT classifiers  $C$  received from  $n$  participating parties;  
validation factors  $c_1, c_2$

**Output:** Ensemble classifier,  $E$

```

1. for  $i = 1$  to  $n$  do           //Population initialization
2.   for  $j = 1$  to  $pop\_size/n$  do
3.     Add a DAHOT classifier  $C_{ij}$  to the population
4.   end for
5. end for
6. Evaluate fitness of each individual in initial population using fitness measure of Equation 7.9 and  $D_s$ 
7. while  $generation < max\_generations$  or Termination criterion is not satisfied do
8.   repeat
9.     Perform genetic operations on parent trees to generate offspring trees for the new population
10.    if (Size of individual  $< c_1 * \text{Size of } best\_so\_far$ ) then
11.      Evaluate its fitness and increment  $new\_pop\_size$ 
12.    else if (Accuracy of individual  $> c_2 * \text{Accuracy of } best\_so\_far$ ) then
13.      Evaluate its fitness and increment  $new\_pop\_size$ 
14.    else
15.      Discard the individual
16.    end if
17.  until  $pop\_size$  individuals are produced
18.     $generation \leftarrow generation + 1$ 
19.     $pop\_size \leftarrow new\_pop\_size$ 
20. end while
21. Let  $Z$  be the sorted population of last generation in decreasing order of fitness function value
22. Apply Bagging to build an ensemble classifier  $E$  using first  $N$  individuals from  $Z$ 

```

---

### 7.5.1 Data streams

Since the work in this chapter extends the work presented in Chapter 6, at the participating sites, the same data streams with identical composition as listed in Table 6.2 are used here. At the merger site too, the same artificially generated data as described in section 6.4.2 is used. The target application, that is, decision making in banking sector remains the same.

### 7.5.2 Baseline methods for comparison

To demonstrate the effectiveness of the proposed algorithm DAHOT-GPeCT-Ensemble, its performance is compared with 3 other methods of privacy-preserving classification of horizontally partitioned data streams. Two of these global classifier induction methods are SDTP-GPeCT and DAHOT-GPeCT, which have been described and implemented in the previous chapter. The third method is SDTP-GPeCT-Ensemble that forms an ensemble of SDTP classifiers evolved using GPeCT. Using the popular Bagging approach (Breiman, an ensemble of  $N$  best classifiers is formed based on fitness of the classifiers.

### 7.5.3 Implementation details

For experiments, the values of algorithmic parameters of GPeCT are same as stated in Table 6.3 of Chapter 6. The number of classifiers in the ensemble,  $N$  is set equal to 5, that is, the global privacy-preserving data stream classifier is formed by bagging of 5 fittest classifiers from last generation of DAHOT-GPeCT built using the fitness function of Equation 7.9. The parametric values for the fitness function of Equation 7.9 are  $\lambda = 0.01$ ,  $\gamma = 2$ ,  $\alpha_1 = 0.4$  and  $\alpha_2 = 0.6$ .

### 7.5.4 Results and discussion

The global privacy-preserving classifiers built on horizontally partitioned data streams are compared for their predictive accuracy at the merger site as well as each local participating sites, the number of nodes in the global classifiers, their training time and the information loss using these global classifiers.

Figure 7.3 shows the results of predictive accuracy (in %) of the global classifiers at the merger site: SDTP-GPeCT, DAHOT-GPeCT and best three solutions (BSF – Best So Far) obtained using SDTP-GPeCT Ensemble and DAHOT-GPeCT Ensemble.

The results of predictive accuracy (in %) of the stated global classifiers on data streams at the three local sites individually is shown in Figure 7.4 to Figure 7.6. The number of nodes in each of these six classifiers is shown in Figure 7.7. In case of ensemble classifiers,

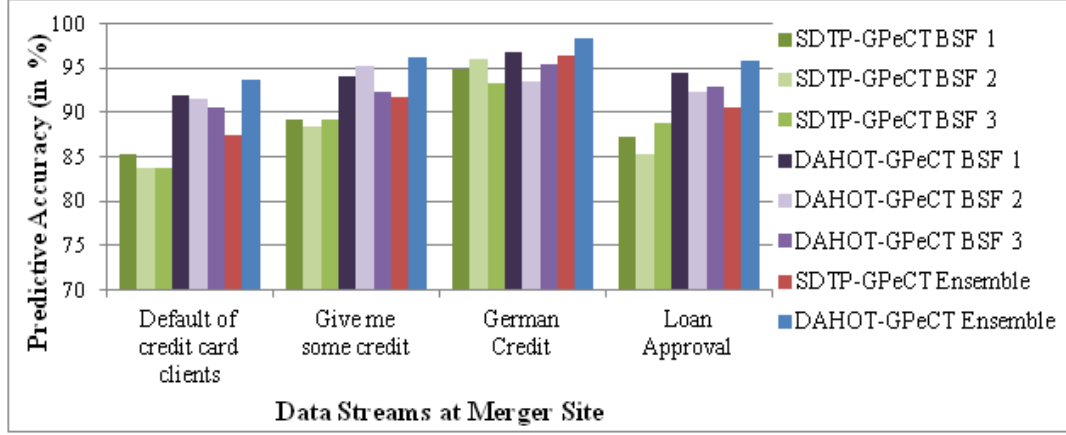


Figure 7.3: Predictive accuracy (in %) of classifiers on data streams at merger site

the number of nodes represents a total of number of nodes combined from each base classifier. Figure 7.8 shows the information loss using the four global classifiers whereas Figure 7.9 shows the training time of each.

The results shown in Figure 7.3 are as expected. The proposed DAHOT-GPeCT Ensemble shows better classification accuracy as compared to the other three methods on the data stream at local sites as well as merger site. The difference in classification accuracy is significant as compared to the best classifiers (BSF) of SDTP-GPeCT and the SDTP-GPeCT Ensemble. At the merger site, the predictive accuracy of DAHOT-GPeCT Ensemble is 93.64% on 'default of credit card clients' data stream, 96.15% on 'give me some credit' data stream, 98.32% on 'German credit' data stream and 95.77% on 'loan approval' data stream. This high accuracy is the result of the novel fitness function of Equation 7.9 that gives significance to accurate and diverse classifiers. Such accurate and diverse classifiers have higher fitness and become members of the ensemble.

