12-15-2016

Privacy Preserving Data Mining For Horizontally Distributed Medical Data Analysis

Yunmei Lu
Georgia State University

PRIVACY PRESERVING DATA MINING FOR HORIZONTALLY DISTRIBUTED MEDICAL DATA ANALYSIS

by

YUNMEI LU

Under the Direction of Yanqing Zhang, PhD

ABSTRACT

To build reliable prediction models and identify useful patterns, assembling data sets from databases maintained by different sources such as hospitals becomes increasingly common; however, it might divulge sensitive information about individuals and thus leads to increased concerns about privacy, which in turn prevents different parties from sharing information. Privacy Preserving Distributed Data Mining (PPDDM) provides a means to address this issue without accessing actual data values to avoid the disclosure of information beyond the final result. In recent years, a number of state-of-the-art PPDDM approaches have been developed, most of which are based on Secure Multiparty Computation (SMC). SMC requires expensive communication cost and sophisticated secure computation. Besides, the mining progress is inevitable to slow down due to the increasing volume of the aggregated data. In this work, a new framework named Privacy-Aware Non-linear SVM (PAN-SVM) is proposed to build a PPDDM model from multiple data sources. PAN-SVM employs the Secure Sum Protocol to protect privacy at the bottom layer, and reduces the complex communication and computation via

Nystrom matrix approximation and Eigen decomposition methods at the medium layer. The top layer of PAN-SVM speeds up the whole algorithm for large scale datasets. Based on the proposed framework of PAN-SVM, a Privacy Preserving Multi-class Classifier is built, and the experimental results on several benchmark datasets and microarray datasets show its abilities to improve classification accuracy compared with a regular SVM. In addition, two Privacy Preserving Feature Selection methods are also proposed based on PAN-SVM, and tested by using benchmark data and real world data. PAN-SVM does not depend on a trusted third party; all participants collaborate equally. Many experimental results show that PAN-SVM can not only effectively solve the problem of collaborative privacy-preserving data mining by building non-linear classification rules, but also significantly improve the performance of built classifiers.

INDEX WORDS: Privacy preserving, Distributed data mining, Classification, Feature selection, Support Vector Machine, Kernel matrix approximation and decomposition

PRIVACY PRESERVING DATA MINING FOR HORIZONTALLY DISTRIBUTED MEDICAL DATA ANALYSIS

by

YUNMEI LU

A Thesis/Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

PRIVACY PRESERVING DATA MINING FOR HORIZONTALLY DISTRIBUTED MEDICAL DATA ANALYSIS

by

YUNMEI LU

Committee Chair: Yanqing Zhang

Committee: Yi Pan

Rajshekhar Sunderraman

Yichuan Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

December 2016

DEDICATION

Thanks so much to my beloved parents for their love, encouragement over the years. To my beloved husband, Wenzhuo for his support! Thanks to my daughter Amy and my unborn little boy, they are my strength to work hard all the time!

ACKNOWLEDGEMENTS

This dissertation work would have no possible been finished without the guidance of my committee members, the help and supports from my group and friends. I would like to express my great gratitude to all of them.

I would like to show my deepest gratitude to my Ph.D. advisor, Dr. Yanqing Zhang, for his excellent inspiration and guidance, for his always generous support and encouragement, for his caring and patience. I would like to show my special thanks to my committee member, Dr. Yi Pan, who was willing to be my co-advisor and gave me much generous advice on my study and life. I would like to thank my committee members, Dr. Raj Sunderraman, who gave me a lot of generous support and guidance during my Ph.D. study at Georgia State University, and Dr. Yichuan Zhao, who taught me a lot from his statistics classes, gave me a lot of excellent advice for my projects.

I would like to show special thanks to Dr. Jenny Yang, who guided my first year for my Ph.D. program and I learned a lot from her group. I would also like to show my gratitude to Dr. Ying Xu at the University of Georgia, who was my advisor when I was at Jilin University, China, but gave me a lot of support and advice for my life in the USA.

I also would like to thank all of the professors and staffs at our department, especially Ms. Tammie Dudley, Ms. Adrienne Martin and MS. Venette Rice, thanks for their great assistance, which makes my study and life much easier and colorful. Thanks to my group members, Piyaphol Phoungphol and Yun Zhu, for their support and help. Thank all of my friends at Georgia State University and Atlanta; they made my Ph.D. life happy and colorful.

At last, I would like to acknowledge the continued financial support from the Computer Science Department and the Molecular Basis of Disease (MBD) fellowship at GSU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Background and Motivations

In recent two decades, data mining approaches have been widely used to analyze the massive amount of data, and they have become increasingly important tools to discover useful knowledge in many domains, such as medical data, consumer purchase data and census data. Assembling datasets maintained by different sources have become increasingly common, and applying data mining techniques on the aggregated datasets may build more reliable prediction models and attain useful patterns, which benefits for medical research, improving customer service and homeland security, and so forth.

However, this multi-data source system might divulge sensitive information about individuals. It thus leads to increasing concerns about privacy during the process of data mining, which in turn prevents different parties from sharing information. For examples, the Centers for Disease Control (CDC) may want to identify the trends of some disease to understand its progression via data mining techniques but has no relevant data. Insurance companies that have considerable data are unwilling to share these data due to patient privacy concerns. Another example is: a multinational corporation would like to mine its data for globally valid results, but national laws may prevent trans-border data sharing. Privacy Preserving Distributed Data Mining (PPDDM) provides a means to address this issue without accessing the actual data values to avoid the disclosure of information beyond the final results. Therefore it involves in great interests and has been studied extensively.

1.2 Definition of Privacy

To protect confidentiality and measure privacy, we have to define it. However, this is the hardest part, since privacy can mean different things to different people, at different

environments, in various contexts; and across different cultures. It is inevitable to get entirely different answers as you ask a selection of individuals what privacy is. Although the boundaries and content of privacy are different among individuals, groups and cultures, common principles should be the same. It is most common that individuals consider something inherently special or sensitive as privacy. The domain of privacy partially overlaps security, but privacy is not security, which can include the appropriate use and how to protect the individual information. According to the view of Ruth Gavison [1], the privacy can be defined in the term of access that others have to us, as well as our information. A general definition of privacy must to be one which is measurable of values and actionable.

The common definition [2] of privacy in the community of cryptography limits the information that is leaked by the distributed computation function, while information learned from the output regards as no-privacy leakage, since it is inevitable and designed by the secure computation function. For example, if two millionaires would like to know who is richer without telling the other his/her net worth. A secure computation function must return the result without revealing private information. Suppose one has $10,000,000 net worth, and he knows that he is richer from the function output. Therefore, he can learn that the net worth of the other one is less than $10,000,000, and this information leakage is inevitable.

In addition, privacy preserving is not only in the interest of individual but also to the public. On the other hand, privacy preserving is for the sake of both people and the society. Nowadays, many laws are issued to protect privacy, and various techniques are developed to prevent privacy from disclosure when using personal or public databases.

1.3 Popular Research Directions of Privacy Preserving Data Mining

A number of state-of-the-art techniques of privacy preserving data mining have been developed to leverage the privacy and mining issue, including classification, clustering, association rules and regression. Several key directions in this area are as follows:

Privacy preserving data publishing: These kinds of techniques tend to protect sensitive data information and privacy before data get published. Therefore, data obfuscation based techniques that are associated with privacy are studied, including randomly modify data, swap values between records and controlled modification of data to hide secrets. Among these methods, the most popular ones are randomization [3], k-anonymity [4] and l-diversity [5].

Changing mining results associated with privacy: In many cases, data mining results may comprise the privacy, summarization based methods have to be developed to expose only the needed facts and thus protect privacy from being revealing. The typical approaches are overall collection statistics and limited query functionality.

Cryptographic methods for distributed privacy: it has emerged significant interests in distributed data mining due to more and more available datasets on multi-site. Some data separation based methods are developed. Thus data can be held by an owner or a third party. In such case, a variety of cryptographic protocols usually needed to communicate with different parties. Secure Multiparty Computation (SMC) is a possible way to make it possible of distributed data mining without divulging sensitive information.

1.4 Models and Algorithms of Privacy Preserving Data Mining

Many methods for privacy preserving data mining employ data transformation techniques to protect privacy and sensitive data. The granularity of representation of data is usually reduced after transformation to mitigate the risk of divulging privacy, which results in the loss of

information or effectiveness of data management and data mining algorithms. However, it is inevitable and usually a trade-off between privacy and information loss. The top techniques used more frequently are as follows:

The randomization method: The randomization method is traditionally used to distort data by a probability distribution. In the case of privacy preserving, noise data are usually added to mask the value of original records, and then reconstruction techniques are needed to reconstruct the distribution of the original data. Normally noise is sufficiently large so that only the original distribution can be recovered but the values of each record. After distorting, the aggregated distributions are extended to data mining algorithms. The method of randomization can be described as follows: let X denote the original data records by $X = \{x_1, x_2, ..., x_n\}$, and then for each record of $x_i \in X$, a noise component y_i will be added, where y_i is drawn from the probability distribution of $f_Y(y)$, $\{y_1, y_2, ..., y_n\}$ are independent and identically distributed random variables. The new distorted records can be denoted by $Z = \{x_1 + y_1, x_2 + y_2, ..., x_n + y_n\}$. Since the probability distribution of Y is publicly known, and for the large number of n, the probability distribution of Z can be approximated by a number of techniques such as the kernel density estimation. Thus, it is possible to approximate the original probability of *X* by subtracting *Y* from the approximated distribution of, like *X = Z - Y*. In general, to make sure the original values of records cannot be guessed easily, only the original probability distribution of *X* can be approximated, it will assume that the variance of the noise data Y is large enough.

The *k*-anonymity model and *l*-diversity: the candidate key or combination of attributes can be used to identify individual records from public databases exactly. The *k*-anonymity model is developed to reduce the probability of being identified by candidate key and thus protect privacy. *K*-anonymity model reduces the granularity of data representation by using the techniques of

generalization and suppression. As a result, any given record will map onto at least k other records in the database. However, when there is the homogeneity of sensitive values within a group, *k*-anonymity cannot protect *k* individual records from being identified. Thus *l*-diversity model is developed to fix the weakness of k-anonymity.

Distributed privacy preserving data mining: when aggregate results needed to be obtained from multi-source databases, which might be collected and owned by multiple parties, it is common that performing privacy preserving data mining algorithms across distributed datasets. Suppose that the data can be represented in terms of a matrix ($m \times n$), where each row corresponds to an individual record or entry and each column corresponds to the attributes; then the data can be distributed among multiple sites as two typical ways:

- *Horizontally partitioned***:** individual records are distributed across multiple parties, and each of them has the data with all the same attributes, which can be represented in the form of a sub-matrix as $m_i \times n$, where $m_i < m$.
- *Vertically partitioned:* each party has the same set of entries, but the individual entries may contain different attributes, it can be represented by a sub-matrix of $m \times n_i$, where $n_i < n$.

The problem of Privacy Preserving Distributed Data Mining (PPDDM) overlaps closely with the field of cryptography for determining the secure multiple computation, which aims to design secure protocols to make sure those different parties, can perform joint computation by providing inputs without actual disclosure or sharing the individual inputs.

1.5 Organization

The remaining work is organized as follows: chapter one introduces the background and research motivations; chapter two lists the relevant work and literatures; chapter three proposes a privacy preserving framework for binary non-linear classification problem; chapter 4 proposes a

privacy preserving multi-class classification algorithm, chapter 5 and chapter 6 proposes two privacy preserving feature selection methods and chapter 7 concludes the work and future work.

2 RELATED WORK

2.1 Introduction

Data mining technology is widely used in many domains as a means of identifying useful knowledge, trends, and patterns from a massive amount of data. As the increasing available datasets, it has emerged more and more applications and needs for distributed data mining, where data are spread out across multiple parties. Thus it results in the growing concerns about privacy of data mining since it poses real privacy issues. For example, the public agency of the Centers for Disease Control (CDC) would like to analyze the health records via data mining techniques to identify patterns of some disease, and they need the data of patient disease and prescriptions from different insurance companies. The problem is insurance companies are unwilling to share their data due to privacy constraints or business interests. Since data mining is generally aimed at identifying patterns and producing some models rather than learning specific data, one solution to this privacy issue can be addressed by the Privacy Preserving Distributed Data Mining (PPDDM) model. The PPDDM model will not access the original data, but still perform data mining rules to get the desired mining results, thus the opportunity for misuse data will be reduced.

Therefore, we can define the privacy in the PPDDM model. No site should learn anything new from another site beyond the mining results. On the other hand, anything new learned in the process of mining must be derivable given one's data and the final result [1]. The principle of PPDDM, therefore, can be summarized that nothing can be learned from other data except the final mining results. To achieve this goal, we can either aggregate the data to a trust third party being analyzed in the third party, or we can use the Secure Multiparty Computation (SMC) to make data stay with the owner and be communicated securely among multiple parties. In case

that the trust third party is unlike to do the analyzing work or unable to analyze data by performing data mining algorithms, the latter strategy using SMC can meet the increasing needs in distributed data mining without disclosing the sensitive information of individuals.

2.2 Secure Multiparty Computation

Secure Multiparty Computation (SMC) is derived from Yao's Millionaires' problem [6], which states that two millionaires would like to know who is richer without telling each other their net worth. For simple, this problem can be restated by comparing two numbers, each held by one party, and either party is unwilling to disclose its number to the other. Yao presented one solution to this problem and generalized it to any efficiently computable functions restricted to two parties. There are two basic adversarial models in SMC:

Semi-Honest: in this model, the participants will follow the secure protocol, but keep curious and may attempt to dig some sensitive information from the received data from the other parties during the execution of the protocol.

Malicious: in this case, the participants may do anything to learn sensitive information, such as abort the protocol at any time, send sophisticated inputs to others, or send spurious messages and collude with other malicious parties.

The semi-honest model may seem questionable for preserving privacy if a party can be trusted to follow the secure protocol, why don't we trust them with the data? The following example can explain this. Consider the situation that several credit card companies would like to detect fraud via jointly building data mining models, and every business has been authorized to access that data. Once the data processing is completed, the data are supposed to be removed, since storing data brings the companies responsibility and cost to save the data. If there is a way to build data

mining models across distributed parties without actually accessing the original data, then they can save the responsibility and cost to protect the data from other parties other than their own.

However, no matter how secure the computation is, it is inevitable to leak some information. Still take the two millionaires as an example; once one party knows another party is richer or poorer, it can learn the upper bound or the lower bound of their net worth. In general, two kinds of information will leak, the information leaks from the secure computation function, and the information leaks from the computation process. Whatever is leaked from the former case, it is unavoidable as long as the function has to be computed. The latter case of information leakage during secure computation is provable prevented. Another key point is how to demonstrate that the security of the secure protocol used in the privacy preserving distributed data mining. It is common to restrict the secure against polynomial time adversary.

2.3 Secure Protocols

According to the SMC literature, the composition theorem [7] is a very useful theorem.

Composition Theorem for the semi-honest model: Suppose that g is privately reducible to f and that there exists a protocol for privately computing f. Then there exists a protocol for privately computing g.

The composition theorem states that if the sub-protocols are proved secure, then the entire protocol is secure. Therefore, if algorithms can be efficiently implemented on the sub-protocols, it can significantly improve the overall efficiency. Thus a lot of privacy preserving distributed data mining algorithms can be developed following the sub-protocols. These sub-protocols can be described using *homomorphic encryption* techniques [8]. Homomorphic encryption techniques allow operations such as search, comparison on encrypted data and obtain the same results as those based on plaintext data. Decryption becomes unnecessary during the whole

computing process. Thus data and computation do not need put in a third party, the risk of revealing information to other can be deduced. The following protocols only use homomorphic encryption, and all of them are secure in the semi-honest model with no collusion. According to the composition theorem, they can be combined to produce new privacy-preserving algorithms.

2.3.1 Secure Sum Protocol

In this secure protocol, the sum of values from each site will be securely calculated. Let *v* denote the sum and be represented as: $v = \sum_{i=1}^{s} v_i$, where *v* is known in the range [0…n]. In this secure sum protocol [9], one site will be assumed as a master site, numbered 1, and 2…s for the left sites. Normally, site 1 will uniformly generate a random number R in [0…n], adds it to its local value v_1, and then sends the sum of $R + v_1 \bmod n$ to site 2. Since R is chosen uniformly from [0…n], and then $R + v_1 \bmod n$ also distributes uniformly in this region. Thus site 2 learns nothing from this value. Site 2 receives this sum from site 1, and sends $S + v_2 \bmod n$ to site 3, where S is the sum received from site 1, and v_2 is its local value. In general, the site l receives: $v = R + \sum_{i=1}^{l-l} v_i \bmod n$. Since v is uniformly distributed, site l learns nothing from another site. It then computes the sum and passes it to next site by $v = R + \sum_{i=1}^{l} v_i \bmod n$.

The last site *s* also performs the above steps and sends the sum to site 1, since only site 1 knows the value of R, and then it can subtract R from this sum value to get the actual result. The details of how this method operates are introduced in [9]. This protocol is proved secure for the semi-honest model but faces a clear problem of leakage information if collusion exists. For example, if the site *l-1* and *l+1* collude, and tell each other the values they sent/received, they can determine the value at the site l.

2.3.2 Secure intersection

If several parties have their sets of items from a common domain, then the secure intersection protocol can be used to securely compute the cardinality of the intersection of these local sets. Given S parties having local sets of $L_1, L_2, ..., L_s$, we wish to compute securely $|L_1 \cap L_2 \cap ... \cap L_s|$. The protocol of secure intersection is very useful to help find common rules or frequent items etc. in data mining algorithms. The owner of the item will be protected without disclosure, [9] provides an efficient solution.

2.3.3 Secure Set Union

The Secure Set Union is useful in data mining if each party gives its rules, frequent item sets, decision tree, etc. without revealing the owner of the item. The union of items can be evaluated using SMC technology if the domain of the item is small. However, the data domains in data mining are usually enormous, so the secure set union protocol becomes inefficient. The description of this protocol and improved version can be referred from [10].

2.3.4 Dot Product Protocol

The sub-protocol or dot product is another important tool to compute the dot product of two vectors securely. Many secure dot product protocols have been proposed in [11, 12].

2.4 PPDDM algorithms on horizontally partitioned data

A distributed decision system that is capable of preserving the privacy of individual records can potentially address these issues of privacy. Recently there have been a number of attempts to solve the privacy-preserving problem in distributed data-mining applications; these include models built using decision trees [13-16], regression [17, 18] and the naive Bayes technique [19-21]. Some other methods have been built based on Support Vector Machines (SVM) [22-25]. The secure protocols mentioned in the above section and other protocols can be used in the

privacy preserving distributed data mining (PPDDM) algorithms on horizontally partitioned data. In each of the PPDDM algorithm, the functionality will be reduced to a computation of the secure protocols. Data at different locations can be horizontally distributed, which means that data records at different locations share common attributes, such as different banks collect data for their customers. Data can also be vertically distributed, that is to say different sites have same records but different attributes. Such as bank, insurance company and auto insurance company collect different information about same people.

2.4.1 Classification

Decision tree*:* The cryptographic technique is used in [11, 12] to protect privacy and for the first time to be employed to construct decision trees. The goal of this work is to securely build an ID3 decision tree based on the data that are horizontally partitioned between two parties. This work employs the secure log algorithm, secure polynomial evaluation, and secure comparison sub-protocols to compute the decision conditional entropy securely. [13, 14] proposed an alternative approach named DIDT (Distributed Id3-based Decision Tree), which uses the statistics of the values of an attribute among classes from multiple hospitals to build a global cross-table matrix, which is then used to build a decision tree.

Support Vector Machine (SVM): SVM is another important classification technique and has been widely employed in main domains. To build SVM, the kernel matrix is needed with $G_{ij} = x_i \cdot x_j$, to securely calculate the dot product for all pairs of training data. The Privacy Preserving Support Vector Machine (PP-SVM) [22] used secure dot protocol to preserve individual information from being revealing. PP-SVM can be applied to the non-linear classification of a horizontally-partitioned dataset, but it requires a trusted intermediary to construct the actual SVM, which may restrict the preserving of patients privacy. The Distributed

Privacy Preserving Support Vector Machine (DPP-SVM) method proposed in [23] allows for privacy-preserving collaborative learning by employing a trusted server; however, DPP-SVM just supports linear kernel SVM and only deals with vertically partitioned data – that is, data distributed by data field rather than by patient. Also, DPP-SVM and PP-SVM may become vulnerable when the third party is not trustworthy. Traditional SVMs rely on a centralized dataset to which they have unrestricted access, so sets of partial datasets distributed among several sites create a substantial barrier for these systems in privacy-sensitive scenarios.

Naïve Bayes Classification: A naïve Bayes classification is applied to the distributed car evaluation datasets in [17], where the private information of customers is preserved. To protect the privacy of data, a trusted third party has been used here. The decision-making systems mentioned above, try to improve the classification accuracy by using multiple insufficient datasets without leaking the actual data of the participating sites. However, few of them are practically used in biomedical applications, and some of the collaborative environments are not easy to use.

2.4.2 Clustering

Clustering is a well-studied data mining technique which aims at grouping similar data points together into a cluster. In the k-means clustering method, k initial cluster centers are chosen firstly, and then are updated iteratively. [26] shows that k-means clustering can be implemented on arbitrarily partitioned data via the SMC protocols of the secure dot product, secure summation, and secure comparison. Similarly, [27] employs secure sum protocol in the expectation maximization method for horizontally partitioned data.

2.5 Limitations of PPDDM

Privacy-preserving distributed data mining techniques address many sophisticate approaches aiming to fix the dilemma between information sharing and privacy concerns. However, privacy if not free! Many of the PPDDM algorithms need the assistance of expensive cryptographic operations. Furthermore, protocols that are secure against malicious parties are even more expensive. Parameters that used in distributed data mining protocols need to be set very carefully to avoid an explosion computation. In addition, aggregating data together increase the volume of data for being mined, which brings a challenge to data mining algorithms. Therefore, efficient algorithms are in need to be developed to conquer the expensive cryptographic computation, as well as expensive computation for the bigger aggregated datasets.