It can be seen from Figure 7.4, Figure 7.5 and Figure 7.6 that all the classifiers produce similar predictive accuracy at each site. The minimum and maximum difference between accuracies of SDTP-GPeCT obtained at the three participating sites is 1.8% and 3.72% respectively. Using DAHOT-GPeCT classifier, the minimum and maximum difference between predictive accuracies obtained at the three participating parties is 1.68% and 3.16% respectively. In case of SDTP-GPeCT, the minimum and maximum differ-



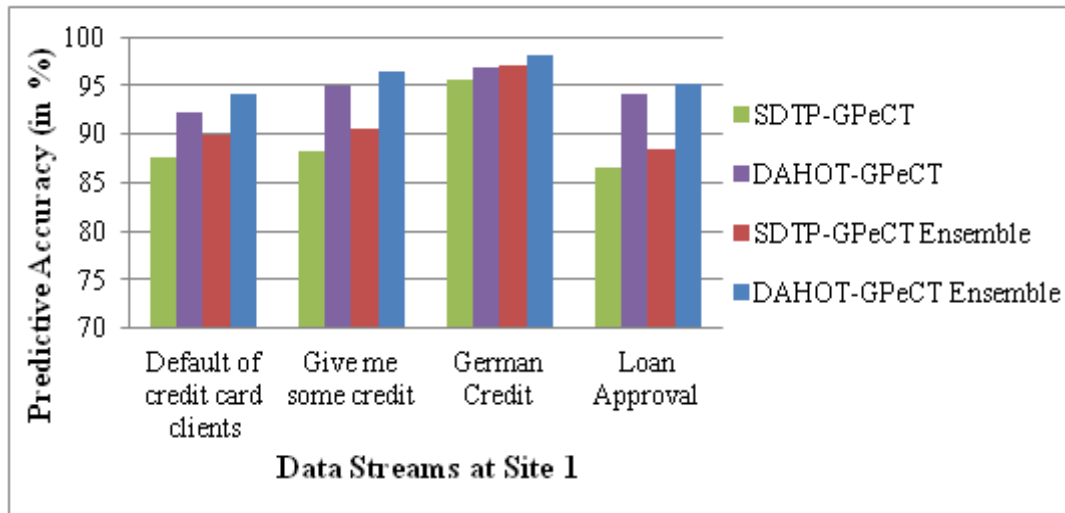


Figure 7.4: Predictive accuracy (in %) of classifiers on data streams at site 1

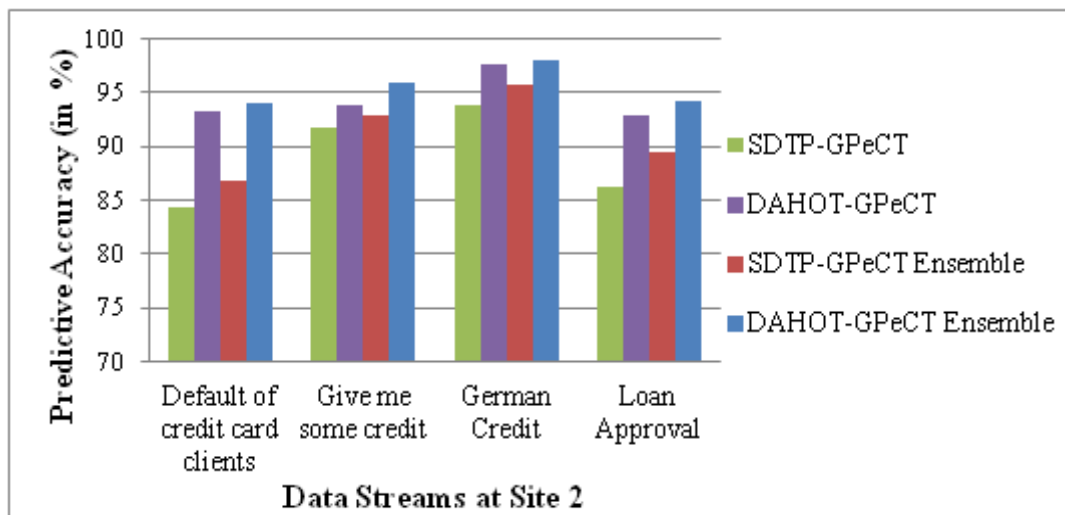


Figure 7.5: Predictive accuracy (in %) of classifiers on data streams at site 2

ence is 1.45% and 3.29% respectively whereas using DAHOT-GPeCT the minimum and maximum difference in accuracies at the three sites is 0.89% and 2.84% respectively.

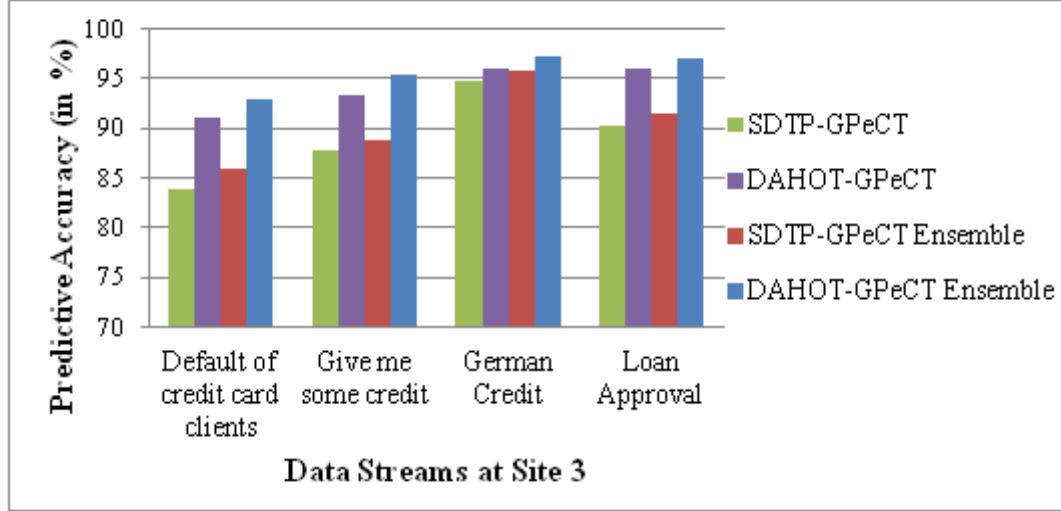


Figure 7.6: Predictive accuracy (in %) of classifiers on data streams at site 3

This minor difference in accuracy obtained at different sites shows the stability of each of the algorithms and the stability strengthens the conclusion about DAHOT-GPeCT-Ensemble being the most suitable for privacy-preserving classification of horizontally partitioned data streams.

The number of nodes in each of the classifiers is shown in Figure 7.7. For the ensemble classifiers, the cumulative number of nodes in all the base classifiers is considered and is predictably high. But, the advantages these ensemble classifiers offer, especially those offered by the proposed DAHOT-GPeCT-Ensemble, makes the loss in interpretability acceptable. As mentioned in Chapter 6, once the local classifiers are induced, no more information loss occurs in formation of the global classifier but one may consider that the information loss using any global classifier is eventually due to training instances at the merger site which are misclassified by the classifier. Information loss using the global classifiers targeted in this chapter is shown in Figure 7.8. Such information loss can be easily ignored as no information from the arriving data streams is lost. Irrespective of this, the information loss due to DAHOT-GPeCT-Ensemble is small and minimum.

Figure 7.9 shows the time required to induce the global classifiers at the merger site.