Although secure multiparty computation (SMC) provides distributed data mining a means for information sharing without actually reveal individual information, compared to noise addition method, the cryptographic techniques for PPDDM is lack of the flexibility of trade-off between privacy and accuracy. In the noise addition method, the variance of added noise data can be used as a parameter to adjust the information loss and increase the privacy. As a result, new algorithms are needed for PPDDM to a tradeoff between privacy and accuracy.

PPDDM algorithms are developed to make a global decision system via revealing nothing other than the final result; however, not revealing anything may be overkill in some case. In some situation, revealing some information, such as the summarized information may not a privacy breach, while the techniques used to protect such information may increase the computation burden, hence slow down the whole process. Therefore, methods that leverage the different levels of privacy and efficiency will be developed.

This work addresses the three aspects of the limitations of PPDDM and aims to design a novel privacy preserving framework to improve the PPDDM algorithms in the applications of classification, based on which new data mining algorithms can then be developed.

3 PRIVACY PRESERVING NON-LINEAR SVM FRAMEWORK

3.1 Introduction

Over the past two decades, advances in data collection and storage technologies resulted in data explosion in every scientific domain. Machine learning methods become increasingly important tools to analyze massive amounts of data to discover useful knowledge in many applications [28-31]. In some domains, especially the detection of medical condition, decision must be made efficiently and reliably. However, training a classification model with a small dataset or a specific group of data may lead to an unreliable or inaccurate decision. Unfortunately, it is very common that local datasets of a certain diagnosis in some hospitals do not have sufficient records due to the difficulty and cost in data acquisition. In such a case, multiple volumes of distributed data need to be combined to yield strengthened capabilities for diagnosis prediction. Consequently, mining collective distributed similar databases from multiple sources or hospitals can possibly lead to a reliable decision-making model with higher accuracy.

However, the distributed mining can result in issues in protecting the privacy of patients and the security of distributed datasets, which in turn restricts the free sharing of data due to the potential risk of divulging patient privacy [32-35]. Moreover, some data cannot be shared due to legal and commercial reasons. For example, the laws of HIPAA [36, 37] in the United States require that medical data cannot be released without appropriate authorization. Therefore, methods for building a global decision model based on distributed insufficient datasets should be developed to improve the classification accuracy and decision reliability without revealing the privacy of datasets at the same time.

Numerous efforts were devoted recently to solve the privacy-preserving problem with reliable diagnosis based on distributed data, including models built using decision trees [13-16],

regression [17, 18] and naïve Bayes [19-21]. The cryptographic technique is used to protect privacy and to construct decision trees for the first time in [13, 14]. An alternative approach named DIDT (Distributed Id3-based Decision Tree) is proposed, which uses the statistics of the values of an attribute among classes from multiple hospitals to build a global cross-table matrix for subsequently building decision tree [15, 16]. A privacy preserving linear regression model is presented to address the important tradeoff between global statistical analysis and privacy [18]. In literature [17], the regression model is applied to the full statistical analysis of the combined database without actually combing the distributed databases. A naïve Bayes classification is applied to the distributed car evaluation datasets in [19], where the private information of customers are also preserved. To protect the data privacy, a trusted third party has been used here. The decision-making systems mentioned above try to improve the classification accuracy by using multiple insufficient datasets without leaking the actual data of the participating sites. However, few of them were practically used in biomedical applications. Moreover, some of the collaborative environments are difficult to use as well.

Support Vector Machine (SVM) is one of the top 10 widely used tools for decision support [38]. The traditional model of SVM is built on a centralized repository of the dataset with free access to the dataset, while the distributed hospital datasets create a substantial barrier for researchers to build an efficient SVM classification model when privacy matters. Several privacy-preserving concern classification models based on SVM are developed in [22-25]. The Distributed Privacy Preserving Support Vector Machine (DPP-SVM) method proposed in [23] enables the privacy-preserving collaborative learning by employing a trusted server to integrate "privacy-insensitive' intermediary results. The global model of DPP-SVM guarantees that the decision result is the same as that learned from combined data. However, DPP-SVM just

supports linear kernel SVM and only deals with the vertically partitioned data, in which patients with similar features are not distributed while the features of patients are distributed. Another privacy-preserving solution is also proposed for supporting vector machine classification (PP-SVM) on horizontally partitioned data [22]. PP-SVM constructs a global classification model from distributed multiple parties with data represented by binary feature vectors. The gram matrix composed of the dot products of every data pair is computed using the secure set intersection cardinality to obtain the kernel matrix and then the global SVM model without disclosing any data. Though PP-SVM works for non-linear kernels and horizontally partitioned data, an untrusted intermediator is required to construct SVM, which may restrict the preserving of patients privacy.

A distributed decision system based on multiple datasets that can preserve privacy effectively and make decision efficiently will become attractive. The current work aims to develop such a system by proposing a privacy preserving framework. In this framework, we also employ SVM as a classifier to make a decision based on horizontally partitioned data from multiple parties. There is no trust third party needed here; data will be encrypted via the *Secure Sum Protocol*, as mentioned in subsection 2.3.1. The party who wants to do distributed data mining will firstly send his encoded data to his next neighbor and then receive the complete encrypted data from the last participator. After this encrypting and collecting procedure, the party can start his data mining work. Since the kernel matrix in SVM involves lots of computation and memory space by calculating the dot product of any pair of data, including pairs of data in different locations, we employ the Nystrom approximation technique [39] to approximate the kernel matrix, thus reduce the computation and communication cost. Besides, the k-means clustering method [40] is used to select landmarks for Nystrom approximation, and thus the private information can further

be protected, details can be found from the subsection of 3.3. Moreover, to make the proposed framework feasible to non-linear SVM, we employ a kernel matrix decomposition method [41] to convert the non-linear separable SVM into a linear one. Thus a global linear classification model can be conducted from multisource data. In addition, the cutting-plan technique [42] is used to accelerate the training process of SVM. Consequently, the framework designed in this work cannot only solve the collaborative problem of privacy-preserving by building a global classification model via kernel approximation and decomposition, but also work for the non-linear pattern. Moreover, costs for complex computation and unnecessary communication are avoided, and the training process is shortened; as a result, this framework is also feasible for dealing with the large scale of data sources.

The proposed framework named Privacy-Aware Non-linear SVM (PAN-SVM in brief) is designed as a high-perform PPDDM framework. It is tested on 12 different datasets, and the results show that the proposed framework of PAN-SVM can efficiently achieve the privacy preserving distributed data mining with competitive classification accuracy compared with the results from a single dataset; details are introduced in the following subsections.

3.2 Methods

3.2.1 Support Vector Machines

Support vector machines (SVMs) are state-of-the-art classification methods firstly introduced by Vapnik et al. [38]. They have been widely used in many fields due to their high accuracy and their ability to deal with high-dimensional data. For a binary classification problem, given a training dataset D of n samples $\mathrm{D} = \{ (x_i, y_i) \mid i = 1...n \}$, where $x_i \in R^p$ is a sample with p features and $y_i \in \{-1, 1\}$ is the class label of the sample x_i. SVMs construct hyperplanes that separate the two classes in the training data. The optimal hyperplane will maximize the margin of separation

while minimizing classification errors and the optimization problem can be formulated as in equation (3.1):

$$\min_{w,\xi_i,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ \forall i, y_i(w\cdot x_i + b) \geq 1-\xi_i, \quad \xi_i > 0 \tag{3.1}$$

Where **w** is the weight vector of a hyperplane $f(x) = wx + b$, and b is bias, C is the penalty parameter and ξ is slack variable. In the case that the two classes cannot be linearly separated, classification can still be performed by mapping the data from the original space into a higher dimensional space; we then attempt to select a mapping function such that the data are separable in the higher dimensional space. However, it can be very difficult to select an appropriate mapping, due in part to the huge number of dimensions. Fortunately, the "SVM problem" presented in equation (3.1) has a corresponding dual form, which is formulated by equation (3.2):

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left\langle \varphi(x_i), \varphi(x_j)\right\rangle - \sum_{i=1}^{n}\alpha_i$$
$$s.t. \sum_{i=1}^{n} y_i\alpha_i = 0 \ \ and\ \forall i, 0 \leq \alpha_i \leq C \tag{3.2}$$

In this dual form, only the dot products between pairs of inputs $\varphi(x_i)\cdot\varphi(x_j)$ are required. It is much easier to define a satisfactory function $K(x_i, x_j) = \varphi(x_i)\cdot\varphi(x_j)$ than it is to apply the mapping function to both inputs and then calculate the dot product on the transformed data points. The function *K* is called a *kernel function*. There are a number of kernel functions in common use, including the radial basis function (RBF) and the polynomial kernel.

By substituting the kernel function *K* into equation (3.2), the dual objective function can be rewritten as in (3.3):

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n}\alpha_i \tag{3.3}$$

For a single data source with a given kernel function, it is possible to calculate a kernel matrix *K* for each pair of data points; however, this is not feasible when using multiple data sources that do not share data, since for a pair of data points x_i and x_j, it is not possible to compute $K(x_i, x_j)$ if x_i and x_j reside in different data sources.

3.2.2 Proposed Framework

The main structure diagram of the proposed Privacy-Aware Non-linear SVM (PAN-SVM) [43] for distributed data sources is shown in Figure 3.1. PAN-SVM consists of three layers: the bottom layer preserves privacy via encrypting protocol to protect local data and make them invisible to other parties; the medium layer approximates kernel matrix and converts the global non-linear SVM model into a linear one, and thus lots of computation are reduced and it makes PAN-SVM be feasible to with non-separable data; finally, the top layer accelerates the training process of the linear SVM model that receives from the medium layer. The working details and techniques used in each component layer of PAN-SVM are described in the following subsections.

3.2.2.1 Bottom Layer: Privacy Preserving

In this layer, the security of the data is guaranteed by the *Secure Sum Protocol* proposed in [9]. Suppose there are three or more data sources, site 1 uniformly generates a number $X \sim uniform[1, S]$, adds it to its local value v_1, and sends the sum $X + v_1 \bmod S$ to site 2. Since *X* is uniformly selected from *1*~*S*, $X + v_1 \bmod S$ is also uniformly distributed in the range of *1*~*S*. Therefore, site 2 knows nothing about the local value of site 1. Site 2 receives the sum and adds it to its local data v_2 mod *S*, and then passes the new sum to the next site without disclosing its

local values, and so on, until to the first site. Since the sum is always uniformly distributed in *1* ~ *S*, each site cannot learn the preserved privacy information from the previous sites. In this layer, the encrypted data that will be sent to the medium layer are called landmarks [39], which are used to approximate the kernel matrix in the medium layer.

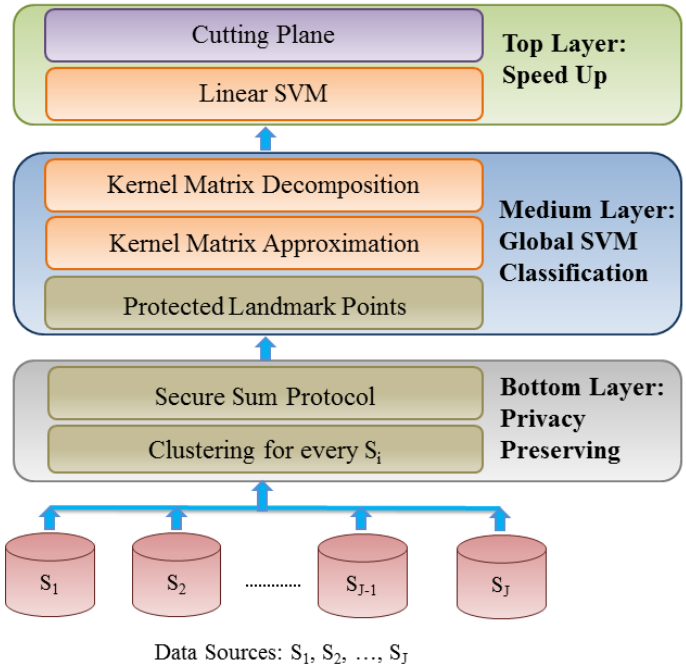


Figure 3.1. Proposed framework of PAN-SVM.

To build the proposed global SVM classification model, the low-rank Nystrom method [39] is used to approximate the kernel matrix in the medium layer. Since the quality of Nystrom approximation highly depends on landmarks, many sampling schemes [39, 44-46] are proposed to select the best landmarks. Among those state-of-the-art sampling approaches, [44] shows that the k-means clustering method can achieve significant performance and provide a low approximation error bound; helpfully, the k-means clustering algorithm is also simple to implement. Therefore, the k-means algorithm is adopted in every local dataset in the system. The data centers that are selected by k-means at each single site are treated as landmarks, which are encrypted and sent to the medium layer. Compared with sending all data to the initiator, only a

small size of landmarks are needed, a large number of costly ommunication is then alleviated. Besides, the data centers selected by k-means clustering method, not data samples themselves are used to approximate kernel matrix; individual private information can further be protected.

3.2.2.2 Medium Layer: Building a Global Classification Model

Supposes there are total l landmarks are selected from all the data sources, and once they are transferred to the medium layer, the low-rank Nystrom approximation technique [44] and kernel matrix decomposition method are adopted to build a global linear classification model based on the landmarks.

Kernel Matrix Approximation: the Nystrom method randomly picks l global landmark points, named a set of L, from all data sources, and then infers the kernel value of $K(x_i, x_j)$ implicitly from the relations of x_i and x_j and with these landmarks. Let R_i be a $1 \times l$ vector that contains kernel values between x_i and L respectively: $R_i = [K(x_i, L_1), K(x_i, L_2), \ldots, K(x_i, L_l)]$ and similarly, $R_j = [K(x_j, L_1), K(x_j, L_2), \ldots, K(x_j, L_l)]$; finally, let A be the $l \times l$ kernel matrix between any pair of l andmarks. Then, $K(x_i, x_j)$ can be approximated by equation (3.4):

$$\mathrm{K}(x_i, x_j) = R_i A^{-1} R_j^T \tag{3.4}$$

By approximation using the Nystrom method, the kernel values between any pair of samples can by replaced by equation (3.4). There is one main disadvantage of using standard Nystrom approximation; that is landmarks are sampled from the original data, therefore, sending these points directly to other data sources will increase the risk of divulging privacy even though under the protection of secure protocol. In this system, the secure sum protocol is used to solve the privacy issue and the k-means clustering method used for selecting landmarks further masks the original data.

Kernel Decomposition: The kernel decomposition technique used here is proposed in [41], which attempts to convert the dual form of the SVM problem in equation (3.1) back to its primal form of equation (3.2). Zhang et al. in [41] show that the kernel matrix *K* can be decomposed into the form of $K = FF^T$. If a kernel matrix of *n* samples can be decomposed into FF^T, where F is a $n \times m$ matrix, then F can be treated as *virtual inputs* for a linear SVM model by mapping X from the original higher *p*-dimensional space into a much lower *m*-dimensional space, $p \gg m$. Equation (3.4) can then be rewritten in a general form as in equation (3.5):

$$K = RA^{-1}R^T = R(U\Lambda U^T)^{-1}R^T = R(U^T\Lambda^{-1}U)R^T \tag{3.5}$$

Here "A" is an $l \times l$ symmetric and positive semi-definite matrix; thus Eigen-decomposition of A can be expressed as $A = U\Lambda U^T$, where U and Λ are the eigenvectors and eigenvalues of A, respectively. If *K* is decomposed into a $K = FF^T$ form, it is obvious that F can be approximated as in equation (3.6):

$$F = RU\Lambda^{-1/2} \tag{3.6}$$

It is interesting to note in equation (3.6) that it is not necessary to calculate any pair of the kernel values $K(x_i, x_j)$ across data sources at all. Only the kernel value between each data point and the chosen landmarks need to be calculated, which can then be mapped on to the eigenvectors of the landmarks. Since approximating all pairs of (x_i, x_j) that are located at different locations requires a large amount of communication among data sources, which does not scale well when the number of data or data sources is significant. Since only small sizes of samples are used to approximate the kernel matrix, a large number of complex communication and computation cost are avoided. Moreover, the non-linear SVM is converted into a linear one by the Nystrom approximation and matrix decomposition techniques with the kernel matrix $K = FF^T$, where F can be regarded as virtual points. Thus the global "*linear"* classification

model has been constructed with virtual points of $F = RU\Lambda^{-1/2}$. This procedure is described in Algorithm 1.

Algorithm 1 Convert Non-linear to Linear Space

Input:

X : training data in multiple sources.
ℓ : the number of landmarks.
k : kernel function.

1: $L \leftarrow global\_cluster\_center(X, n\_cluster = \ell)$
2: $R \leftarrow k(X, L)$
3: $A \leftarrow k(L, L)$
4: $U\Lambda U^T \leftarrow SVD(A)$
5: $F \leftarrow R\, U\Lambda^{-1/2}$
6: return F

Figure 3.2. Matrix Decomposition

3.2.2.3 Top Layer: Accelerating Training Process

After converting all data from their representation in the non-linear space into virtual points in the final linear space, a linear SVM model is built. To further improve the efficiency of the proposed model, the *cutting-plane technique* introduced by Franc et al. [42] is used here. This approach can not only produce a linear SVM from large-scale data efficiently, but also be easily applied to the problem of preserving privacy when multiple data-sources are in use.

Accelerate SVM with Cutting-Plane Technique: Traditionally, training SVM from a large dataset via equation (3.1) is a rather difficult task, because the size of the equation expands as the dataset expands: it has *n* slack variables ξ_i and *n* constraints, where *n* is the number of samples. To address this, the *cutting–plane* technique eliminates all slack variables by replacing them with a single variable L, which is a summation of all ξ_i'*s*. However, this results in 2^n constraints: the combinations of *n* constraints in (3.7) and the predicted values for *n* data points. For a point *i*, $c_i = 0$ if this point is correctly classified, and $c_i = 1$ otherwise. The resulting new problem is then formulated in (3.7):

$$\min_{w,b,R} \frac{1}{2}\|w\|^2 + CL$$
$$s.t.\ \ \forall C \in \{0,1\}^n \quad \sum_{i=1}^{n} y_i c_i (w \cdot x_i + b) \geq \sum_{i=1}^{n} c_i - L \tag{3.7}$$

In practice, equation (3.7) can be solved easily with a smaller subset of 2^n constraints, starting by removing all constraints, and then iteratively adding the most violated constraint back. The optimal solution will be found within a few iterations. The process of finding the optimal solution can be formally defined in Algorithm 2.

Algorithm 2 Distributed Cutting-Plane to Solve Linear SVM

Input:

f_i, y_i : virtual inputs and labels, $i = 1, \ldots, n$.
S_j : data sources, $j = 1, \ldots J$
ϵ : tolerance

$w = \vec{0};\ \Omega = \varnothing$
repeat
  // distributed
  for all S_j **do**
    for $i \in S_j$ **do**

$$c_i = \begin{cases} 1 & \text{if } y_i(w \cdot f_i) < 1 \\ 0 & \text{otherwise} \end{cases}$$

    end for
    $m_j \leftarrow \sum_{i \in S_j} c_i\ ,\ a_j \leftarrow \sum_{i \in S_j} c_i y_i f_i\ ,\ d_j \leftarrow \sum_{i \in S_j} c_i y_i$
  end for

  // centralized
  $\Omega \leftarrow \Omega\ \cup\ \{w \cdot \sum_{j=1\ldots J} a_j + b \sum_{j=1\ldots J} d_j \geq \sum_{j=1\ldots J} m_j - L\}$

  $\{w, b, L\} \leftarrow \text{argmin}_{w,b,L \geq 0}\ \frac{1}{2}\|w\|^2 + C\ L$

  s.t. $\forall\ \Omega$

until $\left| w \cdot \sum_{j=1\ldots J} a_j + b \sum_{j=1\ldots J} d_j - \sum_{j=1\ldots J} m_j + L \right| < n\epsilon$

return $\{w, b\}$

Figure 3.3. Algorithm for speeding up PAN-SVM.

The cutting plane technique also works well for the proposed distributed classification model. First, the virtual data points derived from the Nystrom low-rank approximation and decomposition techniques have a smaller size, which can be solved effectively in a few iterations, regardless of the size of the dataset. Second, in each iteration, the data sources only need to compute two parameters for a given value of *w*, in line 7 of Algorithm 2 in Figure 3.3.

The proposed model effectively decreases the size of the global classification through matrix approximation and decomposition techniques; therefore, the iteration number also significantly decreases. In the following subsection, we will give a brief description of the Optimized Cutting Plane Algorithm (OCA) technique, proposed by Franc et al. in [42], which accelerates the converge process. OCA shows that the number of iterations required to converge to a stop criterion is approximately linear in the sample size.

Linear Search: Franc et al. in [29] proposed an Optimized Cutting Plane Algorithm (OCA) to improve the convergence rate of the optimizing process of equation (3.7) by ensuring that the new constraint added in each iteration will lead to the lower objective. Originally, Algorithm 2 will use the new *w* derived from line #10 to create a new constraint in line #4-9. On the other hand, OCA will keep the value of *w* before and after line #10 as w_b and w_a, respectively. Then, it will search for the optimal *w* in a (w_a - w_b) direction that has the minimum objective value. The new constraint created from this ***w*** guarantees that the iteration number required by OCA decreases.

$$\partial obj(w) = \frac{1}{2}\left\|w_b + k(w_a - w_b)\right\|^2 + C\sum_{i=1}^{n} c_i y_i f_i \cdot (1 - (w_b + k(w_a - w_b))) \tag{3.8}$$

$$\frac{\partial obj(w)}{\partial k} = k\left\|w_a - w_b\right\|^2 + w_b \cdot (w_a - w_b) - C\sum_{i=1}^{n} c_i y_i f_i \cdot (w_a - w_b) \tag{3.9}$$

Algorithm 3 Linear Search for Distributed Data Sources

Input:

f_i, y_i : virtual inputs and labels, $i = 1, \ldots, n$.
S_j : data sources, $j = 1, \ldots J$
w_b : w before line#10 of Alg 2
w_a : w after line#10 of Alg 2 Statex λ : search's step size

1: $\mathcal{D} \leftarrow w_a - w_b;\ k \leftarrow 0$
2: **while** true **do**
3: $w \leftarrow w_b + k\ \mathcal{D}$
// distributed
4: **for all** S_j **do**
5: **for** $i \in S_j$ **do**
6: $$c_i = \begin{cases} 1 & \text{if } y_i(w \cdot f_i) < 1 \\ 0 & \text{otherwise} \end{cases}$$
7: **end for**
8: $a_j \leftarrow \sum_{i \in S_j} c_i y_i f_i$
9: **end for**
// centralized
10: $\frac{\partial obj(w)}{\partial k} \leftarrow k \|\mathcal{D}\|^2 + w_b \cdot \mathcal{D} - C\ \mathcal{D} \cdot \sum_{j=1 \ldots J} a_j$
11: **if** $\partial obj(w)/\partial k \geq 0$ **then**
12: **break**
13: **end if**
14: $k \leftarrow k + \lambda$
15: **end while**
16: return w

Figure 3.4. Linear search.

Generally, w can be defined as $w \leftarrow w_b + k(w_a - w_b)$, where $k \geqslant 0$. The objective of value w can be written as where c_i equals 1 when point i is misclassified by w and 0 otherwise. OCA searches for the optimal point by investigating in (3.8) and (3.9) when it changes from negative to positive. However, adopting the original OCA technique in the current scenario is not straightforward because it performs an extensive search by checking all possible k's corresponding to individual data points. This process requires sharing data information among data sources. Instead, we propose to do a linear search with a constant step-size, λ. The search will start from w_b and try $w_b + \lambda(w_a - w_b), w_b + 2\lambda(w_a - w_b), \ldots$, until the value of $\partial obj(w) / \partial k$ changes from negative to positive. If the derivation at w_b is equal or greater than 0, then w_b is the optimal solution for the problem. This simple search will avoid sharing data among data sources

while still speeding up the process. The linear search for privately distributed data sources is defined in Algorithm 3 as shown in Figure 3.4.

3.3 Experimental Results and Discussions

This section presents the experimental results obtained by PAN-SVM with real data and compares the performance of PAN-SVM with other existing techniques in common use. The experiments are conducted using MATLAB by simulating multiple data sources. All of the following algorithms are either implemented in pure MATLAB (that is to say, no .mex files), or by using MATLAB's Statistics Toolbox [47].

3.3.1 Datasets

12 real world datasets gathered from different problem domains are used here for testing PAN-SVM; the datasets are listed in Table 3.1. The *Pima Diabetes* and *Adult* datasets are downloaded from the University of California, Irvine (UCI) Machine Learning Repository [48], and the microarray dataset *GSE2990* is from the Gene Expression Omnibus (GEO) [49]; all others are from the repository of LIBSVM [50]. C is the penalty parameter for SVM and γ is a free parameter RBF kernel function. They are generated by 10-fold cross validation.

Table 3.1: Summary of datasets used for testing PAN-SVM.

Dataset	**# of Features**	**# of Samples**	**C**	**γ**
Australian	14	690	512.0	0.0078125
Breast cancer	10	683	0.125	0.125
Pima Diabetes	8	768	512.0	0.0078125
German	24	1,000	8.0	0.03125
Heart	13	270	2048.0	0.0001220703125
Ionosphere	34	351	8.0	0.5
Liver disorders	6	345	8.0	0.5
Splice	60	3,175	8.0	0.5
Fourclass	2	862	8.0	0.03125
Adult	123	32561	100.0	0.5
Cod_rna	8	59535	32.0	0.5
GSE2990	11119	183	32.0	0.0000305

3.3.2 Effectiveness

The effectiveness of PAN-SVM is assessed on the 12 datasets described in Table 3.1 from three aspects as follows: 1) selection of landmarks, 2) comparison with existing single-dataset classifiers, and 3) comparison with existing distributed classifiers.

3.3.2.1 Selection of Landmarks

To simulate the multiple data sources that PAN-SVM is designed to work with, the data are randomly split into s equally sized groups (suppose there are $s = 5$ data sources here); a constant percentage of each dataset will be selected as landmarks. For example, in the *Australian* dataset, if 15% of the samples in the original dataset are selected as landmarks, then each of the 5 subsets contains 20 landmark points (floor $((690/5) * 0.15) = 20$).

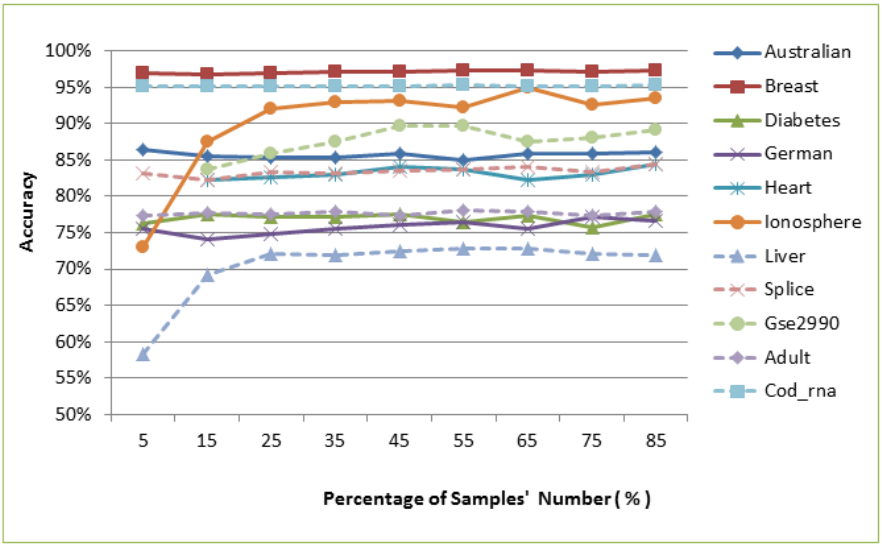


Figure 3.5. Classification accuracy varies as different numbers of landmarks.

The Nystrom approximation method depends on the landmarks, but how many landmarks should be used to approximate the kernel matrix? Several tests are conducted on all of the datasets except the *Four-class* data to specify this question, as shown in Figure 3.5, from which

it can observe that as the percentage of landmarks varies from 5% to 85%, the classification accuracy of PAN-SVM does not vary dramatically (~3%). Based on these results, we can observe that PAN-SVM can achieve the highest classification accuracy when selecting 15% ~ 35% samples from the data records as landmarks. Unlike the classification accuracy, the average training time increases as the number of selected landmarks increases (details are not shown here). Therefore, the number of landmarks can be selected varying from 15% to 35% of the original data size to tradeoff the loss of classification accuracy and the training efficiency of the classifier.

3.3.2.2 Comparison with Single-dataset Classifiers

We also evaluate our approach via eight small datasets, namely, *Australian, Breast cancer, Pima Diabetes, German, Heart, Ionosphere, Liver disorders and Splice*. The experimental results of classification accuracy are compared with those from a number of traditional classification models, such as Naïve Bayes, Decision Tree, LIBSVM with linear kernel function and LIBSVM with RBF kernel function, as shown in Table 3.2, which shows the performance comparison between PAN-SVM and other traditional classifiers on a single dataset. Note that these traditional models operate on a single integrated set of training data, which is unlike PAN-SVM. Two separate PAN-SVM models with RBF kernel are trained based on 15% and 25% of samples as landmarks, respectively. 5-fold cross-validation is used for each dataset. It can observe from Table 3.2 that both of the two PAN-SVM models can yield almost the same level of classification accuracy as the tested traditional SVM classification models. There is a slight sacrifice in accuracy when compared with LIBSVM with RBF kernel, which is reasonable. It is noticeable that PAN-SVM performs better than the Naïve Bayes classifier and the decision tree classifier in most cases, especially when a larger number of landmarks are used.

Table 3.2. Performance comparison between PAN-SVM and other traditional classifiers

Datasets	**Na ïve Bayes**	**Decision Tree**	**LIBSVM Linear**	**LIBSVM RBF**	**PAN-SVM (15%)**	**PAN-SVM (25%)**
Australian	79.85	85.65	85.51	86.38	85.01	85.46
Breast cancer	96.19	95.46	96.98	97.21	96.88	97.01
Pima Diabetes	75.91	71.22	77.10	77.60	76.84	77.32
German	72.40	72.20	76.58	76.30	75.40	75.50
Heart	84.80	77.78	83.83	84.07	82.85	83.48
Ionosphere	82.05	89.74	88.27	94.80	89.31	92.19
Liver disorders	56.23	68.70	68.68	73.04	71.65	71.97
Splice	84.20	92.90	79.97	87.70	82.36	83.02

3.3.2.3 Comparison with other Distributed Classifiers

Most distributed classification models usually assume that the data contained in the individual data source have similar properties, such as same distribution. In practice, however, it might not be the case. Therefore, it is useful to evaluate the classification accuracy of PAN-SVM by considering two different scenarios: 1) data points are randomly assigned to the different data source to make sure each source has similar data pattern or statistical distribution. 2) data points are split equally into *s* subsets based on different value segments of one feature and assign them to *s* data sources. For example, the *Four-class* database can be divided based on the value of feature one (*f1)* according to its three value segments: [-1, -0.307692], (-0.307692, 0.285714] and (0.285714, 1], as shown in Figure 3.6. This splitting method makes sure that the data at each source have different statistical distributions.

PAN-SVM is then compared with a number of existing distributed classifiers: 1) SVM-Ensemble [51], which uses a simple, voting-based approach to a privacy-preserving distributed classification. Each participating institution trains their local model separately, and then the prediction outcome is determined by majority voting among the trained models. 2) Consensus-based SVM [52], which also uses landmark points to handle non-linear patterns. However, it trains the global linear SVM by constructing local models in each of the participating data

centers and then iteratively comparing and adjusting those local models until they all agree on the same set of parameters. This synchronized process is usually implemented by the Alternating Direction Method of Multipliers (ADMM) technique [53], which adds the parameters' deviation as a penalty to the objective functions of the local models.

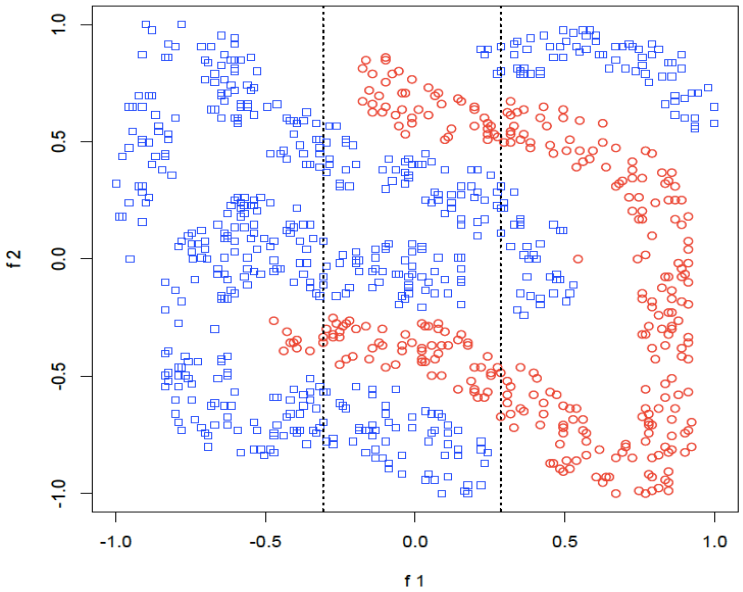


Figure 3.6. Four-class dataset is split into 3 groups by values of f1.

Table 3.3: Performance comparison based on different distributions.

Four-Class	**SVM-Ensemble**	**PAN-SVM**	**SVM-ADMM**
Random Distribution	99.16	99.90	99.46
Different Distributions	**84.48**	99.94	99.69

Table 3.3 presents the classification accuracy of PAN-SVM, SVM-Ensemble and SVM-ADMM under two data division scenarios. PAN-SVM is built based on 35% samples as landmarks. It can be clearly the three classifiers are competitive when distributed data have a similar distribution. However, when the data are partitioned based on the value segments of some feature, the SVM-Ensemble method achieves poorer classification accuracy, while PAN-SVM and SVM-ADMM still work well. A simple explanation is that SVM-Ensemble uses only local information in predicting the unseen data; in general, the unseen data might come from a dataset

that has an entirely different statistical data distribution than the local dataset. Besides, it can be seen from Table 3.3 that PAN-SVM achieved comparable or a bit better performance regarding classification accuracy.

3.3.3 Efficiency

Both PAN-SVM and SVM-ADMM use landmark points to handle linear inseparable classification, and they are competitive regarding classification accuracy. The main difference between the two approaches is the method used to solve SVM. PAN-SVM uses the cutting-plane technique to solve a global SVM, while SVM-ADMM builds local models from each data source and iteratively synchronizes their parameters. To compare the efficiency of PAN-SVM with SVM-ADMM, multiple data source scenarios are simulated in MATLAB, and the average time and an average number of iterations required to solve SVM training process in each case are recorded. The implementation and parameters used for ADMM are based on Boyd's example [53]. This efficiency test is divided into two parts: one is based on a small dataset, which is intended to test the stability of PAN-SVM, and the other is evaluated by larger datasets to test PAN-SVM's scalability.

3.3.3.1 Stability

The stability testing is conducted on the *Fourclass* and *Pima* datasets. For the test, the data are split into multiple groups using the same two strategies as in the *Effectiveness* test section. Tests are performed via simulating 5, 10 and 20 distributed data sources. The time required to solve SVM in each case is recorded. The results are shown in Table 3.4 and Table 3.5, where μ and σ are the mean and standard deviation, respectively. Figure 3.7 (a) and (b) show the average training time taken by PAN-SVM and SVM-ADMM according to different distributions of multi-source datasets, respectively. (c) and (d) show the average training time taken by PAN-

SVM and SVM-ADMM according to different numbers of data sources. From the testing results on Fourclass and Pima datasets, as shown in Table 3.4, Table 3.5 and Figure 3.7. We can observe that PAN-SVM is much faster and more consistent than SVM-ADMM. For the same small dataset, PAN-SVM takes less 0.1 seconds to build SVM, while SVM-ADMM spends more than 2.5 seconds, which is about 250 times speedup.

Table 3.4: Time (second) spent in solving PAN-SVM and SVM-ADMM on Four-class dataset.

# of data sources	**PAN-SVM**			**SVM – ADMM**		
	5	10	20	5	10	20
Same Distribution	μ=0.094 σ=0.005	μ=0.089 σ=0.008	μ=0.087 σ=0.007	μ=2.72 σ=0.02	μ=0.73 σ=0.05	μ=2.59 σ=0.75
Different Distributions	μ=0.093 σ=0.007	μ=0.089 σ=0.008	μ=0.084 σ=0.008	μ=3.39 σ=0.60	μ=1.96 σ=0.43	μ=7.37 σ=2.68

Table 3.5: Time (second) spent in solving PAN-SVM and SVM-ADMM on Pima dataset.

#of data sources	**PAN-SVM**			**SVM - ADMM**		
	5	10	20	5	10	20
Same Distribution	μ=0.092 σ=0.006	μ=0.088 σ=0.006	μ=0.067 σ=0.007	μ=8.04 σ=2.13	μ=18.91 σ=1.18	μ=7.38 σ=0.50
Different Distributions	μ=0.091 σ=0.008	μ=0.090 σ=0.008	μ=0.066 σ=0.008	μ=9.73 σ=3.97	μ=18.09 σ=2.69	μ=7.62 σ=2.21

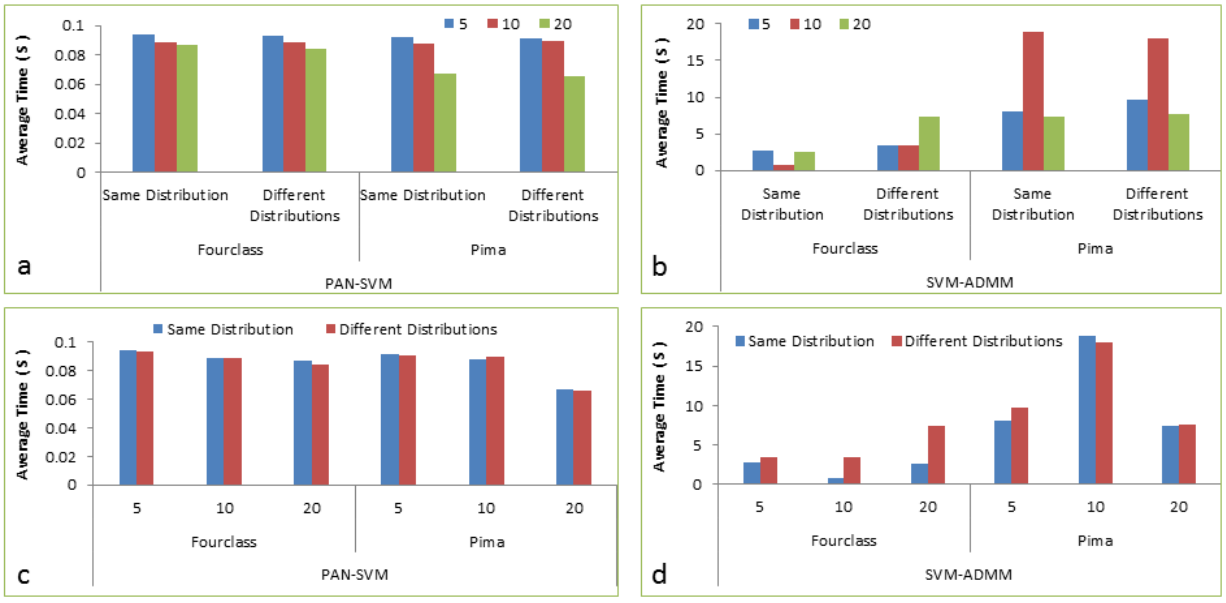


Figure 3.7. Average training time of PAN-SVM and SVM-ADMM testing on Fourclass and Pima datasets.

The other two major differences are: 1) the speed of PAN-SVM is not affected by the distributions of different data sources, and the time spent to solve PAN-SVM almost keeps unchanged no matter whether the data distribution is the same or not. On the opposite, SVM-ADMM is dramatically affected by the data distribution. 2) As the number of data sources increases from 5 to 20, the time needed to conduct PAN-SVM decreases; on the contrary, time that needed to build SVM-ADMM increases sharply as the number of data sources increases. Besides, the averages and standard deviations of the training time for SVM-ADMM are much larger than those of PAN-SVM (details are not shown here). Therefore, PAN-SVM is more much stable than SVM-ADMM when dealing with distributed data with different data properties (such as distributions).

It is also interesting to note that for the *Pima* dataset SVM-ADMM requires less training time on 20 data sources than it is from 10 data sources. This may be because the larger the number of data sources, the faster SVM-ADMM can solve the quadratic optimization problem since each data source will have a smaller number of data. However, larger numbers of data sources tend to increase the deviation of parameters in each data-source. This can be seen in Figure 3.8 and Figure 3.9, which show the average number of iterations that ADMM used for the *Fourclass* and *Pima* datasets, respectively. In these figures, *RandomSplit-5* represents 5 local datasets are split randomly; *FeatureSplit-5* denotes data are split by features at 5 local data sources, and so forth. In both cases, the number of iterations and the variance of the number of iterations in each case increase when the number of data sources increases. It will also take longer for ADMM to build the SVM model when the data in the different data sources have different properties (like statistical distributions).

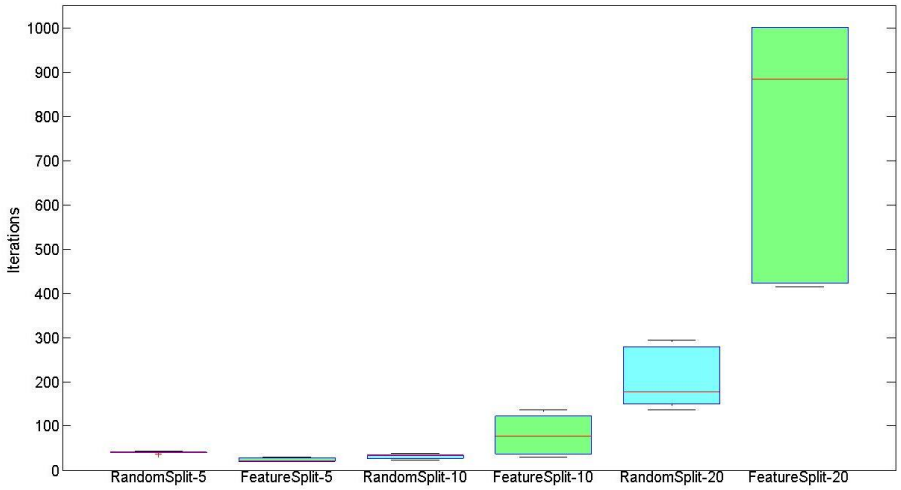


Figure 3.8. Number of iterations for quadratic optimization when building ADMM-SVM on Four-class dataset.