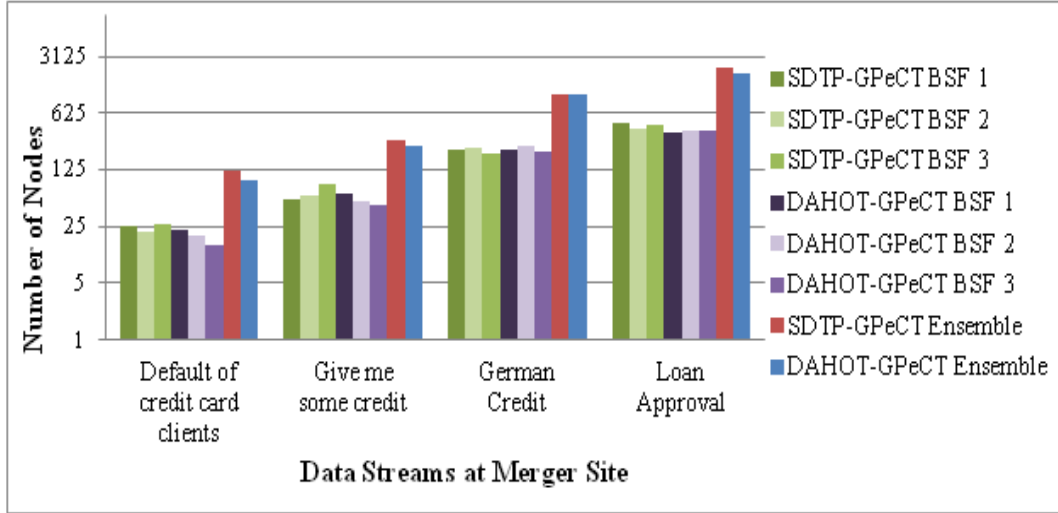


Figure 7.7: Number of nodes in global classifiers on data streams at merger site

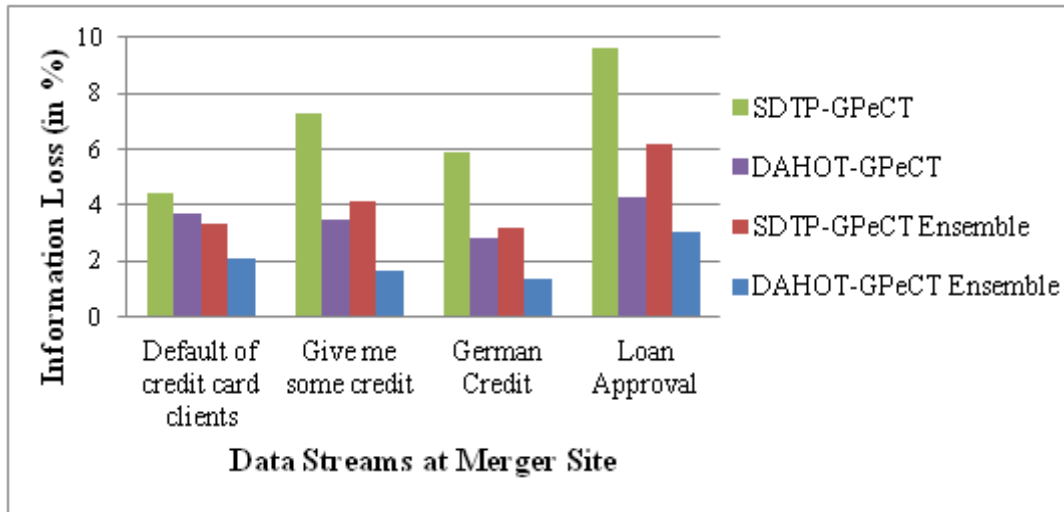


Figure 7.8: Information loss (in %) of classifiers on data streams at merger site

Each of the classifiers is evolved using GP and run for a number of generations. Since each run consumes little time due to computation of fitness function, the overall training time of these classifiers is high. Further, it can be seen that the time required by all four classifiers is nearly equivalent and proportional to the classification tree size. But again, as the construction of classifier is a one-time cost and once generated, it can be used to classify the newly arriving data stream instances for long and the training time of these classifiers is acceptable. The prediction time required by each of these classifiers is very small and almost similar and hence isn't shown.

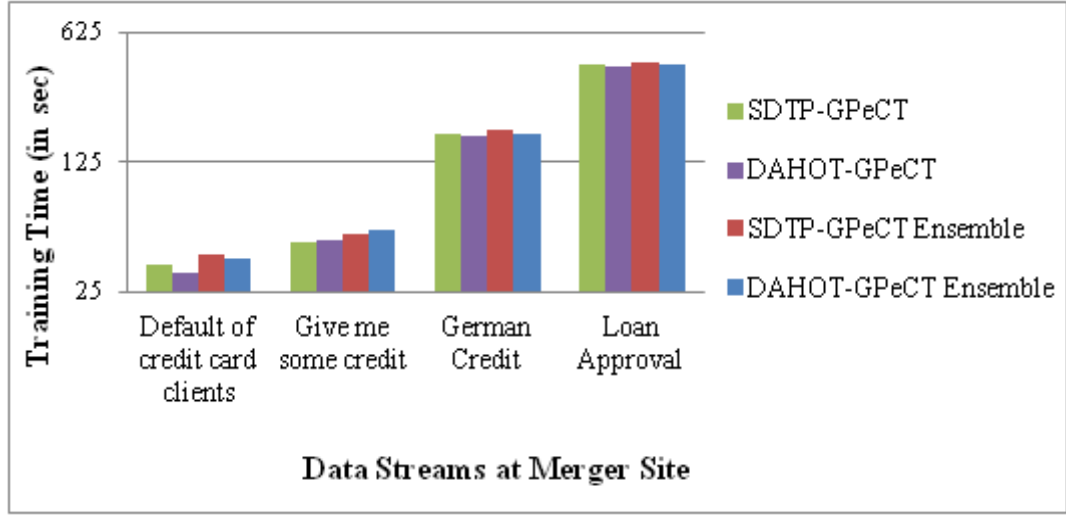


Figure 7.9: Training time (in s) of global classifiers on data streams at merger site

The next section extends the privacy-preserving classification of horizontally partitioned data streams for the scenario where these global classifiers need to be updated.

## 7.6 Classifier Updation

At each local site, the data streams are continuously arriving with some labeled and some unlabeled records. The performance of most effectual classifier build from the streaming data may degrade as time proceeds and new data arrives.

Specifically, when the data stream is not generated by a stationary stochastic process, the data distribution and underlying concept of the data may change with time and may

no longer match the previously received data. As the result of such concept-drift, the prediction accuracy drops. Thus, it is required to revise or refine the classifier model by incorporating new instances as they arrive. The model must not only capture the latest trends and patterns in the streaming data but should also be capable to discard the effect of obsolete patterns.

Since streaming data is arriving at multiple parties with each party willing to contribute in mining valuable patterns, any concept-drift detected at either of the sites may be useful to the other sites. Thus, since each site is using the same global ensemble classifier, it gets essential to update the other sites about any change in the classifier at a particular site.

### 7.6.1 Proposed approach

Several scenarios exist where the classifier needs to be updated due to concept-drift. Such scenarios and the update processes are described as follows:

#### **Scenario 1: At participating (local) sites, after receiving global classifier**

Once the global classifier (say G1) is built, it is sent to all the participating (local) sites where it is employed to work upon the newly arriving data. This newly arriving data stream would contain some labeled and some unlabeled instances.