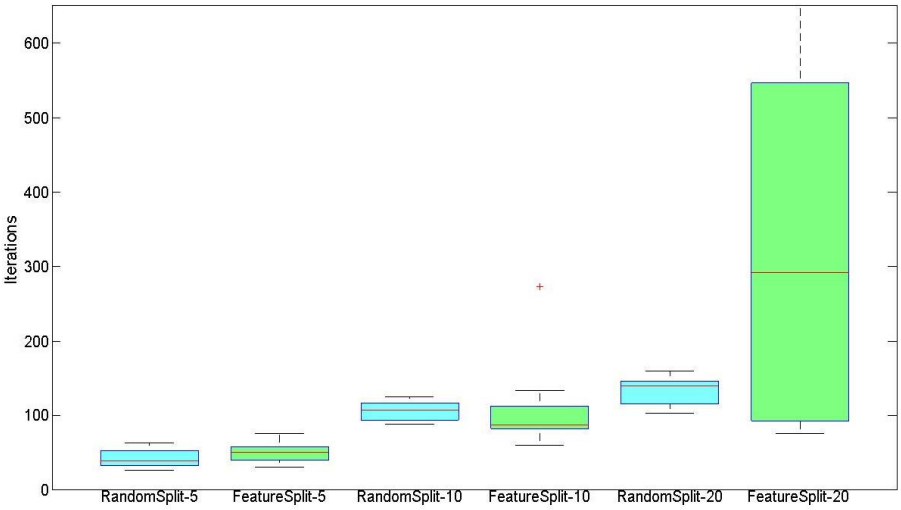


Figure 3.9. Number of iterations for quadratic optimization when building ADMM-SVM on Pima dataset.

3.3.3.2 Scalability to Large Scale Dataset

Testing is also done to assess the ability of PAN-SVM to scale up to larger datasets of *Adult*, *cod-RNA* and *GSE2990* datasets. All of the datasets are split into 5 simulated data sources, and tested through 5-fold cross-validation, and the results are shown in Figure 3.10 (a) and (b) show

the average training time and iterations of optimization as different sample sizes and features. The experimental results show that the average training time for small datasets (say sample size less than 1000) is less than one second. Moreover, Figure 3.10 (a) also show that the average time taken by PAN-SVM increases as sample size increases, but remains very fast on these tested datasets. 14 seconds are required for the *Adult* dataset, and 47 seconds for *cod_rna* dataset. In addition, Figure 3.10 (b) shows that the average training time required to train PAN-SVM is not affected significantly as the number of features increases.

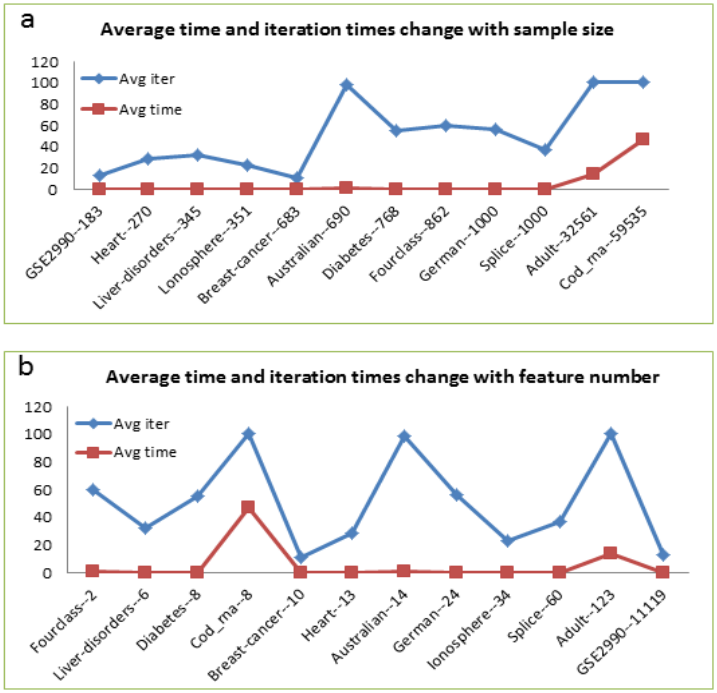


Figure 3.10. The average training time and average iteration counts of PAN-SVM according to sample size (a) and number of features (b), respectively.

The blue curves in Figure 3.10 (a) and (b) show the changing trends of the average iteration counts when solving the quadratic optimization by PAN-SVM according to different sample size and feature numbers, respectively. The axis represents '*database name--sample size'* in (a), and '*database name--feature number'* in (b). Unlike the average training time, the average iteration

counts do not always increase as the number of samples increases or the number of features increases.

Further experiments are conducted to test the training time of PAN-SVM as the number of landmarks changes. The results are shown as illustrated in Figure 3.11, the average training time for PAN-SVM according to different numbers of landmarks (training sample size changes). From Figure 3.11 we can observe that the average training time increases as the number of landmarks increases. But as mentioned in previous paragraphs, PAN-SVM still works effectively on the testing datasets, only 14 seconds the *Adult* dataset, and 47 seconds for *cod_rna* dataset.

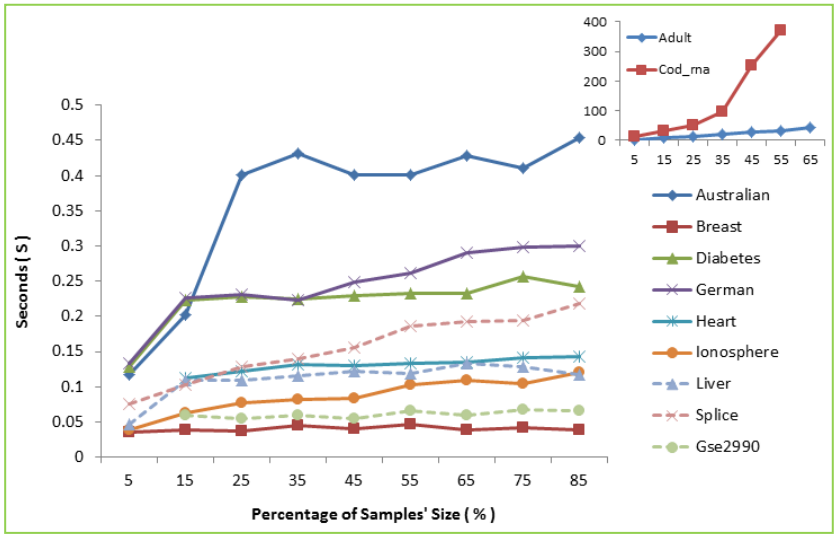


Figure 3.11. Average training time of PAN-SVM as number of landmarks changes.

Table 3.6: Rough comparisons of training speed in second.

Dataset	PAN-SVM	SVM-ADMM[52]	LIBSVM[54]	Fourier+LS[55]	Binning+LS [55]
Adult	13.9	2245.7	550.2	9	90

Table 3.6 shows a comparison of training time among several methods using the *Adult* dataset. The results presented for LIBSVM [54] and random feature techniques [55] are taken from the relevant literature listed in Table 3.6. PAN-SVM outperformed LIBSVM and the Binning+LS random feature method [55]; notably, PAN-SVM also significantly outperforms

SVM-ADMM and is comparable with the random feature method Fourier+LS. The comparison results of relative training efficiency may be varying if same programming language and same platform are set up.

Besides the time for the training process, the time consumed for k-means clustering, Nystrom approximation and matrix decomposition are also recorded, as shown in Figure 3.12, where *trn_time* denotes the time required by the training process, *KM_time* for k-means clustering, *Nystrom_time for* Nystrom approximation and *Decop_time* for matrix decomposition, respectively.

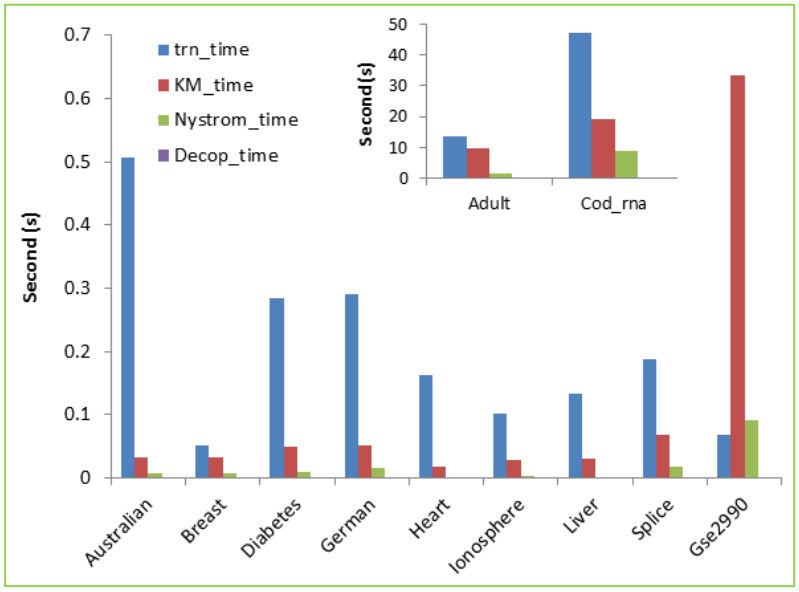


Figure 3.12. Time consumed by different procedures of PAN-SVM.

It can be seen that the training process takes most of the time needed by PAN-SVM for most datasets except GSE2990, whose time for k-means clustering is still less than 1 second. The time required by matrix decomposition is always less than one second, which can be ignored here. K-means clustering, as well as Nystrom approximating process, also only occupy a minor part of the total time required by PAN-SVM. Combining the experimental results as shown in Figure

3.12, we can conclude that PAN-SVM may hence still be scaled to large-scale datasets accordingly.

Franc et al. [42] demonstrates that the quadratic optimization used in the Optimized Cutting Plane Algorithm will converge within a limited number of steps. PAN-SVM is tested on 12 datasets – including the *Adult* and *cod_rna* datasets, and the highest iteration count converged to a constant number of 101. Franc et al. [42] also shows that an SVM using the OCA can be trained in a practical amount of time on a large scale dataset (one with 12-million features and 50 million samples) – in fact, they achieves a new performance record while doing so. Therefore, it is likely that PAN-SVM, which also uses OCA, can be scaled up to similar large datasets. However, the current implementation is based on Matlab, whose memory limitations prevent further experiments on such large-scale datasets; it is expected that, if PAN-SVM is re-implemented in a less-limited language, it will be able to handle such large-scale datasets with millions of features and samples.

3.4 Conclusions

In this chapter, we proposed a framework to solve privacy-preserving classification for multi-source data. PAN-SVM consists of three layers, which collaborate to make classification efficiently and prevent the disclosure of local data to third-parties. The k-means clustering method is employed to help the participating local data centers select better landmark points; these are then sent to the medium layer after being encrypted via *the secure sum protocol*, which prevents local data from being disclosed to third-parties. A global SVM is securely constructed in the medium layer from distributed datasets via Nystrom low-rank approximation and kernel matrix techniques, and the linear inseparable SVM is converted into a linear one. In the top layer, cutting-plane techniques are employed to accelerate the SVM training process.

PAN-SVM has been tested on 12 datasets, and the experimental results show that it yields better classification accuracy than traditional classification methods (like Naive Bayes classification or decision trees) and that it possesses the same level of accuracy as traditional SVM, such as LIBSVM with RBF kernel function. PAN-SVM can be effectively trained in a distributed manner and can yield comparable or superior accuracy than some existed distributed classifiers. Moreover, PAN-SVM performs stably even when the data stored in different sources contain very different patterns or distributions. In addition, it can handle enormous numbers of data sources, because the average training time tends to decrease or does not vary significantly as the number of data sources increases.

PAN-SVM is also tested on three larger datasets, the *Adult* and *cod_rna* datasets each contain more than 30,000 samples, and *GSE2990* contains more than 10,000 features. Experimental results show that even conducting on such large datasets, the training time is still less than one minute. The average training time is not affected by the number of features present either. Unlike the average training time, the average number of iterations required by the training process is bounded by a constant, which is approximate to 100 in our test. Even though these datasets are not big enough, PAN-SVM may still be scaled to large datasets with millions of features and records, if it is implemented in an efficient programming language as demonstrated by [42].

4 PRIVACY PRESERVING MULTI-CLASS CLASSIFICATION FOR HORIZONTALLY DISTRIBUTED DATASETS

4.1 Introduction

Nowadays, machine learning and data mining tools have become increasingly important to analyze and discover useful knowledge in many applications. Classification is a problem of identifying the categories for data belong to unknown groups by building effective classifiers based on known data samples as the training set. It is a very an important issue in machine learning and data mining research areas. Multi-class classification, as a branch of classification problem, has been being a hot topic and research direction in many domains during the past years and become more and more important in the era of big data. Researchers have proposed a significant number of state-of-the-art multi-class classification approaches and algorithms based on traditional but popular classification algorithms, such as Support Vector Machine (SVM), Decision Tree (DT), Na ïve Bayes (NB) and K-Nearest Neighbor (KNN).

Currently, the methods for solving multi-class classification problem can mainly be formulated into two cases. The first case aims to directly solve multi-classification by extending existed classifiers, such as SVM, DT classifier, NB classifier and KNN classifier to multi-class classifiers. On the opposite, the second case tries to solve the problem by converting it to multiple binary classification problems.

As the interests of assembling data mining on distributed data increase, the privacy concerns also increase. Therefore, to develop privacy preserving multi-class classification algorithms has become urgent. This chapter introduces a Privacy Preserving Multi-Class Classification (PPM2C) [56] method for horizontally distributed data, details are represented in the Methods section, which is followed by the experimental results and conclusions.

4.2 Methods

4.2.1 Multi-class Support Vector Machine

Support Vector Machine (SVM) is a well-known sophisticated classification method and has been widely used in many domains. SVM was originally designed for binary classification problem and approaches used to extend it to multi-class classification problem are simply divided into two types. One is directly considering all data in one optimization formula, and the other indirectly solve the problem by constructing and combining multiple binary classifiers, which are usually SVM classifiers. The indirect way can also be formulated in two cases: One-Versus-All (OVA) and One-Versus-One (OVO) or All-Versus-All (AVA), we name it OVO in the current work.

1) One versus All: for a ***k***-class classification problem, the OVA method constructs ***k*** SVM classifiers, the h^{th} SVM is trained by taking all of the samples in the h^{th} class with a positive label (+1), and all samples in the rest classes with a negative label (-1), as illustrated in Figure 4.1.

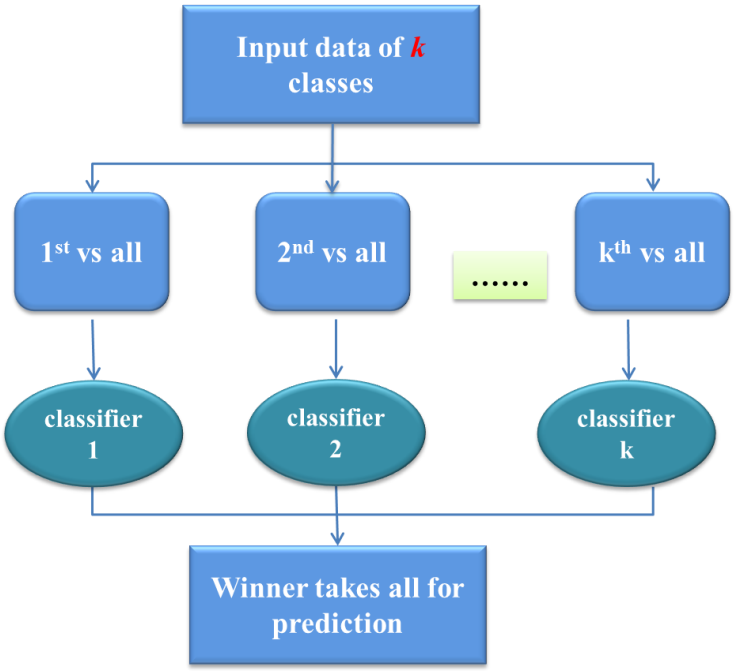


Figure 4.1 One-Versus-All multi-class classifiers.

Thus given a dataset with D of m samples $\mathrm{D}=\{(x_i, y_i) \mid i=1...m\}$, where $x_i \in R^n$ is a sample with n attributes and $y_i \in \{1,2,...,k\}$ is the class label of x_i, and the h^{th} SVM solves the following problem in (4.1):

$$\begin{aligned} &\min_{w^h, b^h, \xi^h} \frac{1}{2}(w^h)^T w^h + C\sum_{i=1}^{m} \xi_i^h \\ &(w^h)^T \Phi(x_i) + b^h \geq 1 - \xi_i^h, \ \ if \ \ y_i = h, \\ &(w^h)^T \Phi(x_i) + b^h \leq -1 + \xi_i^h, \ \ if \ \ y_i \neq h \\ &\xi_i^h \geq 0, i = 1,..., m \end{aligned} \tag{4.1}$$

Where Φ is a kernel matrix, the Radial Basis Function (RBF) kernel is used here $\Phi = \exp(-\gamma \|x - x'\|^2)$ and C is the penalty parameter. To train the h^{th} SVM is to find the maximal separate hyperplane by maximizing the term $2/\|w^h\|$. After solving (4.1), there are k decision functions in the predicting step as descripted in (4.2):

$$\begin{aligned} &(w^1)^T \Phi(x) + b^1, \\ &\quad \vdots \\ &(w^k)^T \Phi(x) + b^k. \end{aligned} \tag{4.2}$$

There will be k output values for k classifiers. If the predicted value for x is positive, we say x in the class with the positive label in the current classifier. Otherwise, we say x in the class which has the largest value of the decision function in (4.3):

$$\textit{class labe of } \ x \equiv \arg\max_{h=1,2,...k}((w^h)^T \Phi(x) + b^h). \tag{4.3}$$

The advantages of OVA scheme is that only k binary SVM classifiers have to be trained for a k-class classification problem, which speeds up the whole training process. However, the one versus all method might make the training data unbalanced dramatically.

2) One Versus One: for a ***k***-class classification problem, OVO will constructs *k(k-1)/2* binary classifiers to separate each one of the other, such as class ***1*** vs. class ***2***, class ***1*** vs. class ***3***,…, class ***2*** vs. class ***3***…, class ***k-1*** vs. class ***k***, as illustrated by Figure 4.2.

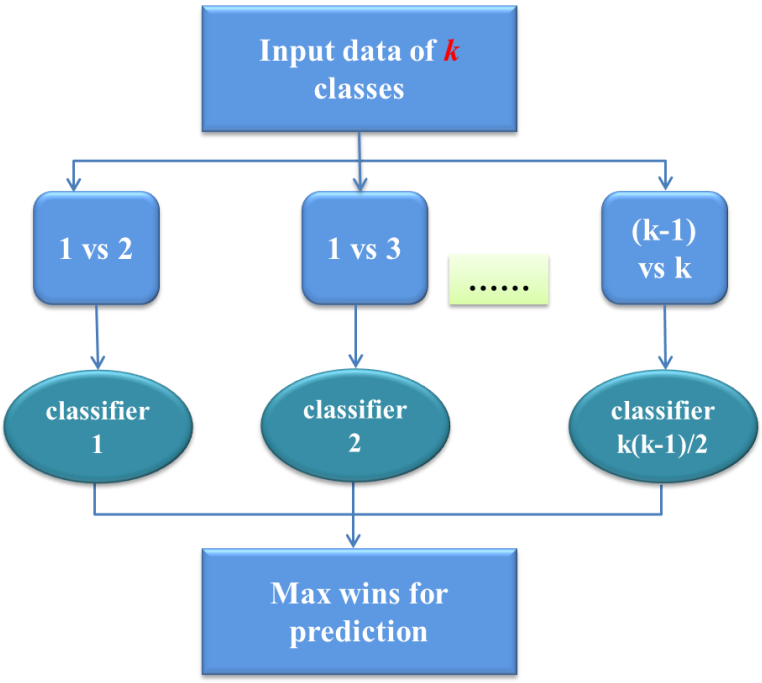


Figure 4.2 One-Versus-One multi-class classifiers.

For training data from the i^{th} and j^{th} class, OVO will solve the classification problem formulated by (4.4).

$$\begin{aligned} &\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij} \\ &(w^{ij})^T \Phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \ \ if \ \ y_t = i, \\ &(w^{ij})^T \Phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \ \ if \ \ y_t = j, \\ &\xi_i^{ij} \geq 0. \end{aligned} \tag{4.4}$$

There are different approaches that can be used to test unknown data after all the *k(k-1)/2* SVM classifiers are built. This is called a "Max Wins" strategy by a sign function. If it says *x* in the i^{th} class, then the vote for the i^{th} class will increase one; otherwise, the vote for the j^{th} class

will add one, then the predicted class for *x* is the one with largest voting value. Although it might have to train more binary classifiers for OVO than the OVA strategy, it is much faster. Since constructing several more SVM classifiers with smaller size is much faster than building fewer classifiers with larger size due to the quadratic programming optimization problems used in SVM. However, the "Max wins" might not be a good strategy in a case that the two classes have identical votes. In the current work, the OVA method is used for PPM2C.

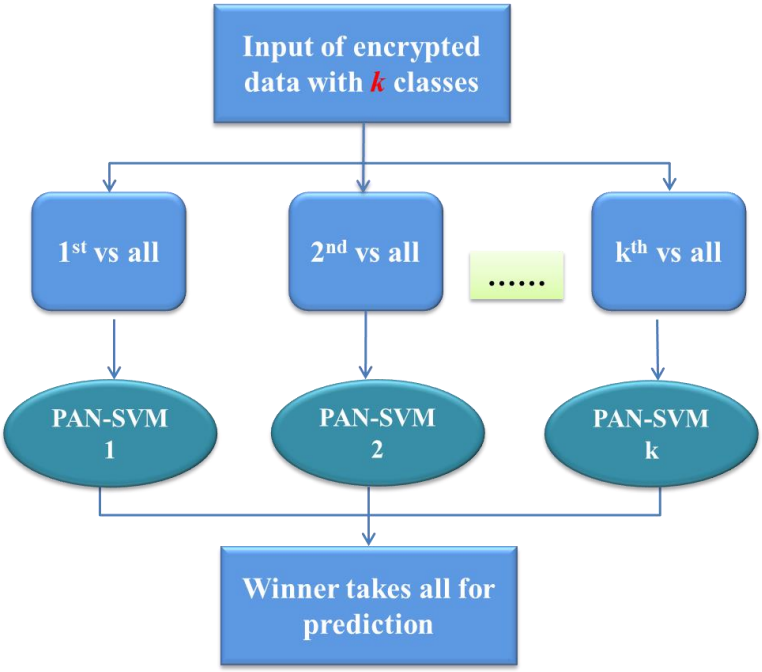


Figure 4.3. The workflow of PPM2C by using PAN-SVM.