The unlabeled instances at each participating site are classified using this global classifier G1. Further, the labeled instances arriving at this moment are also passed to the global classifier G1 to predict their outcome and verify the performance of the classifier. Instances may be misclassified either because the global classifier is inaccurate or due to a concept-drift. In either case, the instances that get misclassified by the global classifier G1 are stored in a buffer B. Any individual in the global ensemble classifier G1 that misclassifies the instances is given a negative vote, one per instance misclassification. Simultaneously, a fixed-size sliding window W (initially empty) is maintained at each site and the newly arriving instances are continuously added to the window W. The window keeps sliding over the instances to accommodate the newest instances and discards the

old instances. The combination of buffering and windowing borrows its concept from ADWIN (Bifet and Gavalda).

**Scenario 2: At participating (local) sites, once the buffer B is full**

Assume that the buffer B gets full at participating site P1. In such a case, an output-privacy-preserving decision tree classifier T1 is induced using the instances in the buffer and traditional decision tree induction algorithm CART. Similar to Hoeffding tree, CART uses Gini index as an attribute selection measure and forms binary tree but works on static data. The resultant decision tree is further sanitized to satisfy diversity and anonymity constraints.

The instances in W are used to evaluate the newly created classifier T1 and individuals of the global classifier G1. Classifier T1 or any individual in ensemble G1 that misclassifies the instances is given a negative vote, one per instance misclassification. If any individual of G1 receives more negative votes (at least 15% more) as compared to T1, this newly created classification tree T1 replaces that weakest ensemble member. If such a replacement occurs, a new ensemble classifier G1' is formed at a local site P1.

If there is a momentary concept-drift, the newly created classifier will not be able to compete with the individuals in the ensemble and hence won't replace any ensemble member.

The same scenario can occur at any of the participating sites and the same process of updating classifier is to be followed.

**Scenario 3: At participating (local) sites, when classifier update occurs at other local site**

When a new classification tree replaces an ensemble member at any participating site, this newly created tree is sent to all other participating sites. Assume that a newly induced classification tree T2 replaces one of the ensemble members at site P2. All other sites on receiving T2 will evaluate the ensemble members as well as T2 on the current window and gather votes for each of the classifiers.

If T2 outperforms any of the ensemble members at any site (say site P1), then such a weak ensemble member is replaced by T2 and a new ensemble classifier named G1''

is created at site P1. This locally updated ensemble  $G1''$  is now used at site P1 to classify further instances. The same classifier updating procedure is carried out at all the participating sites.

**Scenario 4: At participating (local) sites, periodically**

Periodically at each site, all the members of ensemble  $G1$  and the newly induced or received trees are evaluated using the instances in the current window  $W$ . Any individual that misclassifies the instances is given a negative vote, one per instance misclassification. It is to be noted that at any local site, even if an individual is replaced by a new classifier, the vote for all the ensemble members of the original global classifier  $G1$  are accumulated. Further, at each site, misclassification penalty votes are accumulated for the newly added classifiers (self-induced or received from other sites, if any) too. Accuracies (weights) of classifiers are computed based on the proportion of votes received.

Each site sends accuracies of all ensemble members as well as the newly added tree to the merger site.

**Scenario 5: At merger site, when classifiers are received from each participating (local) site**

Based on the average of received accuracies of ensemble members and new trees, a fresh ensemble classifier  $G2$  is induced using the most-accurate classifiers. However, totally discarding the trees that formed the previous ensemble might be an over-killing act. Hence, such trees are added to the final population of GP.

The updated global classifier  $G2$  is again transferred to all the local sites and is used to classify unseen instances in the data stream arriving at those sites.

Periodically, all the sites send their votes for every ensemble member to the global site. An additional run of GP is conducted when the performance of the global classifier decreases below a user-specified threshold for minimum accuracy of the ensemble members. The classifier performance is only verified periodically. Re-induction occurs only when the performance gets degraded.

## 7.6.2 Results and discussion

The issue of classifier updation has been addressed by introducing a concept-drift in the ‘Loan Approval’ data stream. Since this data stream is synthetically generated using MOA (Bifet et al.), introducing the concept-drift in the stream after induction of the global classifier is done efficiently. The parameters are set as follows: buffer size  $|B|=1000$  and window size  $|W|=2000$ . The performance of the proposed DAHOT-GPeCT-Ensemble classifier in terms of predictive accuracy evaluated for different scenarios is depicted in Figure 7.10.

The results shown in Figure 7.10 are for a scenario when the incoming data stream at participating site 2 results into updating the global classifier. The newly induced classifier from the incoming data stream replaces a weak ensemble member at site 2 to form a new global ensemble and this new classifier is forwarded to the remaining two sites. After evaluation of the new classifier at site 1 using the instances in its window, the global classifier at site 1 is also updated with a weak ensemble member being replaced with the newly received classifier. But, at site 3, the newly arrived classifier does not prove to be efficient than the old ensemble members and the global classifier is not updated.

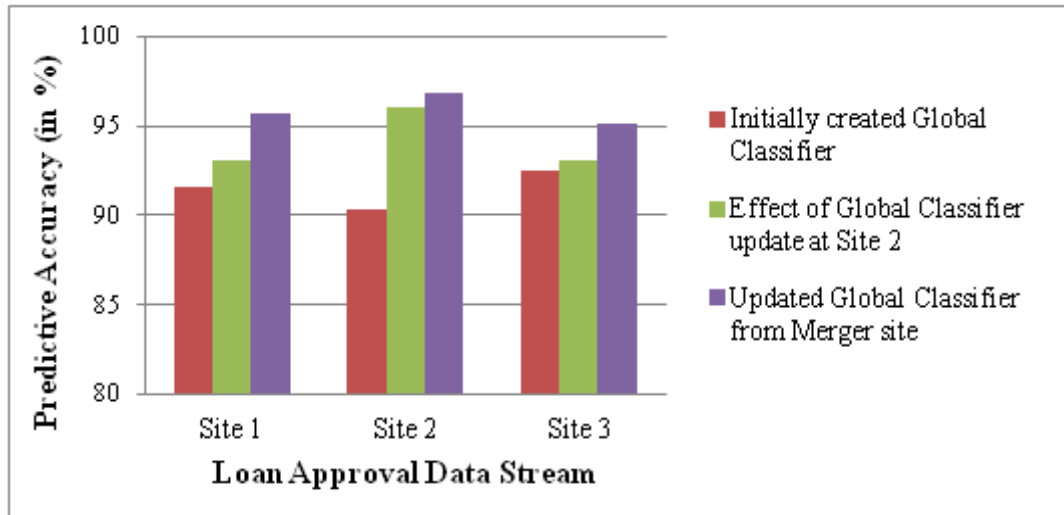


Figure 7.10: Predictive accuracy (in %) of classifiers before and after classifier update

Figure 7.10 shows three different predictive accuracies: 1) the initially created global



classifier evaluated using the newly arrived data stream instances 2) the effect global classifier update at site 2, which results into global classifier update at site 1 as well but not at site 3 and 3) global classifier updated at the merger site on evaluation at participating sites.

For obvious reasons, the effect of global classifier updated at site 2 will result into improved accuracy as compared to the initially created global classifier. Further, it can be seen that since classifier 1 updates the global ensemble by replacing its weakest member with the newly received classifier from site 2, the accuracy of the updated global ensemble classifier at site 1 is improved as compared to the initially build global classifier. The accuracy of DAHOT-GPeCT-Ensemble at site 3 is already high initially and the update at site 2 does not replace any ensemble member at site 3 and ultimately, the initially created global classifier is not updated. But still, the accuracy remains high.

When the global classifier is updated at the merger site and sent back to each of the participating parties for future classification, the predictive accuracy at all these participating parties is improved. This proves that the proposed approach is successful in updating the classifier for performance improvement.

## **7.7 Summary**

This chapter used ensemble learning to improve the performance of DAHOT-GPeCT classifier proposed in the previous chapter. A novel GP fitness function that prefers accurate and diverse individuals is proposed. With the proposed approach, named DAHOT-GPeCT-Ensemble, privacy-preserving classification of horizontally partitioned data streams is carried out by forming an ensemble of these accurate and diverse classifiers evolved using GP.

The proposed DAHOT-GPeCT-Ensemble classifier outperforms all other classifiers on the four data streams concerning decision making in banking sector used in this thesis. Also, the issues in data stream classification presented in Chapter 2 are addressed. The minimum and maximum improvement in predictive accuracy using DAHOT-GPeCT-

## *CHAPTER 7. ENSEMBLE-BASED PRIVACY-PRESERVING CLASSIFICATION OF HORIZONTALLY PARTITIONED DATA STREAMS*

Ensemble classifier at the merger site as compared to other classifiers is 5.03% and 10.37% respectively. Further, the results of updating the global DAHOT-GPeCT-Ensemble classifier for ‘Loan Approval’ data stream show how the periodic update in classifier can improve the prediction accuracy.

The results obtained from the experimental evaluation of the proposed DAHOT-GPeCT-Ensemble classifier are satisfactory and prove the classifier’s efficacy on the data streams relating to the targeted application of credit-risk in banking sector. Thus, the goal of the work in this thesis, that is, privacy-preserving classification of horizontally partitioned data streams has been successfully accomplished.



# Chapter 8

## Conclusions and Future Scope

### 8.1 Conclusions

The techniques proposed in this thesis succeed in achieving the objectives described in Chapter 1.

Having done an in-depth study and empirical analysis of existing literature as well as the proposed approaches for privacy-preserving classification of horizontally partitioned data streams, several conclusions are drawn.

Following conclusions are derived while working to fulfill the first objective of the work, that is: *“to develop an efficient method for preserving output-privacy in data stream classification”*:

To achieve this objective, that is, for the participating parties to publish a local classifier from the continuously arriving data streams, an output-privacy-preserving data stream classifier named DAHOT is proposed.

Applying  $l$ -diversity principle along with  $k$ -anonymity through DAHOT intensifies the uncertainty in linking patterns derivable from the published classifier to individuals whose data is being mined. Since record linkage and attribute linkage attacks on the published classifier are prevented, the privacy requirement for the targeted application is satisfied. Using Hoeffding trees as base classifiers offers several advantages such as easy

amalgamation with privacy principles, high classification accuracy, interpretability and the availability to predict at any point. The empirical performance comparison of DAHOT with few other methods also proves it's efficacy in classifying massive data streams while preserving the required privacy.

While accomplishing the second objective of the work, that is: *“to develop a systematic method for privacy-preserving classification of horizontally partitioned data streams”*, following conclusions are derived:

The DAHOT, genetic programming and ensemble-learning based framework thrives in addressing privacy-preserving classification of horizontally partitioned data streams.

The novelties introduced such as reverse rank selection strategy for mutation, variable-size population, an additional validation stage, accuracy and interpretability based fitness function, etc., make the genetic programming-based approach, named DAHOT-GPeCT competent enough in achieving the desired objective.

An improved version of DAHOT-GPeCT, named DAHOT-GPeCT-Ensemble classifier, proves that an ensemble of output privacy-preserving data stream classifiers (DAHOTs) selected through a novel fitness function that emphasizes classification accuracy, interpretability and diversity is a suitable solution method for parties intending to collaborate and induce a privacy-preserving classifier from their horizontally partitioned data streams.

## 8.2 Future Scope of Work

There are several avenues for future work in this area.

The first one suggests privacy-preserving classification of data streams with multiple sensitive attributes. This requires modification in DAHOT algorithm in terms of the nodes to be pruned. This modification will reduce the amount of pruning and hence decrease the information loss.

Further, the framework proposed in this work is evaluated for data streams having only two classes. The same can be extended for data streams with multiple classes. The scope of the work covers only horizontally partitioned data streams. Privacy-preserving

## CHAPTER 8. CONCLUSIONS AND FUTURE SCOPE

classification can be extended to vertically partitioned data streams as well as arbitrarily partitioned data streams.

The work focuses on preventing record-linkage and attribute-linkage attacks. Privacy models like t-closeness, differential privacy, personalized privacy, etc. designed to prevent various other attacks can be explored to protect the classifier output from malicious attacks.

The proposed framework treats all the attribute values similarly irrespective of its data distribution (which may be skewed). Moreover, the proportion of classes in the data is frequently imbalanced. The proposed framework can be extended to address such issues.

Lastly, the proposed work has been implemented for decision-making in banking sector. The effectiveness of proposed approach can be verified with other data streams of banking sector or other applications of privacy-preserving classification of horizontally partitioned data streams.

These works remain as open issues for researchers in the field of data mining.



# Works Cited

- Abdulsalam, H., D. Skillicorn, and P. Martin. “Classification Using Streaming Random Forests.” *IEEE Transactions on Knowledge and Data Engineering* 23.1 (2011): 22–36. Print.
- Agarwal, R. and R. Srikant. “Privacy-preserving data mining.” *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 2000. 439–450. Print.
- Aggarawal, C. *Data Streams Models and Algorithms, Advances in Database Systems*. Springer Verlag, 2006. Print.
- Aggarwal, C. “On Abnormality Detection in Spuriously Populated Data Streams.” *Proceedings of SIAM Conference on Data Mining*. 2005. Print.
- Aggarwal, C. and P. Yu, eds. *Privacy-Preserving Data Mining: Models and Algorithm, Advances in Database Systems*. Vol. 34. Springer, 2008. Print.
- Aggarwal, C., et al. “A Framework for On-Demand Classification of Evolving Data Streams.” *IEEE Transactions on Knowledge and Data Engineering* 18 (2006): 577–589. Print.
- Aggarwal, G., et al. “Approximation algorithms for  $k$ -anonymity.” *Journal of Privacy Technology* 8 (2005): 1–18. Print.
- Ayala-Rivera, V., et al. “A Systematic Comparison and Evaluation of  $k$ -Anonymization Algorithms for Practitioners.” *Transactions on Data Privacy* 7.3 (2014): 337–370. Print.
- Back, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996. Print.