4.2.2 Workflow of PPM2C

PPM2C [56] is based on the privacy preserving framework of PAN-SVM by converting the multi-class classification problem into building multiple binary PAN-SVM classifiers. Data are encrypted by the secure sum protocol and then transited to the destination securely. Encrypted data will be sampled by k-means clustering method, and then the sampled center data will be used to approximate the kernel matrix, which will be calculated in the process of building SVM

classifier. Thus sampled data with smaller size make the costly computation being avoided sharply and the complex computation of kernel matrix reduced significantly. The workflow of PPM2C is presented in Figure 4.3.

4.3 Results and Discussions

4.3.1 Datasets

PPM2C **is** tested on 6 datasets with a different number of classes, samples size and number of features. *DNA*, *Vowel* and *Letter* datasets are download from LIBSVM repository [50], and *Lung cancer dataset* is download from the University of California, Irvine (UCI) Machine Learning Repository [48]. *Leukemia data* [57] was originally introduced by Golub et al., in 1999 and it contains expression levels of 7129 genes for 47 ALL (Acute lymphoblastic leukemia) leukemia patients and 25 AML (Acute myelogenous leukemia) leukemia patients. The tested Leukemia datasets with 3 and 4 classes are download from [58]. In the 3-class dataset, *ALL* is split into 38 B-cell and 9 T-cell, and in the 4-class dataset, the *AML* is divided into 21 BM and 4 PB.

Table 4.1. The descriptions of multi-class datasets.

Dataset	**# of samples**	**# of features**	**# of class**	**C**	**γ**
Leukemia_3c	72	7129	3	512.0	0.0001220703125
Leukemia_4c	72	7129	4	512.0	0.0001220703125
DNA	2000	180	3	8.0	0.03125
Vowel	528	10	11	2.0	2.0
Lung	32	56	3	2048.0	0.00048828125
Letter	15000	16	26	8.0	2.0

The C in Table 4.1 is a penalty parameter of SVM, and γ is a free parameter in (Gaussian) Radial Basis Function (RBF) kernel. C and γ are generated by LIBSVM [50] by using 10-fold cross validation. The details about the datasets are presented in Table 4.1.

4.3.2 Performance Assessing

In PPM2C, PAN-SVM is employed to construct the multiple SVM classifiers. The performance of PPM2C is assessed via the classification accuracy, which is formulated by (4.5).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.5}$$

Where *TP* represents True Positive, *TN* states True Negative; *FP* denotes False Positive, and *FN* indicates False Negative.

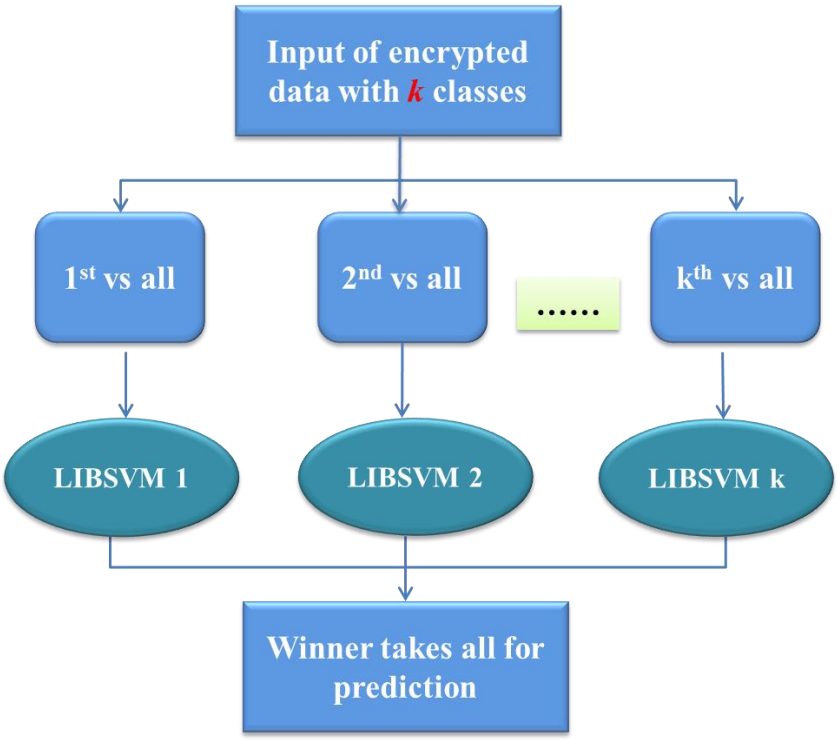


Figure 4.4. Workflow of PPM2C by using LIBSVM

The experimental results of PPM2C are compared with those obtained by using LIBSVM [50] as a regular binary SVM classifier. The scheme is the same as using PAN-SVM, as shown in Figure 4.4. In the following paragraphs, *PrivacySVM* and *RegularSVM* are used to represent the two different binary classifiers of PAN-SVM and LIBSVM. 5-fold cross validation is used for each binary SVM classifier, and the results shown in this chapter are the average value from ten

rounds, each round contains a 5-fold cross validation results. In other word, the experimental results approximate the average of 50 tests. Besides, for *PrivacySVM*, different percentages for landmarks (introduced in chapter 3) are tested, from 25% to 90%. Each percentage is tested by 10 rounds, and the results shown in the following two subsections are the average accuracies of these 14-time tests (25%, 30%, 35%, 40%, 45%, 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85% and 90%). Details are presented in *subsection 4.4.3*.

4.3.3 Feasibility of PPM2C

The feasibility of PPM2C by using PAN-SVM is firstly tested to check whether the proposed framework and scheme are workable or not. We say PPM2C is workable if it can achieve approximate classification accuracies as using a regular SVM, like LIBSVM. The experiments are tested on four benchmark datasets from UCI and LIBSVM repositories and two microarray datasets, and the results are denoted in curves as shown in Figure 4.5, which shows the changes of classification accuracy as the percentages of landmarks change.

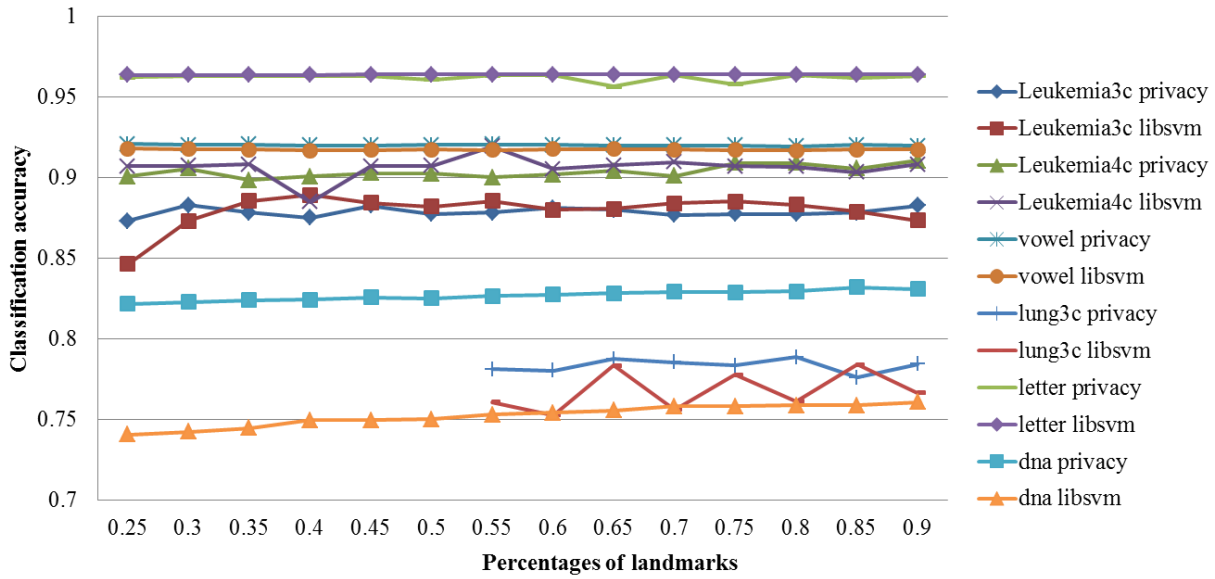


Figure 4.5. Classification accuracy changes as the percentages of landmarks change.

Since PAN-SVM is based on the landmarks, therefore, we calculated the classification accuracy with different percentages of landmarks and compared the results with those obtained by LIBSVM with the same number of samples, from 25% to 90%. Because there are only 47 samples in the *lung3c* datasets, the percentages are chosen from 55% to 90% for this dataset to make sure there are enough landmarks to approximate the kernel matrix. In Figure 4.5, '*privacy*' denotes PAN-SVM and '*libsvm*' represents regular SVM. The classification accuracy is obtained by 5-fold cross-validation.

From Figure 4.5, we can observe that among the six datasets, L*eukemia_3c*, *Leukemia_4c*, *vowel* and *letter* can achieve very close classification accuracy between PAN-SVM and LIBSVM, but there are some sacrifices in accuracy for PAN-SVM when compared with LIBSVM, and this is reasonable because the kernel function of PAN-SVM is approximated. For *lung3c* dataset, the average classification accuracy of PAN-SVM is a little higher than LIBSVM; the reason might be because the small size of samples in this dataset and LIBSVM cannot obtain enough information to build the predicting model. For *DNA* dataset, PAN-SVM outperforms LIBSVM, and the performance can be improved as high as 8%, the reason might be the sparse property of this dataset. Since PAN-SVM employs k-means clustering method to generate the landmarks, more supportive information might be obtained than LIBSVM. These results demonstrate that PPM2C that hires PAN-SVM is workable, feasible and reliable.

4.3.4 Stability of PPM2C

As discussed in previous literatures, the classification accuracy is usually assessed by cross-validation, 5-fold cross validation is used in the current work. During the cross-validation process, data will be randomly split into *k* (*k*-fold) subsets, and at each training round, *k-1* subsets are used as training data, and the left 1 subset is used as testing set. In other word, all of

the samples will be involved in the whole process, we call this kind of cross-validation as CV1, which mentioned in [59]. However, as pointed by [60-62], a CV1 error may severely bias the evaluation, which is demonstrated by [59] via simulation data. [59] gave another evaluation, named CV2, which leaves the test samples out of training set before any feature selection step. Although no feature selection step is needed to test PPM2C, CV2 criterion can also be used to test the performance of PPM2C. In the current work, 1/5 of the total samples are randomly selected to be used as a separate testing set under CV2 and does not involve in the training process at all. All samples will involve in the training process under CV1.

Figure 4.6 and Figure 4.7 show the experimental results tested on PAN-SVM and LIBSVM under CV1 and CV2 test situation. From Figure 4.6 and Figure 4.7, we can observe that the classification accuracy of PAN-SVM is slightly improved under CV2 on the three microarray datasets of *leukemia3c*, *leukemia4c*, and *lung3c*, but there is no significant difference between them. On the opposite, the classification of LIBSVM is reduced for these three microarray datasets under CV2. This phenomenon illustrates that LIBSVM has the problem of over-fitting, while PAN-SVM can mitigate this risk.

Figure 4.6. Classification accuracy of PAN-SVM.

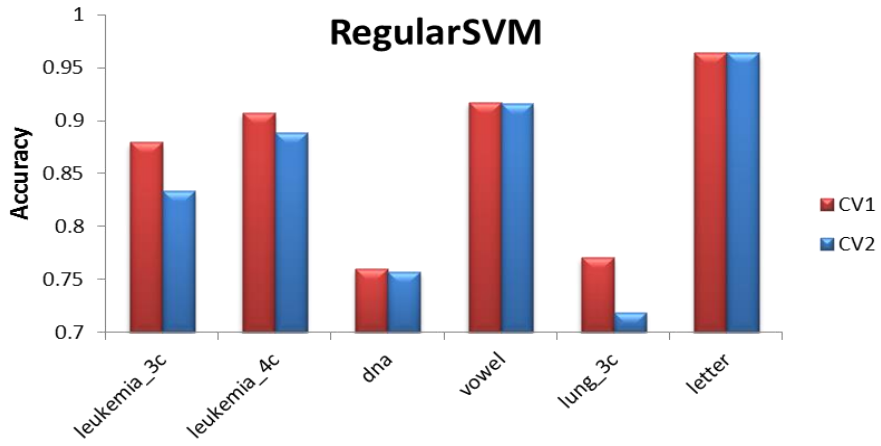
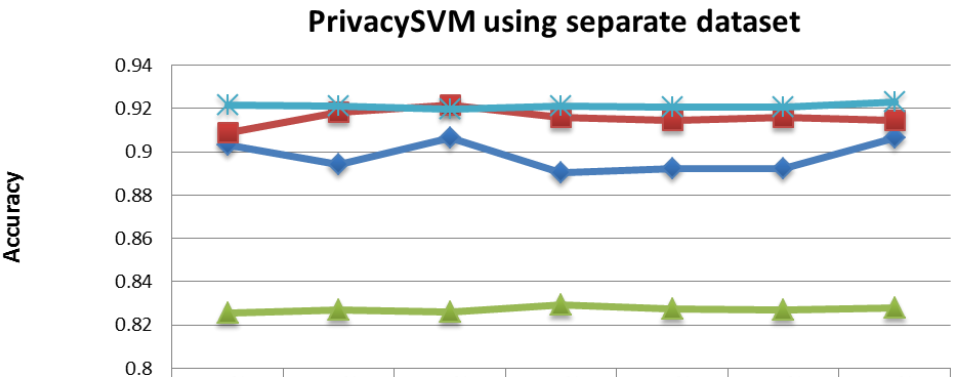


Figure 4.7. Classification accuracy of LIBSVM.

The predicting accuracies are increased by PAN-SVM for datasets *Leukemia_3c, Leukemia_4c,* and *Lung cancer*, but the improvements are very slight (less than 1.68%), it may say that the improvement is not significant. In other word, PPM2C by employing PAN-SVM is stable, no matter for independent (separate data) testing dataset or not. On the opposite, the predicting performance of LIBSVM classifier is decreased (~5.19%) by using independent testing samples LIBSVM, which means that CV1 makes LIBSVM achieve high classification accuracy, especially for small datasets, such as *Leukemia_3c*, *Leukemia_4c*, and *Lung cancer.* An independent dataset being separated from the training process means fewer samples and information are used to construct the classifier, which might be the reason why LIBSVM performs poorly under CV2 situation, which illustrates that the regular SVM has the problem of over-fitting under CV1 situation. On contrast, PAN-SVM is much more stable than LIBSVM and has better classification ability for small data.

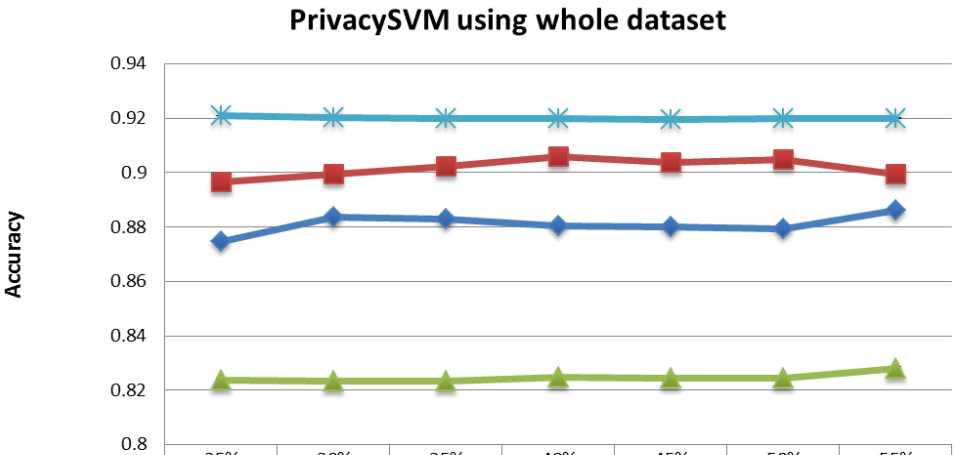
To further demonstrate the stability of PPM2C using PAN-SVM, more tests are done. Since PAN-SVM depends on landmarks for approximating kernel matrix, so the tests are conducted according to different percentages (25%, 30%, 35%, 40%, 45%, 50% and 55%) of landmarks at

CV1 and CV2 situation, respectively. The experiment is tested on *Leukemia_3c*, *Leukemia_4c*, *DNA* and *Lung cancer datasets* and the experimental results are as shown in Figure 4.8 and Figure 4.9.



	25%	30%	35%	40%	45%	50%	55%
leukemia_3c	90.32%	89.43%	90.63%	89.05%	89.21%	89.24%	90.63%
leukemia_4c	90.89%	91.82%	92.14%	91.57%	91.43%	91.57%	91.43%
dna	82.56%	82.69%	82.58%	82.93%	82.72%	82.68%	82.81%
vowel	92.14%	92.11%	91.95%	92.11%	92.08%	92.06%	92.29%

Figure 4.8. Classification of PAN-SVM under CV2 with different landmarks.



	25%	30%	35%	40%	45%	50%	55%
leukemia_3c	87.45%	88.38%	88.29%	88.06%	88.01%	87.92%	88.61%
leukemia_4c	89.65%	89.93%	90.24%	90.59%	90.38%	90.49%	89.93%
dna	82.37%	82.32%	82.32%	82.47%	82.45%	82.45%	82.81%
vowel	92.10%	92.03%	91.98%	92.01%	91.97%	91.99%	92.00%

Figure 4.9. Classification of PAN-SVM under CV1 with different landmarks.

From Figure 4.8 and Figure 4.9, we can observe that no matter using complete data at CV1 case or separate data at CV2 case, the classification accuracy has no obvious change by using different numbers of landmarks, and the change are from 0.13% to 1.16% for the whole dataset, and from 0.34% to 1.59% for the separate dataset. The accuracy curves generated under CV1 are relatively smooth than those under CV2. This evidence demonstrates PAN-SVM's ability to classify data with small size; the predicting accuracy also keeps stable under different landmarks (sample sizes). On the opposite, LIBSVM becomes less effective when dealing with the very small dataset.

4.4 Conclusions

In this chapter, a Privacy Preserving Multi-Class Classification (PPM2C) method is proposed based on our previously proposed privacy preserving classification framework of PAN-SVM. PPM2C converts the multi-class classification problem into multiple binary classifiers, which are PAN-SVM classifiers here. It works just like PAN-SVM, data are encrypted via the *Secure Sum Protocol* at the bottom layer, and sampled landmarks are used to approximate kernel matrix, which has to be computed during SVM training process. PPM2C inherits the privacy preserving and effectiveness properties of PAN-SVM but can solve multi-class classification problem.

The performance of feasibility and stability of PPM2C are assessed by testing on six benchmark datasets under two situations, say CV1 and CV2 and compared between *Privacy SVM* (PAN-SVM) and *Regular SVM* (LIBSVM). In case of CV1, all data involve in the cross-validation process for training and testing, while for the type of CV2, an independent dataset is randomly sampled from the whole dataset and used as test samples. Firstly, the feasibility of PPM2C is tested under CV1 and compared that with LIBSVM, and the experimental results indicate that the privacy SVM can work as effective as regular SVM and can even achieve higher

classification accuracy for some datasets with small size or sparse data. Tests on the separate data show that PPM2C with PAN-SVM outperforms the LIBSVM at the level of predicting accuracy, especially for small data. However, PAN-SVM has no significant improvement when using separate data (CV2) compared with complete data (CV1). LIBSVM works on the opposite; the predicting accuracy decreases via using separate data. These experimental results demonstrate that PPM2C is stable and can reduce the risk of over-fitting like LIBSVM.

Further experiments are conducted for PAN-SVM using different percentages of landmarks. The testing results show that PPM2C's ability to predict is not affected by sample size, and it works much more efficiently than LIBSVM for a dataset with small size.

5 PRIVACY PRESERVING FEATURE SELECTION VIA VOTED WRAPPER METHOD FOR HORIZONTALLY DISTRIBUTED DATASETS

5.1 Introduction

In the era of big, data mining approaches have been widely used to analyze the massive amount of data, and they have become increasingly important tools to discover useful knowledge in many domains. Nowadays, a lot of scientific fields have experienced a huge growth in data volume and data complexity, which brings data miners many opportunities, as well as challenges. For example, assembling datasets from distributed locations has become increasingly common [63-65], since applying data mining techniques on the aggregated datasets can build much more reliable prediction models and attain useful patterns from a wider picture, which benefits for medical research, improving customer service and homeland security, etc. However, mining on sharing data might divulge the sensitive information about individuals; it thus leads to increasing concerns about privacy during the process of data mining, therefore new sophisticated distributed data mining algorithms that can preserve privacy needed to be developed.

The huge number of data attributes or dimensions often makes a curse to data mining tasks. Feature selection techniques address the issue of dimensionality reduction by selecting some available subset of features via predetermined selecting criteria to decrease the complexity the data mining tasks and thus improve the performances (such as classification accuracy) of data mining algorithms. Take the classification problem into consideration, by doing feature selection, irrelevant and redundant features are usually eliminated. Thus the computational complexity of classification procedure is reduced, and a better classifier with generalization ability will be constructed, and the risk of over-fitting is also be reduced. Therefore, feature selection plays a vital role in optimizing classification procedure.