- Baesens, B., et al. "Benchmarking state-of-the-art classification algorithms for credit scoring." *Journal of the Operational Research Society* 54 (2003): 627–635. Print.
- Bayardo, R. and R. Agrawal. "Data privacy through optimal k-anonymization." *Proceedings of 21st International Conference on Data Engineering*. 2005. 217–228. Print.
- Bertino, E., et al. "Privacy and ownership preserving of outsourced medical data." *Proceedings of 21st International Conference on Data Engineering*. 2005. 521–532. Print.
- Bhowan, U., M. Johnston, and M. Zhang. "Evolving diverse ensembles using genetic programming for classification with unbalanced data." *IEEE Transactions on Evolutionary Computation* 17.3 (2013): 368–386. Print.
- Bhowan, U., et al. "Reusing Genetic Programming for Ensemble Selection in Classification of Unbalanced Data." *IEEE Transactions on Evolutionary Computation* 893-908 (2013). Print.
- Bifet, A. and R. Gavalda. "Adaptive Parameter-free Learning from Evolving Data Streams." *Polytechnic University of Catalonia* (2009). Print.
- Bifet, A., et al. "Data Stream Mining - A Practical Approach." *Technical Report, New Zealand: Department of Computer Science, San University of Waikato* (2011). Print.
- Bifet, A., et al. "MOA: Massive Online Analysis." 2014. <http://moa.cms.waikato.ac.nz>, Accessed 31 March 2017. Web.
- BIS. "International convergence of capital measurement and capital standards - A revised framework." *Basel Committee of Banking Supervision Bank for International Settlements* 38-43 (2004). Print.
- Bot, M. and W. B. Langdon. "Application of genetic programming to induction of linear classification trees." *Proceedings of the 3rd European Conference on Genetic Programming*. Springer-Verlag, Berlin, Germany, 2000. 247–258. Print.
- Breiman, L. "Bagging Predictors." *Machine Learning* 24.301 (1996): 123–140. Print.
- Breiman, L., et al. *Classification and Regression Trees*. Chapman and Hall, 1993. Print.
- Breimann, L. "Random Forests." *Machine Learning* 45.1 (2001): 5–32. Print.

- Cao, J., et al. "CASTLE: A delay-constrained scheme for ks-anonymizing data streams." *Proceedings of the 24th International Conference on Data Engineering*. IEEE, 2008. 1376–1378. Print.
- Chandra, A. and X. Yao. "Ensemble learning using multi-objective evolutionary algorithms." *Journal of Mathematics Modeling Algorithms* 5.4 (2006): 417–445. Print.
- Chao, C., P. Chen, and C. Sun. "Privacy-Preserving Classification of Data Streams." *Tamkang Journal of Science and Engineering* 512.3 (2009): 321–330. Print.
- Chawla, N. and J. Sylvester. "Exploiting diversity in ensembles: improving the performance on unbalanced datasets." *Proceedings of the 7th International Conference on Multiple Classifier Systems*. MCS, Springer Verlag, 2007. 397–406. Print.
- Chen, H. and X. Yao. "Multiobjective neural network ensembles based on regularized negative correlation learning." *IEEE Transactions on Knowledge and Data Engineering* 22.12 (2010): 1738–1751. Print.
- Chhinkaniwala, H. and S. Garg. "Tuple Value based Multiplicative Data Perturbation approach to preserve Privacy in Data Stream Mining." *International Journal of Data Mining and Knowledge Management Process* 3.3 (2013): 53–61. Print.
- Chhinkaniwala, H., K. Patel, and S. Garg. "Privacy Preserving Data Stream Classification Using Data Perturbation Techniques." *Proceedings of International Conference on Emerging Trends in Electrical, Electronics and Communication Technologies*. 2012. Print.
- Clifton, C., et al. "Tools for privacy preserving distributed data mining." *ACM SIGKDD Explorations Newsletter* 4.2 (2004): 1–7. Print.
- Dam, H., et al. "Neural-Based Learning Classifier Systems." *IEEE Transactions on Knowledge and Data Engineering* 20.1 (2008): 26–39. Print.
- Davis, R. A., et al. "Novel feature selection method for genetic programming using metabolomic h nmr data." *Chemometrics Intelligent Laboratory Systems, Elsevier* 81.1 (2006): 50–59. Print.
- Domingos, P. and G. Hulten. "Mining high-speed data streams." *Proceedings of 6th ACM International Conference on Knowledge Discovery and Data Mining*. 2000. Print.

- Emekci, F., et al. "Privacy preserving decision tree learning over multiple parties." *Data and Knowledge Engineering* (2007): 348–361. Print.
- Enodu, T. and Q. Zhao. "Generation of comprehensible decision trees through evolution of training data." *Proceedings of Congress on Evolutionary Computation*. IEEE, Washington, USA, 2002. 1221–1225. Print.
- Espejo, P. G., S. Ventura, and F. Herrera. "A survey on the application of genetic programming to classification." *IEEE Trans. Syst. Man Cybern. Part C: Appl. and Rev.* 40.2 (2010): 121–144. Print.
- Ferrer-Troyano, F., J. Aguilar-Ruiz, and J. Riquelme. "Discovering Decision Rules from Numerical Data Streams." in *ACM Symposium on Applied Computing*. 2004. 649–653. Print.
- Finlay, S. "Multiple classifier architectures and their application to credit risk assessment." *European Journal of Operational Research* 210 (2011). Print.
- Florez-Lopez, R. and J.M. Ramon-Jeronimo. "Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal." *Expert Systems with Applications* 42 (2015): 5737–5753. Print.
- Folino, G., C. Pizzuti, and G. Spezzano. "Ensemble techniques for parallel genetic programming based classifiers." *Proceedings of 6th European Conference on Genetic Programming*. 2003. 56–59. Print.
- Folino, G., C. Pizzuti, and G. Spezzano. "Boosting technique for combining cellular GP classifiers." *Proceedings of the 7th European Conference on Genetic Programming (EuroGP-04)*, LNCS. Coimbra, Portugal: Springer Verlag, 2004. 47–56. Print.
- Follino, G., C. Pizzuti, and G. Spezzano. "Improving cooperative GP ensemble with clustering and pruning for pattern classification." *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York: ACM, 2006. 791–798. Print.
- Fortmann, S. "Understanding the Bias-Variance Tradeoff." 2012. <http://scott.fortmann-roe.com/docs/BiasVariance.html/>. Web.
- Friedman, A., R. Wolff, and A. Schuster. "Providing  $k$ -anonymity in data mining." *The International Journal on Very Large Data Bases* 17.4 (2008): 789–804. Print.

- Fung, B., K. Wang, and P. Yu. "Top-Down Specialization for Information and Privacy Preservation." *Proceedings of 21st International Conference on Data Engineering*. 2005. 205–216. Print.
- Fung, B., et al. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. 1st. Chapman & Hall/CRC, 2010. Print.
- Fung, B. C. M., K. Wang, and P. Yu. "Anonymizing classification data for privacy preservation." *IEEE Transactions on Knowledge and Data Engineering* 19.5 (2007): 711–725. Print.
- Fung, B. C. M., et al. "Anonymity for continuous data publishing." *Proceedings of the 11th International Conference on Extending Database Technology*. ACM, 2008. Print.
- Gaber, M., S. Krishnaswamy, and A. Zaslavsky. *On-board Mining of Data Streams in Sensor Networks*. Edited by S. Bandyopadhyay. Springer London, 2005. 307–335. Print.
- Gagne, C., et al. "Ensemble learning for free with evolutionary algorithms?" *Proceedings of Genetic and Evolutionary Computation Conference*. ACM, 2007. 1782–1789. Print.
- Gama, J. and C. Pinto. "Discretization from Data Streams: Applications to Histograms and Data Mining." *Proceedings of the 2006 ACM symposium on Applied computing*. 2006. 662–667. Print.
- Gama, J., R. Rocha, and P. Medas. "Accurate decision trees for mining high-speed data streams." *International Conference on Knowledge Discovery and Data Mining*. 2003. 523–528. Print.
- Godase, A. and V. Attar. "Classifier ensemble for imbalanced data stream classification." *Proceedings of ACM CUBE International Information Technology Conference*. 2012. Print.
- Golab, L. and T. Ozsu. *Data Stream Management*. San Mateo: Morgan and Claypool Publishers, 2010. Print.
- Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1989. Print.