Feature selection methods can be grouped into two categories according to their searching directions: *forward selection* and *backward selection*. Forward selection usually starts searching relevant features from an empty subset and adds one or some at each step until a stop criterion is met. On contrast, the backward selection methods usually start searching for the whole feature space and eliminate or remove one or some at each step, until some the predetermined stop criteria are reached.

Moreover, feature selection methods can also be classified into three main groups: *filter, wrapper* and *embedded approaches* [66] according to different selecting strategies and procedures of algorithms. The filter methods usually take account of the statistical properties of features and rank them according to some criteria of relevant information. This step is always before the classification step and is entirely independent of data mining algorithms; they are usually fast. Just as the name implies, the wrapper methods often wrapped the feature selection step in the process of mining algorithms. Compared with the filter methods, wrapper methods have the advantages of taking account into the performance of mining algorithms or tasks. Thus a better classification model will be built with high performance, says high classification accuracy. However, it needs to repeatedly train and test the data and build classification model at each step when a subset of features are selected; the computational complexity thus increased sharply. In recent years, many approaches of wrapper feature selections are developed [59, 67-70]. The third kind of feature selection approaches is named embedded method, which performs feature selection in the process of the building data mining model by adding or modifying the optimizing process of classification [71, 72].

Feature selection algorithms can also be classified into two categories based on the relationship of features: *feature ranking* and *subset selection*. In the ranking list, the importance

of each gene is unequal. Usually the most top one is supposed to be the most important one, and so forth; while in the subset selection, each feature is equal, they work together making the classifier obtain the best performance.

Nowadays, feature selection has become an important research field and been playing a crucial step for data mining algorithms via eliminating the curse of dimensionality. Many feature selection approaches related to data mining tasks have been proposed as data are integrated into a central location. However, as the needs for new privacy preserving data mining algorithms increase, the needs for privacy preserving feature selection algorithms also grow rapidly, and the privacy concerns of sharing data by distributed parties also brings significant challenges to feature selection. In this chapter, a Privacy Preserving Feature Selection algorithm via Voted Wrapper methods (PPFSVW) [73] is proposed. PPFSVW is based on our previous work PAN-SVM [43] to protect individual privacy and tested on six benchmark datasets, including gene expression datasets. Details about PPFSVW are described in *Methods* section, and the experimental results are shown in the *Results and Discussion* section, followed by the conclusion at last.

5.2 Methods

5.2.1 PAN-SVM Classifier

As mentioned above, wrapper methods usually integrate feature selection step in the process of mining algorithms. When applied to a classification problem, methods used for selecting features are closely related to classifiers. In the current work, the classification accuracy is used as the wrapper method, and one of our previous works, PAN-SVM introduced in chapter 3 is used to be as the classifier for preserving privacy during the step of feature selection.

PAN-SVM contains three layers, which can finish corresponding functions. The bottom layer protects individual data privacy, where sampled data from multiple parties will be encrypted via the *Secure Sum Protocol* and sent to the remote miner. Data are sampled by k-means clustering methods and used as landmarks. At the medium layer, the landmarks will be used to approximate kernel matrix via Nystrom technique and the computation cost of kernel matrix will be further reduced via eigenvalue decomposition method. After the step of kernel matrix approximation and decomposition, non-linear separable SVM will be converted a linear separable one in this layer. Linear SVM will be optimized and speeded up by linear search and cutting plane techniques at the top layer. Although the classification accuracy of PAN-SVM sacrifices slightly when compared with the traditional SVM, such as LIBSVM with RBF kernel, the individual private information is preserved; furthermore, the training process is speeded up when compared with other distributed classification methods. Details about PAN-SVM can be found from [43].

5.2.2 Wrapper Methods

5.2.2.1 SVM-RFE

Just as the name implies, the wrapper methods often wrapped the feature selection step in the process of mining algorithms. Compared with the filter methods, wrapper methods have the advantages of taking account into the performance of mining algorithms or tasks. Thus a better classification model will be built with high performance, says high classification accuracy. However, it needs to repeatedly train and test the data and build classification model at each step when a subset of features are selected; the computational complexity thus increased sharply. In recent years, many approaches of wrapper feature selections are developed [59, 67-70]. Among these methods, the Recursive Feature Elimination (RFE-SVM) proposed by Guyon [74] is very popular. RFE-SVM employs Support Vector Machine as a classifier and aims to find the best

subset with r features by ranking the whole feature set according to a criterion of w^2, which is formulated in equation (5.1):

$$w_i = \sum_i^m \alpha_i y_i x_i. \tag{5.1}$$

Where w is the weighted vector of SVM classifier, α_i is nonzero if x_i is support vector, otherwise, α_i equals to zero. Therefore, this criterion can also be explained as the weighted sum of support vectors, which tries to achieve high performance by maximization the separation margin in SVM. The elimination procedure can be described by three steps:

Step 1. Train SVM classifier.

Step 2. Calculate the ranking scores w^2 for all features according to equation.

Step 3. Eliminate the feature which has the smallest ranking score.

The elimination procedure iterates the above steps until all features are eliminated and ranked, top features that make the classifier attain highest accuracy performance will be selected. However, over-fitting is an important issue in machine learning study, since SVM-RFE is aiming to find the features that maximum the separation margin, over-fitting also exists.

5.2.2.2 RSVM

To improve the robustness to noise and outliers, another Recursive Support Vector Machine (RSVM) is proposed in [59]. RSVM shares the same iterative procedures with SVM-RFE, but different ranking criterion, which is formulated by equation (5.2). RSVM also starts from the whole feature set and backwardly eliminates the feature with the least ranking score.

$$ranking\ \text{score} = w_j(m_j^+ - m_j^-) \tag{5.2}$$

Where w_j represents the weight of the jth feature, m_j^+ and m_j^- denotes the means of j^{th} feature in the positive and negative class, respectively. Unlike SVM-RFE, this method of RSVM takes

account into the classification information via weight, as well as the data itself by calculating the means of each class. By this recursive iteration step, a feature subset with smaller and smaller size will be selected, and the classification can also be performed on the selected features at each step. Top features with high selected-frequency will be chosen as the final selection results. However, this method is greatly affected by the class label, since the class means are used to calculate the ranking criterion, which makes the selection method unstable.

5.2.2.3 SVM-t

To conquer the disadvantages of RSVM and develop a stable selection method, Tsai et al. [70] proposed another wrapped feature selection method named SVM-t. It also follows the workflow of SVM-RFE and RSVM to eliminate least important features via backward selection procedure but employs t-statistics to be as the ranking criterion, as denotes in the equation (5.3).

$$\left|t_j\right| = \frac{(\mu_j^+ - \mu_j^-)}{\sqrt{((s_j^+)^2 / n^+) + ((s_j^-)^2 / n^-)}} \tag{5.3}$$

Where n^+ and n^- denotes the number of support vectors for the positive class (+) and negative class (-), respectively. μ_j^+ and μ_j^- indicate the means of the j^{th} feature in class+ and class-; s_j^+ and s_j^- represent the standard deviations of the j^{th} feature in class+ and class-, respectively. SVM-t just uses the most important subset of data, says support vectors, to evaluate the importance of each feature and construct the ranking criterion. It works well when data have significant statistical differences.

5.2.3 Workflow of PPFSVW

SVM-RFE directly chooses the weight vector as a ranking criterion, but it does not consider class information and has a high risk of over-fitting. RSVM outperforms SVM-RFE in the way

of improving its robustness to noise and outliers, but unstable to class label assignment. SVM-t uses only the support vectors information and outperforms other two methods when considering distinct variance between informative and non-informative genes, but it is only suitable for linear support vector machine. The current work proposed a feature selection algorithm via integrating these three methods during the feature elimination stage, for inheriting their advantages and improves the prediction accuracy, in the meanwhile; protect individual privacy via employing the privacy preserving framework of PAN-SVM.

PPFSVW [73] shares the common workflow with SVM-RFE, RSVM, and SVM-t, but has two main differences from them in the way of choosing eliminating feature at each step. First, PPFSVW employs PAN-SVM as classifier, which can guarantee the privacy to be preserved; second, it calculates the ranking scores for each feature according to equations (5.1), (5.2) and (5.3), respectively, and then eliminate the least important one via voting by the three measurements.

- Step 1: Train PAN-SVM.
- Step 2: Calculate ranking scores using the criteria of SVM-RFE, RSVM, and SVM-t.
- Step 3: Rank features according to the scores, and obtain three ranking lists.
- Step 4: Choose one feature that needed to be eliminated at this iteration in the following way:
    - If there is one feature which is selected by at least two methods, remove it, and go to step 1 until all features are ranked; otherwise,
    - Calculate the classification accuracy by 5-fold cross validation for classifiers, which with the three selected features eliminated, respectively, and then remove the feature, which has ***highest negative affection*** to the classifier, and then go to step 1 until all features are ranked.

- Step 5: Return a ranked list.

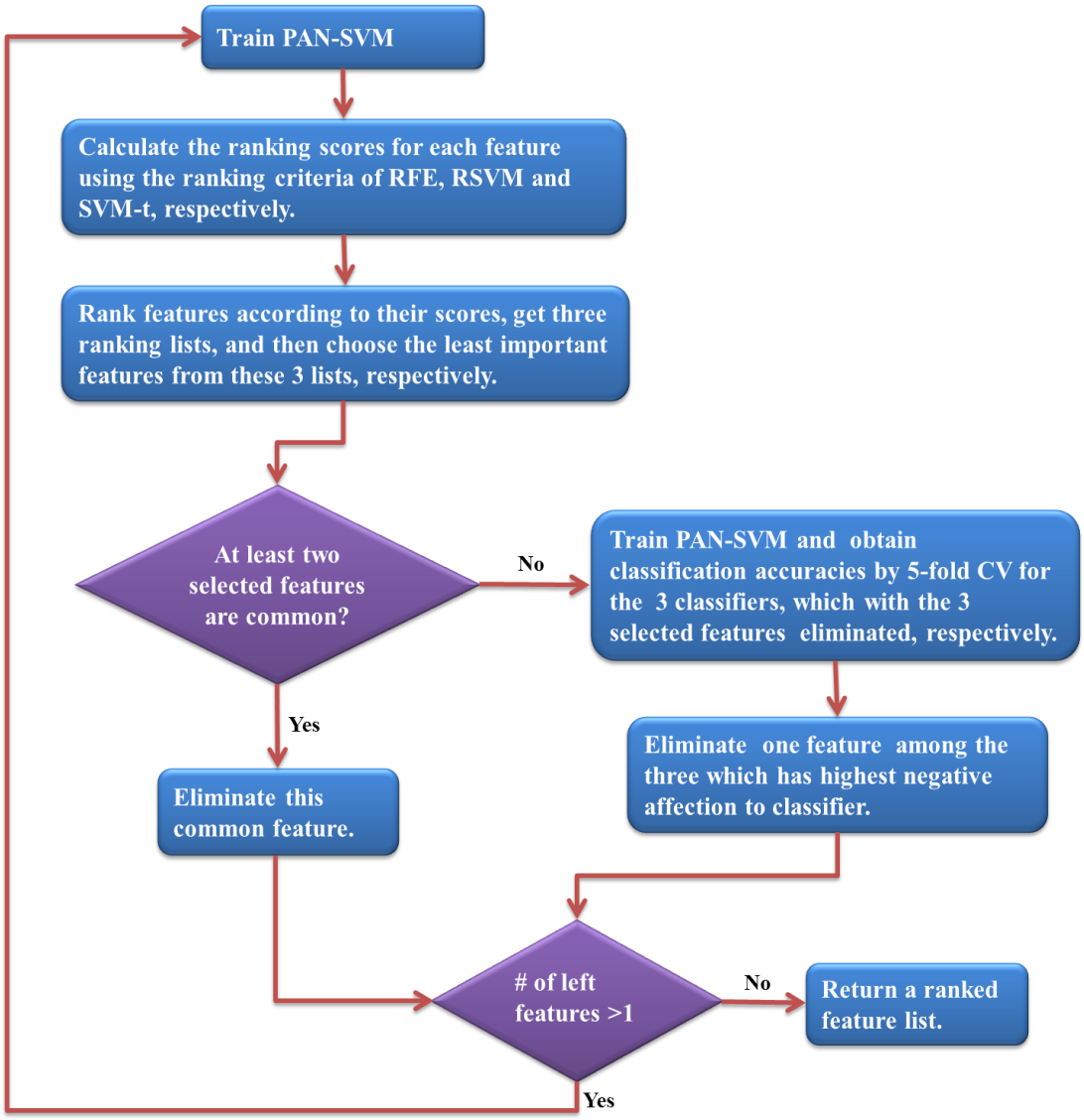


Figure 5.1. Workflow of PPFSVW.

The workflow chart is presented in Figure 5.1. In step 3, three feature ranking lists will be got according to the three ranking criteria formulated in the equation (5.1), (5.2) and (5.3), and the least important one in each list will be temporally chosen and voted in step 4 to decide which one should be eliminated finally at this iteration. If the three temporally selected features are different

from each other, PPFSVW will train classifiers with the three features eliminated respectively by 5-fold cross validation. For example, features 1, 2, and 3 are three temporally selected features waiting for eliminating, PPFSVW will train classifier number one with feature 1 being eliminated and get the classification accuracy of 90%, classifier number two with feature 2 being eliminated and get accuracy of 93%, and classifier number three with feature 3 being eliminated and get accuracy of 92%. Number two classifier obtains the highest accuracy by eliminating feature 2, it says feature 2 makes the highest negative affection to classifier. In other word, it is the least important one among the three temporally selected features; therefore, PPFSVW will eliminate feature 2 at this iteration and restore features 1 and 3. The eliminated feature will be put at the head in the queue of the ranking list. This procedure will repeat until all features are eliminated and ranked, with the most important feature at the top and least important one at the bottom (the tail in the queue).

5.3 Experiment Results and Discussions

5.3.1 Datasets

The performance of PPFSVW is assessed on six benchmark datasets, including 3 microarray datasets with different numbers of features, which are shown in Table 5.1. C and γ are the penalty parameter for SVM and a free parameter for Radial Basis Function kernel (RBF) used in SVM. They are generated by 10-fold cross validation.

The *Diabetes* and *Ionosphere* data are downloaded from LIBSVM repository [50], the Wisconsin Breast Cancer data (WBC) is downloaded from University of California, Irvine (UCI) Machine Learning Repository [48]. The *colon data* [57, 58, 75] contain 62 samples including 22 normal samples and 40 colon cancer samples. Each sample is described by the expression levels of 2000 genes. The *Leukemia data* [57, 58], originally introduced by Golub et

al., in 1999, contains 47 ALL (Acute lymphoblastic leukemia) leukemia patients and 25 AML (Acute myelogenous leukemia) leukemia patients with expression levels of 7129 genes. *DLBCL data* [76], the distinct types of diffuse large B-cell lymphoma (DLBCL) with expression levels of 4026 genes, contains 47 samples, 24 of them are from "germinal center B-like" group and 23 are "activated B-like" group.

Table 5.1 Details about datasets used for PPFSVM.

Dataset	**# of samples**	**# of features**	**C**	**γ**
Diabetes (DIA)	768	8	512.0	0.0078125
Ionosphere	351	34	8.0	0.5
Colon	62	2000	32.0	0.0078125
Leukemia	72	7129	128.0	0.0001220703125
Lymphoma	47	4026	2.0	0.0078125
Breast Cancer (WBC)	569	30	128.0	8.0

5.3.2 Performance Assessing

The performance of PPFSVW will be assessed by the measurement of classification accuracy, which is formulated by the equation (5.4), Where *TP* represents *True Positive*, *TN* denotes *True Negative*, *FP* means *False Positive* and *FN* states *False Negative*.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.4}$$

The Cross Validation (CV) method is often used to assess the performance of classifier due to lack of data that can be utilized as separate testing samples (like 5-fold cross validation, Leave One Out method). During the cross-validation process, data will be randomly split into *k* (*k-fold*) subsets, and at each training round, *k-1* subsets are used as training data, and the left *1* subset is used as testing set. However, as pointed by [59], the feature selection results may vary due to even a single difference in the training set, especially for small datasets. Many feature selection

methods are done with all samples, and the cross-validation step is only done during the classification process, which makes the feature selection external to the cross-validation procedures, and leads to 'information leak' in the feature selection step. It calls this kind of error made by cross-validation as a CV1 error. [60-62] also points out that CV1 error may severely bias the evaluation of feature selection. [59] also demonstrates the existing of the bias via simulation data and suggests another error evaluation method, named CV2. Under the CV2 scenario, a separate dataset is used as test samples and leaves out of training set before any feature selection step. In the current work, PPFSVW will be tested and evaluated under the two testing schemes. We use '*Separate*' to denote that the testing is conducted under CV2, and '*Whole*' to denote the experiment is conducted under CV1. 5-fold cross-validation is used to generated the classification accuracy at each selecting iteration.

5.3.3 Effectiveness and Performance Improvement

In this chapter, a novel feature selection algorithm of PPFSVW is proposed; the proposed workflow can be applied to both regular classifiers and privacy preserving classifiers. In the current work, the effectiveness of the proposed algorithm is firstly assessed via conducting experiments on PAN-SVM, as well as a popular regular SVM package LIBSVM [50, 77].

Table 5.2. Comparison of classification accuracy (%) between before and after feature selection via PAN-SVM.

PAN-SVM	***Separate (CV2)***			***Whole(CV1)***		
Datasets	*Voted*	*NoSelection*	*Improvement*	*Voted*	*NoSelection*	*Improvement*
DIA	79.35	76.48	2.86	80.13	76.76	3.37
Ionosphere	96.86	93.94	2.91	96.29	93.03	3.27
Colon	100.00	81.92	18.08	100.00	82.00	18.00
Leukemia	92.86	87.14	5.72	94.29	89.65	4.64
WBC	96.64	96.64	0.00	96.46	96.69	-0.23
DLBCL	100.00	87.76	12.24	100.00	89.05	10.95
SUM	565.70	523.88	**41.82**	567.17	527.17	**39.99**

The experimental results are represented as bar charts and shown in Figure 5.2 and Figure 5.3, Table 5.2 and Table 5.3, respectively. The experiments are conducted under both CV1 and CV2 testing scenario, which are shown as *Whole* and *Separate*, respectively. PAN-SVM is shown as '*PrivacySVM,*' aiming to emphasize its difference from regular SVM at the aspect of privacy preserving property, and '*RegularSVM'* denotes LIBSVM. '*Voted'* denotes the classification accuracy which is obtained after applying the proposed algorithm and '*NoSelection*' denotes the accuracy that is obtained without a feature selection procedure.

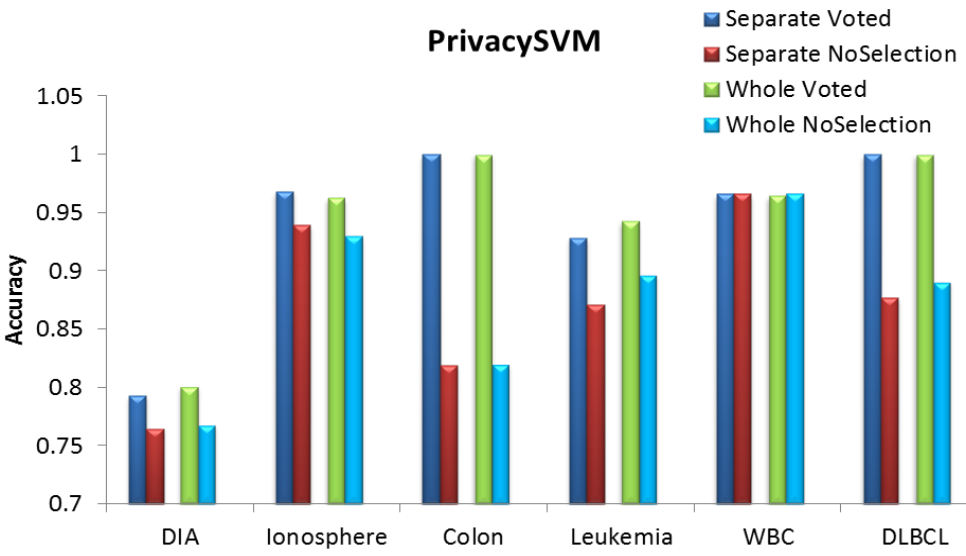


Figure 5.2. Performance improvement achieved after feature selection via PAN-SVM.

From Figure 5.2 and Table 5.2, we can observe that the classification accuracy of PAN-SVM is significantly improved after executing the proposed feature selection algorithm, especially for the *Colon*, *DLBCL* and *Leukemia* microarray data, and improvements are 18.08%, 12.24% and 5.72% under CV2 testing scenario, and 18%, 10.95% and 4.64% under CV1 testing situation. The classification accuracy is also improved for datasets *DIA* and *Ionosphere*, and they are 2.86% and 2.91%, 3.37% and 3.27% for CV2 and CV1, respectively. There is no improvement for *WBC* datasets under CV2 and a slight sacrifice under CV1. The results indicate that PPFSVM works

better for microarray datasets, which always include small sample size and much higher gene number. From Figure 5.2 and Table 5.2, we can also observe that under CV2 test situation, the classification accuracy can be improved slightly higher by PAN-SVM, when compared with the total improvements added from each dataset, but there is no significant difference (41.82% vs. 39.99% in total) between the improvements obtained under CV1 and CV2.

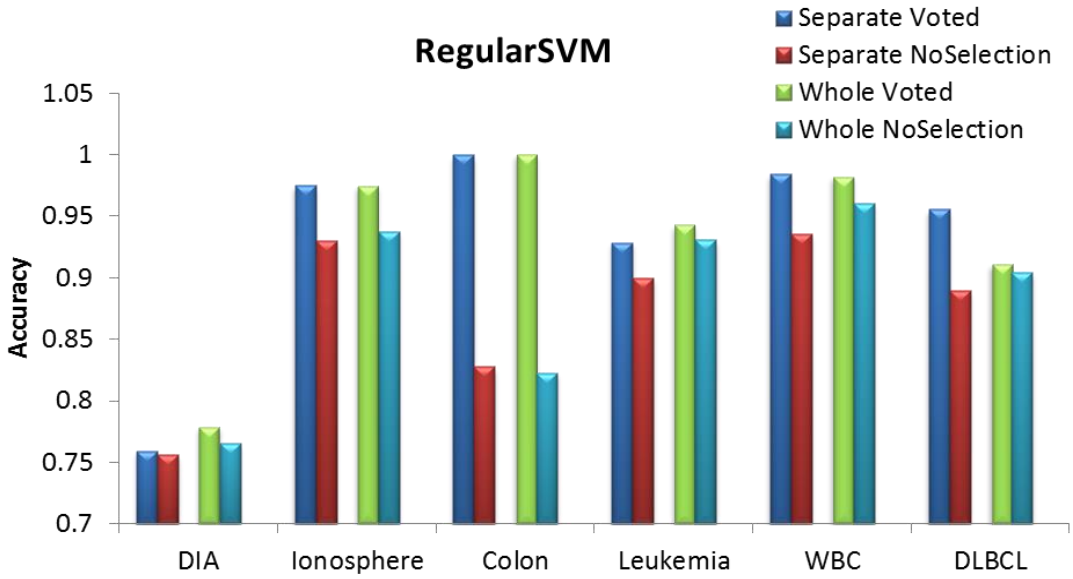


Figure 5.3. Performance improvement achieved after feature selection via LIBSVM.

Figure 5.3 and Table 5.3 also show that LIBSVM can also achieve higher predicting accuracy after executing the proposed feature selection workflow, and can significantly improve the classification performance after selecting informative features or genes. Moreover, LIBSVM can obtain much higher performance improvement under CV2 testing environment than that under CV1 situation; it is 36.32% vs. 26.76% in total when adding all of the improvements from each dataset. The reason is because LIBSVM can achieve slight higher prediction accuracy under CV1 than that under CV2 before a feature selection procedure, which indicates that LIBSVM has the problem of over-fitting, whereas, PAN-SVM has no such a problem; therefore it may say that

PAN-SVM can reduce or avoid the risk of over-fitting when compared with regular SVM. Details about PAN-SVM can be found from the previous work described in Chapter 3 and [43].

The results shown in Figure 5.2, Figure 5.3, Table 5.2 and Table 5.3 indicate that the proposed algorithm or workflow is workable and feasible, and more importantly it works efficiently and can significantly improve the classification performance by selecting informative features or genes no matter for LIBSVM or privacy preserving classifier of PAN-SVM.

Table 5.3. Comparison of classification accuracy (%) between before and after feature selection via LIBSVM.

LIBSVM	***Separate (CV2)***			***Whole (CV1)***		
Datasets	*Voted*	*NoSelection*	*Improvement*	*Voted*	*NoSelection*	*Improvement*
DIA	75.95	75.63	0.32	77.91	76.54	1.37
Ionosphere	97.50	93.04	4.47	97.50	93.78	3.72
Colon	100.00	82.79	17.21	100.00	82.30	17.70
Leukemia	92.86	90.00	2.86	94.29	93.10	1.19
WBC	98.41	93.54	4.87	98.23	96.10	2.13
DLBCL	95.56	88.95	6.61	91.11	90.46	0.65
SUM	560.27	523.94	**36.32**	559.04	532.28	**26.76**

5.3.4 Comparison with Other Methods

5.3.4.1 Classification Accuracy Improvement

We firstly conducted our experiments on the six benchmark datasets and compared some of the results obtained by the proposed algorithm in this chapter with those obtained by other state-of-the-art methods, such as Fisher-SVM, FSV, RFE-SVM and KP-SVM [74, 78]. The accuracies obtained from these four methods shown in Table 5.4 are cited from [78]. *DIA, WBC,* and *Colon* are three common datasets which are used as benchmark datasets in the paper [78] and in the current work.

There is no privacy preserving issue or testing scheme in [78], therefore, we can compared our experimental results conducted via regular SVM under CV1 test situation, which are shown

in the last column in Table 5.4, from which we can observe that the proposed method of PPFSVW outperforms the other methods for all of the three datasets *DIA, WBC, and Colon.* Besides, experimental results obtained by LIBSVM under CV2 and by PAN-SVM are also listed in Table 5.4 for a better comparison, and the results show that the proposed algorithm in this chapter outperforms all the other four state-of-the-art methods.

Table 5.4. Classification accuracy after feature selection achieved by different methods.

Datasets	**Fisher SVM**	**FSV**	**RFE SVM**	**KP SVM**	**Privacy SVM (CV2)**	**Privacy SVM (CV1)**	**Regular SVM (CV2)**	**Regular SVM (CV1)**
DIA	76.42	76.58	76.56	76.74	79.35	80.13	75.95	**77.91**
WBC	94.7	95.23	95.25	97.55	96.64	96.46	98.41	**98.23**
Colon	87.46	92.03	92.52	96.57	1.00	1.00	1.00	**1.00**

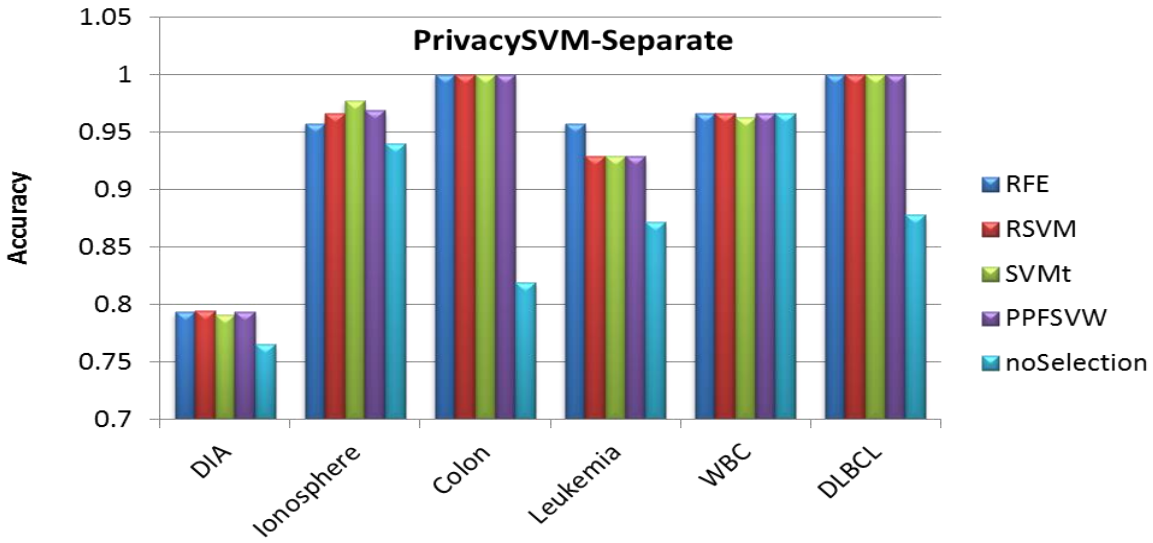


Figure 5.4. Comparison of classification accuracy achieved by PAN-SVM under CV2.

Furthermore, we also conducts our experiments on the six benchmark datasets described in Table 5.1 and compares the results obtained by PPFSVW with those obtained by SVM-RFE, RSVM, and SVM-t. The classification accuracies achieved by different methods under two test

scenarios are shown as in Figure 5.4, Figure 5.5, Figure 5.6 and Figure 5.7 in the form of bar charts and the results of accuracy improvement achieved by these four methods are shown in Table 5.5, Table 5.6, Table 5.7 and Table 5.8, respectively.

Table 5.5. Accuracy improvement achieved by different methods via PAN-SVM under CV2.

	SVM-RFE	**RSVM**	**SVM-t**	**PPFSVW**
DIA	2.86%	2.99%	2.60%	2.86%
Ionosphere	1.77%	2.63%	3.77%	2.91%
Colon	18.08%	18.08%	18.08%	18.08%
Leukemia	8.57%	5.72%	5.72%	5.72%
WBC	0.00%	0.00%	-0.35%	0.00%
DLBCL	12.24%	12.24%	12.24%	12.24%
Sum	43.53%	41.66%	42.06%	41.82%

From Figure 5.4 and Table 5.5, we can observe that all of the four methods, SVM-RFE, RSVM, SVM-t, as well as the proposed method of PPFSVM in this chapter can significantly improve the classifier predicting performance after executing the feature selection procedure for most datasets except the WBC dataset. However, different methods have different behaviors when working with various datasets. For example, the method of SVM-t works better on *Ionosphere* dataset, but achieves worse classification results when compared with the other three methods on the microarray datasets, which contain much more features, and fails to improve the classifier's predicting performance for *WBC* datasets. RFE, RSVM, and PPFSVW can achieve almost the same level accuracy improvement for *DIA*, *Colon*, *WBC* and *DLBCL* datasets, but slightly lower for *Ionosphere* dataset and higher on *Leukemia* data. Compared with the total sum improvement on all datasets, RFE-SVM defeats all other three feature selection methods benefiting from its higher improvement on the *Leukemia* data.

The results in Figure 5.5 and Table 5.6 show the classification performance and comparison of classification accuracy improvements that have been achieved by SVM-RFE, RSVM, SVM-t

and PPFSVM under CV1 test situation using a separate testing sample set. These results indicate a similar pattern made by these four feature selection methods via PAN-SVM under CV1 to that under CV2. All of these four methods can improve the classifier's ability to predict unknown samples, for *DIA*, *Ionosphere*, *Colon*, *Leukemia* and *DLBCL* datasets, and achieve a significant improvement for microarray datasets. The performance improvement has no significant difference among these four methods.

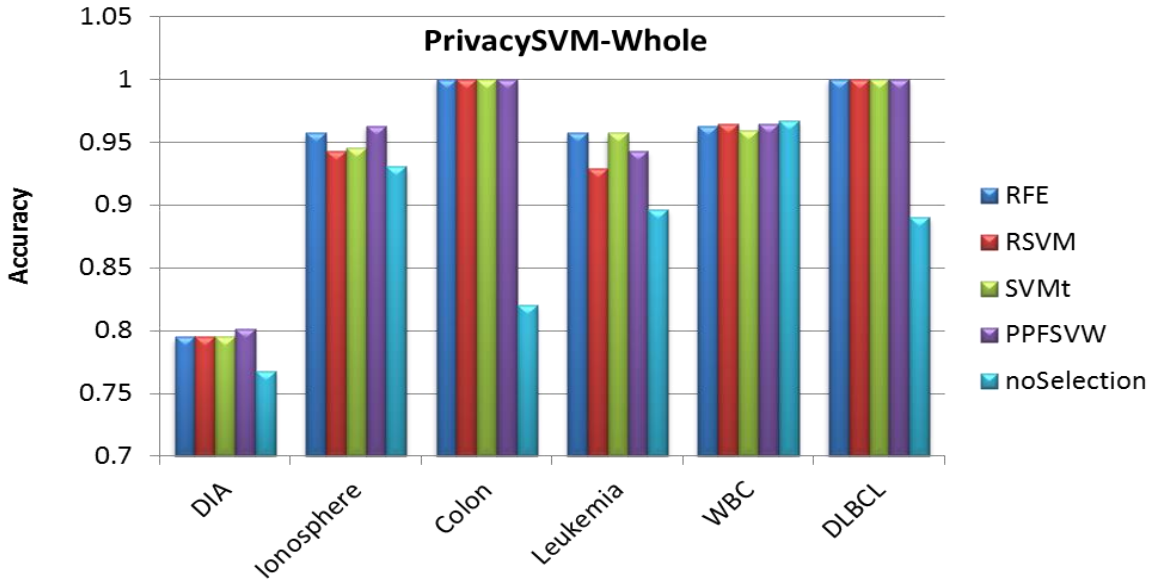


Figure 5.5. Comparison of classification accuracy achieved by PAN-SVM under CV1.

Table 5.6. Accuracy improvement achieved by different methods via PAN-SVM under CV1.

	RFE	**RSVM**	**SVM-t**	**PPFSVW**
DIA	2.71%	2.71%	2.71%	3.37%
Ionosphere	2.69%	1.27%	1.55%	3.27%
Colon	18.00%	18.00%	18.00%	18.00%
Leukemia	6.06%	3.21%	6.06%	4.64%
WBC	-0.41%	-0.23%	-0.76%	-0.23%
DLBCL	10.95%	10.95%	10.95%	10.95%
Sum	40.01%	35.91%	38.52%	39.99%

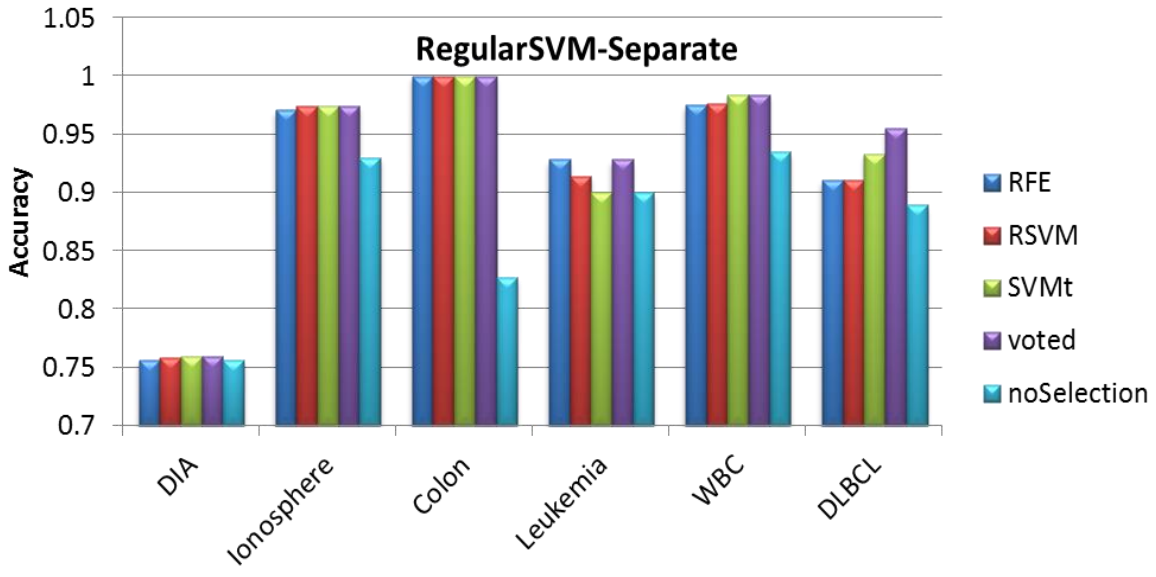


Figure 5.6. Comparison of classification accuracy achieved by LIBSVM under CV2.

Table 5.7. Accuracy improvement achieved by different methods via LIBSVM under CV2.

	RFE	**RSVM**	**SVM-t**	**PPFSVW**
DIA	0.05%	0.18%	0.32%	0.32%
Ionosphere	4.11%	4.47%	4.47%	4.47%
Colon	17.21%	17.21%	17.21%	17.21%
Leukemia	2.86%	1.43%	0.00%	2.86%
WBC	3.99%	4.16%	4.87%	4.87%
DLBCL	2.16%	2.16%	4.38%	6.61%
Sum	30.37%	29.61%	31.24%	36.32%

Figure 5.6 and Figure 5.7 show the comparison of classification accuracies by using LIBSVM as the classifier for SVM-RFE, RSVM, SVM-t and the proposed algorithm workflow in this chapter. Table 5.7 and Table 5.8 show accuracy improvements achieved by these four different methods via employing LIBSVM. Each method is tested under CV1, and CV2 testing mode and the accuracies at each iteration step are obtained by 5-fold cross-validation, as well as the final accuracy using the series of selected features.

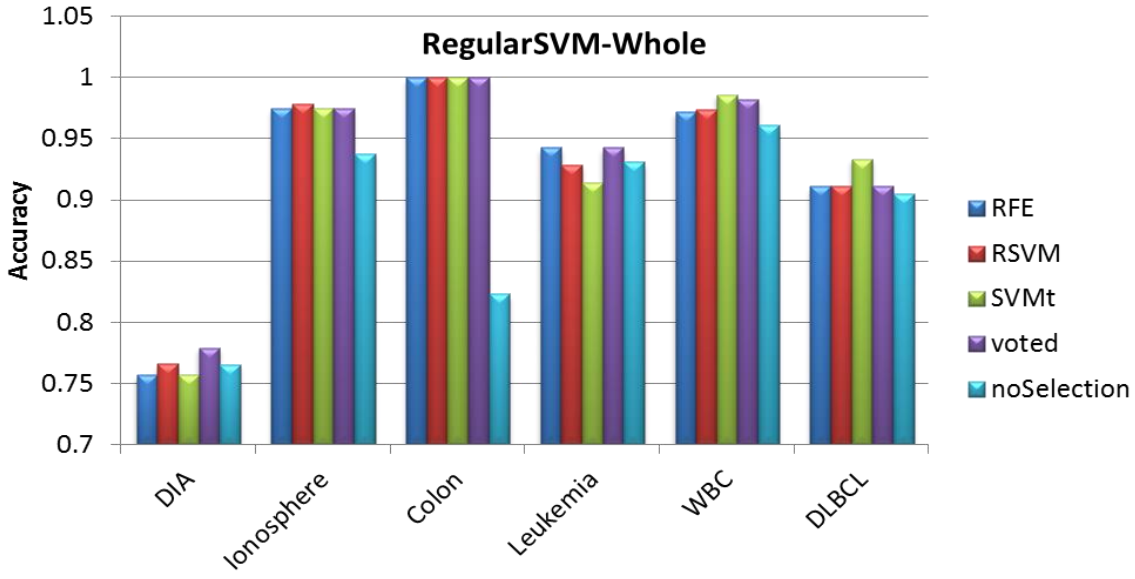


Figure 5.7. Comparison of classification accuracy achieved by LIBSVM under CV1.

Table 5.8. Accuracy improvement achieved by different methods via LIBSVM under CV1.

	RFE	**RSVM**	**SVM-t**	**PPFSVW**
DIA	-0.85%	0.07%	-0.85%	1.37%
Ionosphere	3.72%	4.08%	3.72%	3.72%
Colon	17.70%	17.70%	17.70%	17.70%
Leukemia	1.19%	-0.24%	-1.67%	1.19%
WBC	1.07%	1.25%	2.49%	2.13%
DLBCL	0.65%	0.65%	2.87%	0.65%
Sum	23.47%	23.50%	24.26%	26.76%

From Figure 5.6, Figure 5.7, Table 5.7 and Table 5.8, we can observe that all of these four methods perform better under CV2 testing environment, which is similar to PAN-SVM. However, the overall improvement achieved by LIBSVM under CV2 is much higher than that under CV1 when summarizing all of the improvements together (as shown in the last line in Table 5.7 and Table 5.8) than PAN-SVM, the reason is probably because the ability of PAN-SVM to reduce overfitting, and the regular SVM cannot achieve higher or same level of predicting accuracy for separating testing samples under CV2 scenario. Besides, when compared all the overall improvements (as shown in the last line in Table 5.7 and Table 5.8) achieved by

these four different methods, we can observe that the proposed workflow can make the classifier achieve higher classification accuracy than the other three methods and have significant improvements, no matter under CV1 or CV2 test environment. In other word, the proposed workflow in this chapter works better for regular SVM and can preserve individual privacy when employing PAN-SVM as the classifier.

5.3.4.2 Number of Selected Features

We also compare the selected number of features by these four methods on datasets *DIA*, *Ionosphere*, *Colon*, *Leukemia*, *WBC* and *DLBCL*, and only show the results achieved by PAN-SVM under CV2 test situation. The results are represented as curves in Figure 5.8 and the detail descriptions are shown in Table 5.9, from which we can observe that PPFSVW can make the classifier achieve the highest predicting accuracy for *DIA* dataset by the top 5 features (with accuracy 79.35%), which is the same as SVM-RFE and is fewer than 7 (79.48%) and 8 (79.09%) for RSVM and SVM-t, respectively. For the *Colon* data, the classifier can achieve the best classification performance with top 53 features (with accuracy 100%) after conducting PPFSVW algorithms, but 63 (accuracy 100%), 61 (accuracy 100%) and 617 (accuracy 100%) for RFE-SVM, RSVM, and SVM-t, respectively.

Table 5.9. Selected feature number by different methods.

	DIA	**Ionosphere**	**Colon**	**Leukemia**	**WBC**	**DLBCL**
RFE	5 (79.35)	12 (95.71)	63 (100)	4565 (95.71)	18 (96.64)	114 (100)
RSVM	7 (79.48)	12 (96.57)	61 (100)	6380 (92.86)	17 (96.64)	147 (100)
SVM-t	8 (79.09)	10 (97.71)	617 (100)	5420 (92.86)	21 (96.28)	166 (100)
PPFSVW	5 (79.35)	17 (96.86)	53 (100)	4826 (92.86)_	11 (96.64)	121 (100)

For *WBC* data, the number of best-selected feature subset is 11 (accuracy 96.64%), which is fewer than 18 (accuracy 96.64%), 17 (accuracy 96.64%) and 21 (accuracy 96.28%) for SVM-RFE, RSVM, and SVM-t, respectively.