- Greenwald, M. and S. Khanna. "Space-efficient online computation of quantile summaries." *ACM Special Interest Group on Management of Data Conference*. 2001. 58–66. Print.
- Han, J., M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. San Mateo: The Morgan Kaufmann Series in Data Management Systems, 2011. Print.
- Hartigan, J. and M. Wong. "A k-means clustering algorithm." *Applied Statistics* 28.1 (1979): 100–108. Print.
- HIPAA. "Health insurance portability and accountability act." 2004. <http://www.hhs.gov/ocr/hipaa/>. Web.
- Hoeffding, W. "Probability inequalities for sums of bounded random variables." *Journal of the American Statistical Association* 58.301 (1963): 13–30. Print.
- Huang, C., M. Chen, and C. Wang. "Credit scoring with a data mining approach based on support vector machines." *Expert System with applications* 37 (2007): 847–856. Print.
- Huang, C. L., M. C. Chen, and C. J. Wang. "Credit scoring with a data mining approach based on support vector machines." *Expert System with Applications* 37 (2007): 847–856. Print.
- Huang, Z., et al. "Credit rating analysis with support vector machines and neural networks: A market comparative study." *Decision Support System* 37 (2004): 543–558. Print.
- Hulten, G. and P. Domingos. "VFML a toolkit for mining high-speed time-changing data streams." 2003. <http://www.cs.washington.edu/dm/vfml/>, Accessed 31 March 2017. Web.
- Hulten, G., L. Spencer, and P. Domingos. "Mining time-changing data streams." *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*. 2001. Print.
- Hwanjo, Y., J. Xiaoqian, and J. Vaidya. "Privacy-Preserving SVM using Nonlinear Kernels on Horizontally Partitioned Data." *Proceedings of ACM SAC International Conference*. 2006. 603–610. Print.

- Iyengar, V. "Transforming data to satisfy privacy constraints." *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge discovery and Data Mining*. 2002. 279–288. Print.
- Jabeen, H. and A. R. Baig. "Review of classification using genetic programming." *International Journal of Engineering, Science and Technology* 2 (2010): 94–103. Print.
- Kaggle. "Give Me Some Credit Competition." 2011. <https://www.kaggle.com/c/GiveMeSomeCredit>, Accessed 31 March 2017. Web.
- Kantarcioglu, M. "Privacy-Preserving Data Mining: Models and Algorithms." Edited by C. Aggarwal and P. Yu. Vol. 34. Springer, 2008. Chap. A Survey of Privacy-Preserving Methods across Horizontally Partitioned Data. 313–336. Print. *Advances in Database Systems*.
- Kantarcioglu, M., J. Jin, and C. Clifton. "When do data mining results violate privacy?" *Proceedings of International Conference on Knowledge Discovery and Data Mining*. 2004. 599–604. Print.
- Khoshgoftaar, T. M., Y. Liu, and N. Seliya. "Genetic Programming-Based Decision Trees for Software Quality Classification." *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA. Print.
- Kirkby, R. "Improving Hoeffding Trees." Diss. Department of Computer Science, University of Waikato, 2007. Print.
- Kohavi, R. "A study of cross-validation and bootstrap for accuracy estimation and model selection." *Proceedings of International Joint Conference on Artificial Intelligence*. 1995. 1137–1145. Print.
- Komireddy, P. and T. Soule. "Orthogonal Evolution of Teams: A Class of Algorithms for Evolving Teams with Inversely Correlated Errors." *Genetic Programming Theory and Practice IV, Genetic and Evolutionary Computation Series*. Edited by Rick Riolo, Terence Soule, and Bill Worzel. Springer US, 2007. Print.
- Kotecha, R., V. Ukani, and S. Garg. "An empirical analysis of multiclass classification techniques in data mining." *Proceedings of the 2nd Nirma University International Conference on Engineering*. IEEE, Ahmedabad, 2011. Print.

- Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992. Print.
- Kuo, C. S., T. P. Hong, and C. L. Chen. "Applying genetic programming technique in classification trees." *Soft Computing - A Fusion of Foundations, Methodologies and Applications* 11 (2007): 1165–1172. Print.
- Law, Y. and C. Zaniolo. "An Adaptive Nearest Neighbor Classification Algorithm for Data Streams." *Proceedings of the 9th European Conference on the Principals and Practice of Knowledge Discovery in Databases*. Porto, Portugal: Springer Verlag, 2005. 649–653. Print.
- Lee, T., et al. "Mining the customer credit using classification and regression tree and multivariate adaptive regression splines." *Computational Statistics and Data Analysis* 50 (2006): 1113–1130. Print.
- Lee, T. S., et al. "Mining the customer credit using classification and regression tree and multivariate adaptive regression splines." *Computational Statistics and Data Analysis* 50 (2006): 1113–1130. Print.
- Lee, W. C. "Genetic programming decision tree for bankruptcy prediction." *Proceedings of the Joint Conference on Information Sciences*. Atlantis Press. Print.
- LeFevre, K., D. DeWitt, and R. Ramakrishnan. "Mondrian multidimensional  $k$ -anonymity." *Proceedings of 22nd International Conference on Data Engineering*. 2006. Print.
- Lessmann, S., et al. "Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research." *European Journal of Operational Research* 247 (2015): 1–32. Print.
- Li, J., B. C. Ooi, and W. Wang. "Anonymizing streaming data for privacy protection." *Proceedings of the 24th International Conference on Data Engineering*. IEEE, 2008. 1367–1369. Print.
- Li, N. and T. Li. "t-Closeness: Privacy Beyond  $k$ -anonymity and l-Diversity." *Proceedings of the 23rd International Conference on Data Engineering*. 2007. 106–115. Print.

- Lichman, M. "UCI machine learning repository." *University of California, Irvine, School of Information and Computer Sciences*. 2013. <http://archive.ics.uci.edu/ml>, Accessed 31 March 2017. Web.
- Lindell, Y. and B. Pinkas. "Privacy-preserving data mining." *Proceedings of 20th Annual International Cryptology Conference on Advances in Cryptology*. Springer Verlag, 2000. 436–454. Print.
- Liu, Y., X. Yao, and T. Higuchi. "Evolutionary Ensembles with Negative Correlation Learning." *IEEE Transactions on Evolutionary Computation* 4.4 (2000): 380–387. Print.
- Machanavajjhala, A., et al. "l-diversity: Privacy beyond  $k$ -anonymity." *ACM Transactions on Knowledge Discovery from Data* 1.1 (2007): 45–96. Print.
- Masud, M., et al. "Classification and novel class detection in concept-drifting data streams under time constraints." *IEEE Transactions on Knowledge and Data Engineering* 23 (2011): 859–874. Print.
- Muni, D. P., N. R. Pal, and J. Das. "A novel approach to design classifiers using genetic programming." *IEEE Trans. Evolut. Comput.* 8 (2004): 183–196. Print.
- Oka, S. and Q. Zhao. "Design of decision trees through integration of C4.5 and GP." *Proceedings of the 4th Japan-Australia Joint Workshop Intelligent and Evolutionary Systems*. 2000. 128–135. Print.
- Oreski, S., D. Oreski, and G. Oreski. "Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment." *Expert System with Applications* 39 (2012): 12605–12617. Print.
- Pei, J., et al. "Maintaining  $k$ -anonymity against incremental updates." *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*. IEEE, 2007. Print.
- Peng, Y., et al. "Privacy-Preserving Data Mining for Medical Data: Application of Data Partition Methods." Vol. 123. Springer Berlin Heidelberg, 2008. Chap. Communications and Discoveries from Multidisciplinary Data. 331–340. Print. Studies in Computational Intelligence.



- Rieckert, M., K. M. Malan, and A. P. Engelbrecht. "Adaptive genetic programming for dynamic classification problems." *Proceedings of the 11th conference on Congress on Evolutionary Computation*. IEEE Press, Norway, 2009. 674–681. Print.
- Rouwhorst, S. and A. Engelbrecht. "Searching the forest: Using decision trees as building blocks for evolutionary search in classification databases." *Proceedings of the Congress on Evolutionary Computation*. IEEE, California, USA, 2000. 633–638. Print.
- Samarati, P. "Protecting respondents' identities in microdata release." *IEEE Transactions on Knowledge Engineering* 13.6 (2001): 1010–1027. Print.
- Samarati, P. and 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): 188–206. Print.
- Samet, S. and A. Miri. "Privacy preserving ID3 using Gini index over horizontally partitioned data." *Computer Systems and Applications* (2008): 645–651. Print.
- Saraee, M. H. and R. S. Sadjady. "Optimizing classification techniques using genetic programming approach." *Proceedings of the 12th IEEE International Multi-topic Conference, INMIC*. 2008. 345–348. Print.
- Schapire, R. E., et al. "Boosting the margin: A new explanation for the effectiveness of voting methods." *Proceedings of Fourteenth International Conference on Machine Learning*. 1997. 322–330. Print.
- Shali, A., M. R. Kangavari, and B. Bina. "Using genetic programming for the induction of oblique decision trees." *6th International Conference on Machine Learning and Applications*. 2007. 38–43. Print.
- Suguna, N. and K. Thanushkodi. "An improved k-nearest neighbor classification using genetic algorithm." *International Journal of Computer Science Issues* 7.4 (2010): 1–8. Print.
- Sun, X., et al. "Enhanced P -Sensitive K Anonymity Models for Privacy Preserving Data Publishing." *Transactions on Data Privacy* 1 (2008): 53–66. Print.
- Sustersic, M., D. Mramor, and J. Zupan. "Consumer credit scoring models with limited data." *Expert Systems with Applications* 36 (2009): 4736–4744. Print.

- Sweeney, L. “ $k$ -anonymity: A Model for Protecting Privacy.” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (2002): 557–570. Print.
- Thomason, R. and T. Soule. “Novel ways of improving cooperation and performance in ensemble classifiers.” *Proceedings of Genetic and Evolutionary Computation Conference*. ACM, 2007. Print.
- Tian, H., et al. “A Knowledge Model Sharing Based Approach to Privacy-Preserving Data Mining.” *Transactions on Data Privacy* 5 (2012): 433–467. Print.
- Tsai, M. C., et al. “The consumer loan default predicting model An application of DEADA and neural network.” *Expert Systems with Applications* 36 (2009): 11682–11690. Print.
- Tsakonas, A., et al. “Bankruptcy prediction with neural logic networks by means of grammar-guided genetic programming.” *Expert Systems with Applications* 30 (2006): 449–461. Print.
- Twalaa, B. “Multiple classifier application to credit risk assessment.” *Expert Systems with Applications* 37 (2010): 3326–3336. Print.
- Verykios, V., et al. “State of the art in privacy preserving data mining.” *ACM SIGMOD Newsletter* 33.1 (2004): 50–57. Print.
- Wang, G., et al. “Two credit scoring models based on dual strategy ensemble trees.” *Knowledge-Based Systems* 26 (2012): 61–68. Print.
- Wang, H., et al. “Mining concept-drifting data streams using ensemble classifiers.” *Proceedings of 9th ACM International Conference on Knowledge Discovery and Data Mining*. 2003. Print.
- Wang, K., P. Yu, and S. Chakraborty. “Bottom-Up Generalization: A Data Mining Solution to Privacy Protection.” *Proceedings of 4th IEEE International Conference on Data Mining*. 2004. Print.
- Wang, T. and L. Liu. “Output Privacy in Data Mining.” *ACM Transactions on Database Systems* 36.1 (2011): 1–34. Print.
- Xiao, X. and Y. Tao. “M-invariance: towards privacy preserving re-publication of dynamic datasets.” *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. 2007. 689–700. Print.

- Xiong, L., S. Chitti, and L. Liu. "Mining multiple private databases using a kNN classifier." *Proceedings of ACM SAC International Conference*. 2007. 435–440. Print.
- Xu, Y., et al. "Privacy-Preserving Data Mining: Models and Algorithms." Edited by Aggarwal C. and Yu P. Vol. 34. Springer, 2008. Chap. Privacy-Preserving Data Stream Classification. 487–510. Print. *Advances in Database Systems*.
- Yeh, I. and C. Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." *Expert Systems with Applications* 36 (2009): 2473–2480. Print.
- Yeh, I. C. and C. H. Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." *Expert Systems with Applications* 36 (2009): 2473–2480. Print.
- Yehuda, L. and P. Benny. "Privacy preserving data mining." *In Proceedings of Conference on Advances in Cryptology*. Springer Verlag, 2000. 36–54. Print.
- Yu, L., S. Wang, and K. Lai. "Credit risk assessment with a multistage neural network ensemble learning approach." *Expert Systems with Applications* 34 (2008): 1434–1444. Print.
- Zhan, J. "Privacy-Preserving Collaborative Data Mining." *IEEE Computational Intelligence Magazine* 3.2 (2008). Print.
- Zhang, N., S. Wang, and W. Zhao. "A new scheme on privacy-preserving data classification." *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*. 2005. 374–383. Print.
- Zhou, B., et al. "Continuous Privacy Preserving Publishing of Data Streams." *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. 2009. 648–659. Print.
- Zhou, Z. *Ensemble Methods: Foundations and Algorithms*. 1st ed. Chapman & Hall, 2012. Print. Machine Learning and Pattern Recognition Series.
- Zhuojia, X. and Y. Xun. "Classification of Privacy-preserving distributed data mining protocols." *Proceedings of 6th International Conference on Digital Information Management*. IEEE, 2011. Print.