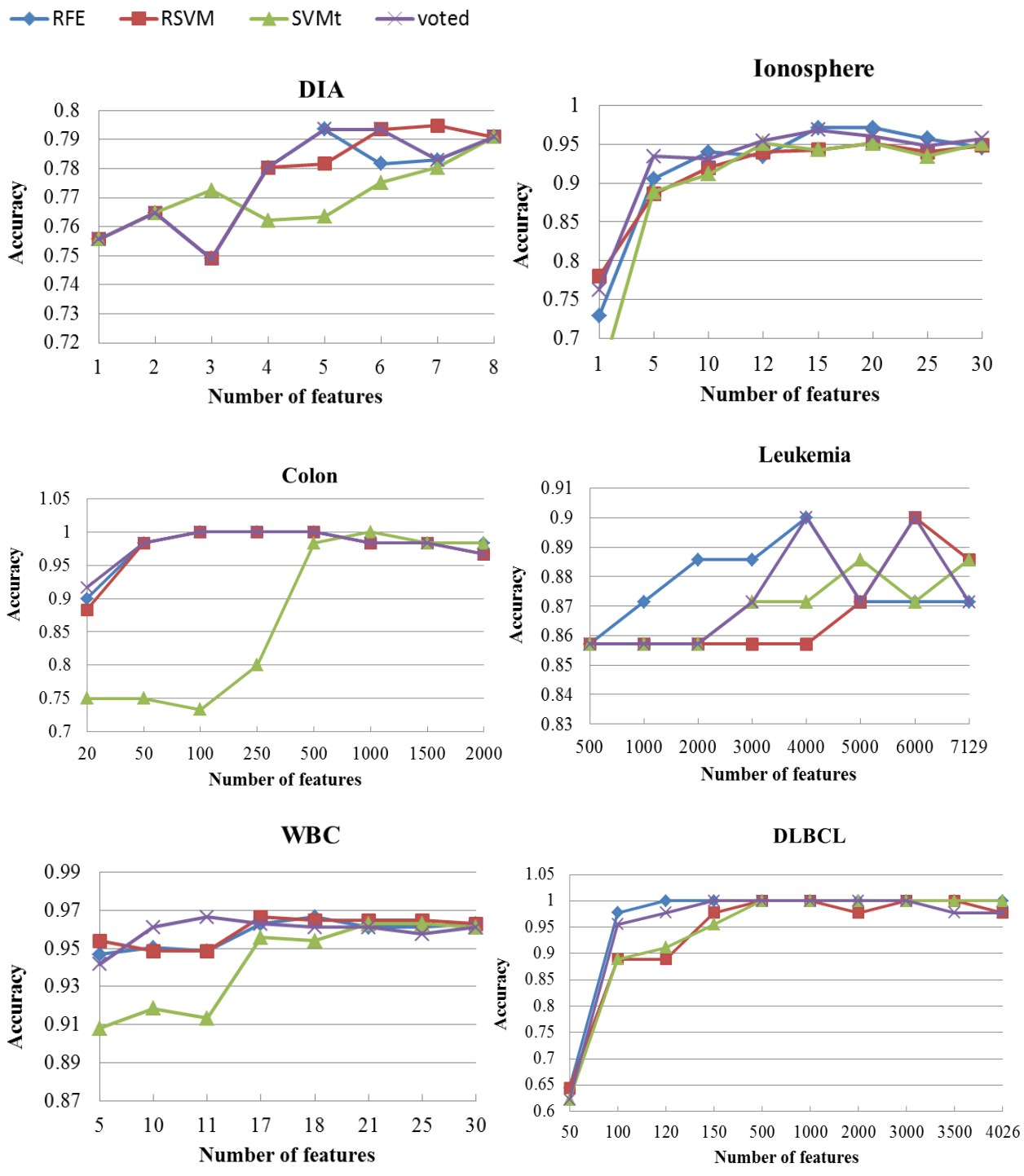


Figure 5.8. Comparison of accuracy as the number of selected features increases.

For the three datasets of *DIA*, *Colon*, and *WBC*, PPFSVW cannot only select fewer features but also keep the classifier with higher or almost same level of classification accuracy. For the *Ionosphere* data, although PPFSVW selects a subset of features with a larger number than other methods, it makes the classifier achieve the highest classification performance. For the *Leukemia* and *DLBCL* data, PPFSVW is defeated by SVM-RFE but still works better than RSVM and SVM-t with fewer features but the same level of accuracy.

From these results, we can conclude that PPFSVW can make the classifier achieve higher or same level of classification performance with fewer features, especially when compared with RSVM and SVM-t. The selected six datasets are different at their sample sizes and feature numbers; the other three existing sophisticated methods outperforms each other on different datasets, but PFSVW can always make the classifier achieve competitive results compared with the other three, which indicate that PPFSVW is much stable and robust.

5.4 Conclusions

In this chapter, we proposed a privacy preserving feature selection method (PPFSVW) via integrating three popular wrapper methods in the way of voting at feature eliminating phase. PPFSVW is based on our previous work of PAN-SVM, which is a privacy preserving framework for binary classification on SVM; therefore, PPFSVM inherits the privacy preserving property of PAN-SVM and can protect individual privacy during the procedure of feature selection. The privacy preserving strategy for distributed data is needed as the privacy concerns increase rapidly nowadays.

PPFSVW shares the common workflow with RFE-SVM, RSVM, and SVM-t, but different from them at the step of choosing to be eliminated feature at each iteration. It combines the three criteria used by these three methods, and votes to be eliminated one. If eliminating feature cannot

be decided by voting, PPFSVW will construct classifiers and compare the negative affection to classifiers which caused by those temporarily selected three features, and the one with highest negative affection will be eliminated at this iteration.

The feasibility and performance of the proposed workflow are assessed on six benchmark datasets, including three microarray datasets, and they are different at sample size and feature numbers. The experiments are also conducted under two different testing situations, CV1 and CV2. Our experimental results indicate that the proposed algorithm workflow can work effectively to improve the classification performance regarding accuracy via selecting informative features and genes, for both PAN-SVM with privacy preserving consideration and LIBSVM without privacy consideration under CV1 and CV2. Besides, PPFSVW outperforms other state-of-the-art feature selection methods of Fisher-SVM, FSV, RFE-SVM and KP-SVM [74, 78] for *DIA*, *Ionosphere* and *Colon* datasets. Furthermore, we also conducted the proposed workflow on PAN-SVM and LIBSVM and compared their classification accuracies with those obtained from SVM-RFE, RSVM, and SVM-t. The experimental results show that PPFSVW has no significant difference from these three methods when employing PAN-SVM, but works better when conducting on LIBSVM. The reason for this is because of the stability and ability of PAN-SVM to reduce the risk of over-fitting. In addition, our experimental results also show that PPFSVM can make the classifier achieve higher or same level classification accuracy with fewer features when compared with SVM-RFE, RSVM and SVM-t.

6 PRIVACY PRESERVING FEATURE SELECTION VIA INTEGRATING FILTER AND WRAPPER METHODS FOR HORIZONTALLY DISTRIBUTED DATASETS

6.1 Introduction

Nowadays, a lot of scientific fields have experienced a huge growth in data volume and data complexity, which brings to data miners lots of opportunities, as well as many challenges. For example, how to mine distributed data and meanwhile preserve individual information as the growing concerns of privacy issue? Recently, assembling data from distributed parties has become increasingly common [63-65], since applying data mining techniques on the aggregated data can build much more reliable prediction models and attain useful patterns from a wider picture, which can benefit from medical research, improving customer service and homeland security, etc. However, this might divulge the sensitive information about individuals. It thus leads to increased concerns about privacy during the process of data mining, which in turn prevents different parties from sharing information.

Besides, data mining tasks often suffer from the curse of high dimensionality of data attributes or features. How to effectively select relevant and informative features and solve the issue of dimensionality? For example, microarray data have been widely used to investigating a lot of biological questions, such as gene expression in different situations, classification of diseases; however, gene selection is still a challenging task in the tumor-related classification. Since the gene expression data usually contain thousands of genes, but only decades or hundreds of samples. Feature selection techniques address the issue of dimensionality reduction by selecting an available subset of features via predetermined selecting criteria. It is usual a pre-processing procedure which aims at speeding up learning the process and decrease the space complexity of classifier. Besides, the selected informative features, such as the genes, are very important in

biological research to help diagnose cancer or special diseases. Feature selection, therefore, plays a vital role in optimizing the mining procedure and selecting informative and relevant attributes.

Many sophisticated methods [79-82] have been employed to find very important genes; however, traditional filter methods of feature selection may help to find dependent genes, but the accuracy cannot be guaranteed, which is much more important to biological data. Wrapper methods of feature selection can guarantee the classification accuracy, but they are usually time-consuming. Distributed feature selection by sharing data from multiple parties can be a solution to the issue of small size, but privacy concerns increase. Many feature selections related to data mining tasks have been proposed as data are integrated into a central location, while the privacy concerns of sharing data by distributed parties bring a great challenge to feature selection.

In this chapter, a Privacy Preserving Feature Selection method via Integrating Filter and Wrapper methods for horizontally distributed data (PPFSIFW) [83] is proposed. Details about PPFSIFW are presented in *Method* section. PPFSIFW is assessed and tested on six datasets, including 3 gene expression datasets, and the results are addressed in the *Results and Discussion* section, and followed by the conclusion at last in the *Conclusion* section.

6.2 Methods

6.2.1 Existing Filter Feature Selection Methods

According to different selecting strategies and procedures of algorithms, feature selection methods can be formulated into three main categories: *filter*, *wrapper* and *embedded* approaches [66]. The *filter methods* usually take account of the statistical properties of features and rank them according to some criteria of relevant information. This step is always before the classification step and is entirely independent of data mining algorithms. Therefore, they are fast, and the effects of the subset of features on the performance of the mining algorithms will also be

avoided. *Chi-square, Fisher Criterion Score* [84], *Welch*-t-*test* [85] and *Between versus Within Class Scatter Ratio* [86] are some measurements that are usually used to filter a subset of features. The importance of each feature will be calculated according to these measurements, and least important ones will be discarded.

The respective ***Fisher score*** [84] to calculate the importance of each feature can be formulated by equation (6.1):

$$F(j)=\left|\frac{\mu_j^+-\mu_j^-}{\left(\sigma_j^+\right)^2+\left(\sigma_j^-\right)^2}\right| \tag{6.1}$$

Where μ_j^+ and μ_j^- represent the mean value of the j^{th} feature in the positive class (with label +1) and negative class (with label -1), respectively, and σ_j^+ and σ_j^- stand for the standard deviations of the j^{th} feature in the positive and negative class, respectively.

The ***welch- t-statistics*** [85] is another filter criterion, as denotes in the equation (6.2).

$$\left|t_j\right|=\frac{(\mu_j^+-\mu_j^-)}{\sqrt{((s_j^+)^2/n^+)+((s_j^-)^2/n^-)}} \tag{6.2}$$

Where $n^+$and n^- denotes the number of support vectors for the positive class (+) and negative class (-), respectively. μ_j^+ and μ_j^- indicate the means of the j^{th} feature in class+ and class-, and $s_j^+$and s_j^-represent the standard deviations of the j^{th} feature in class+ and class-, respectively.

The ***Between versus Within Class Scatter Ratio*** [86] is another popualr filter method for feature selection procedure, and it is formulated as in equation (6.3):

$$\begin{aligned} S_w &= \sum_{i=1}^{c}\sum_{j}^{n_c}(x_j-\mu_i)*((x_j-\mu_i)^T) \\ S_b &= \sum_{i=1}^{c}(\mu_i-\mu)*((\mu_i-\mu)^T) \\ S &= \frac{S_b}{S_w} \end{aligned} \tag{6.3}$$

Where S_w and S_b denote the within class and between class scatter matrix, respectively, and S is the *within versus between class scatter ratio*, it is a vector with n (number of features) elements. c equals to the number of class; here c equals to 2 for binary classification. x_j represents the j^{th} record in the i^{th} class, n_c denotes the number of samples in the i^{th} class; μ_i denotes the mean of the i^{th} class, and μ denotes overall mean in these c classes.

Just as the name implies, the *wrapper methods* often wrapped the feature selection step in the process of mining algorithms. Compared with the filter methods, wrapper methods have the advantages of taking the performance of mining algorithms into account. Thus a better classification model will be constructed. However, it needs to repeatedly train and test the data and build classification model at each step when a subset of features are selected; the computational complexity thus increased sharply. Some popular wrapper methods can be found in [59, 70, 74]. The third kind of feature selection approaches is named *embedded method*, which performs feature selection in the process of the constructing data mining model by adding or modifying the optimizing process of classification, for examples in [71, 72].

Besides, feature selection algorithms can also be classified into two categories based on the relationship of features: *feature ranking* and *subset selection*. In the ranking list, the importance of each gene is unequal. Usually the most top one is supposed to be the most important one, and so forth; while in the subset selection, each feature is equal, they work together making the classifier obtain the best performance.

6.2.2 PAN-SVM Classifier

As mentioned above, wrapper methods usually integrate a feature selection step in the process of mining algorithms. For a classification problem, methods used for selecting features are closely related to classifiers. Features will be evaluated by the classification accuracy at each

step. In the current work, one of our previous works, a privacy-preserving framework named Privacy-Aware Non-linear SVM for Multi-source Big Data (PAN-SVM in brief) [43] is used as the binary classifier for preserving privacy during the step of feature selection. PAN-SVM contains three layers, which can finish corresponding functions. The bottom layer protects individual data privacy, where sampled data from multiple parties will be encrypted via the *Secure Sum Protocol* [9] and sent to miner. Data are sampled by k-means clustering methods and used as landmarks [39, 44]. At the medium layer, the landmarks will be used to approximate kernel matrix via Nystrom technique [39, 44] and the computation cost of kernel matrix will be further reduced via eigenvalue decomposition method. After the step of kernel matrix approximation and decomposition, the non-linear separable SVM will be converted a linear separable one in this layer. Linear SVM will be optimized and speeded up by linear search and cutting plane techniques [42] at the top layer. Although the classification accuracy sacrifices slightly when compared regular SVM, like LIBSVM [87], the individual private information is preserved; furthermore, the training process is speeded up when compared with other distributed classification methods. Details about PAN-SVM can be found from [43].

6.2.3 Workflow of PPFSIFW

In this chapter, the proposed privacy preserving feature selection algorithm PPFSIFW [83] integrates three filter methods of *Fisher score*, *welch-t-test* and *between versus within class scatter ratio*, uses PSN-SVM as classifier and the classification accuracy as a wrapper method. It follows the workflow as following:

Step 1: Calculate the three measurements mentioned in (6.1), (6.2) and (6.3).

Step 2: Choose features that selected by all of the three measurements and remove the rest ones. The feature will be selected if it meets the thresholds of the three measurements. We

call the selected feature set at this step as ***voted feature set***. A large number of features can be eliminated at this step depending on the thresholds that user defines.

Step 3: Ranking the selected features in the voted set (Wrapper method).

1) Computer the overall classification accuracy with the selected features by filter methods, denoted as *overall_acc*.
2) Compute the classification accuracy for each classifier (PAN-SVM) that without feature *i* (*i = 1…k*), for data with *k* features, there will be *k* classifiers, denotes as *ith_acc*.
3) Rank the features by accuracies obtained in previous step.
4) Remove a feature in this way: suppose *ith_acc* is the highest accuracy among the *k* accuracies obtained in 2), if *ith_acc* > *overall_acc*, then remove feature *i*, since it increases the classification accuracy by removing it. Otherwise, if there is no feature increases accuracy, a local maxima accuracy of *ith_acc* is obtained, then set *overall_acc* = *ith_acc* and remove feature *i*, since it gives the classifier highest negative affection.
5) Repeat the ranking step

Step 4: Return a ranked list of features in the voted set.

In step 2, user can define his own threshold to decide the approximate number of features he wants to keep or eliminate. Such as, when using the *Welch-t-test* measurement, features that have p-value larger than 0.01 will be eliminated. Therefore, a large number of features can be removed at this step, only a small size keeps left. In step 3, the highest negative affection can be understood in this way. For example, if removing feature *i*, the classifier can achieve 98% classification accuracy, while removing feature *j*, the accuracy is 90%, it says *i* has higher negative affection to classifier than feature *j*, so feature *i* will be eliminated at this step. 5-fold

cross validation is used at each iteration for every classifier. PPFSIFW returns a ranking list of the selected features by filter methods.

6.3 Results and Discussions

6.3.1 Datasets

PPFSIFW is assessed and tested on six benchmark datasets including 3 microarray datasets with a different number of features, details about these datasets are shown as in Table 6.1.

Table 6.1. Description of testing datasets.

Dataset	**# of samples**	**# of features**	**C**	**γ**
Diabetes (DIA)	768	8	512.0	0.0078125
Ionosphere	351	34	8.0	0.5
Colon	62	2000	32.0	0.0078125
Leukemia	72	7129	128.0	0.0001220703125
Lymphoma	47	4026	2.0	0.0078125
Breast Cancer (WBC)	569	30	128.0	8.0

The *Diabetes* and *Ionosphere* data are downloaded from LIBSVM repository [87], the Wisconsin Breast Cancer data (WBC) is downloaded from University of California, Irvine (UCI) Machine Learning Repository [48], and the microarray datasets of *Leukemia*, *Lymphoma,* and *colon* [57] are download from [58]. C and γ are penalty parameter for SVM and a free parameter for Radial Basis Function kernel (RBF) used in SVM. They are generated by 10-fold cross validation.

6.3.2 Performance Assessing

The Cross Validation (CV) method is often used to assess the performance of classifier according to the classification due to lack of data that can be utilized as separate testing samples. During the cross-validation process, data will be randomly split into *k* (*k*-fold) subsets, and at

each training round, *k-1* subsets are used as training data, and the left *1* subset is used as testing set. However, as pointed by [59], the feature selection results might vary due to even a single difference in the training set, especially for small datasets. Many feature selection methods are done with all samples, and the cross-validation step is only done during the classification process, which makes the feature selection external to the cross-validation procedures, and leads to 'information leak' in the feature selection step. It calls this kind of error made by cross-validation as a CV1 error. [60-62] also points out that CV1 error may severely bias the evaluation of feature selection. [59] also demonstrates the existing of the bias via simulation data and suggests another error evaluation method, named CV2. Under the CV2 scenario, a separate dataset will be used as test samples and leaves out of training set before any feature selection step. The performance of PPFSIFW will be assessed by the measurement of classification accuracy, which is formulated by the equation (6.4):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.4}$$

Where *TP* represents *True Positive*, *TN* denotes *True Negative*; *FP* means *False Positive* and *FN* states *False Negative*.

6.3.3 Effectiveness of PPFSIFW

In order to check whether the proposed feature selection algorithm (Feature Selection via Integrating Filter and Wrapper method, FSIFW in brief) is workable or not and its effectiveness, the proposed workflow mentioned in section 6.2.3 is applied to PAN-SVM [43] and LIBSVM [77], under CV1 and CV2 testing scenario, respectively.

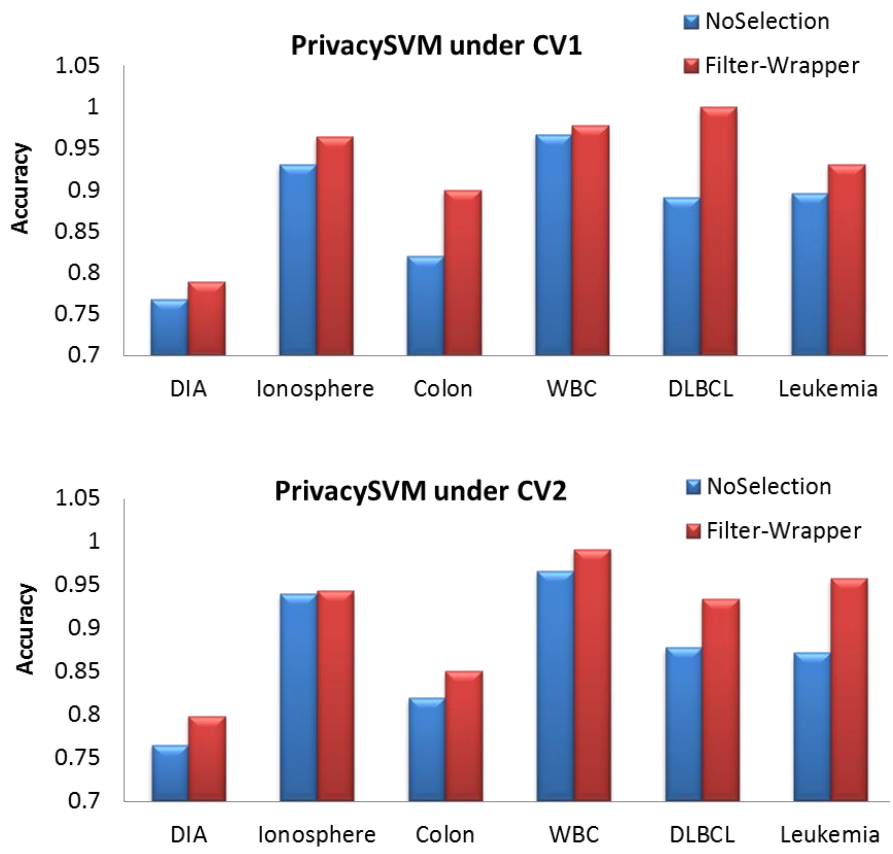


Figure 6.1. Classification accuracy improvement of PAN-SVM by FSIFW.

The comparisons of classification accuracy before and after FSIFW are shown in Figure 6.1 and Figure 6.2, respectively. In both figures, the '*NoSelection*' in blue bar denotes the overall classification accuracy with all features (without feature selection step), and '*Filter-Wrapper*' in *red bar* represents accuracy that improved by FSIFW (with the proposed feature selection step). '*PrivacySVM*' denotes PAN-SVM, we used PAN-SVM as the classifier to preserve individual privacy, and '*RegularSVM*' means no privacy aware, we use the popular SVM package of LIBSVM [77] as a regular SVM as a classifier.

It can be seen from these figures, the classification accuracy of PAN-SVM is improved after executing the proposed algorithm of FSIFW for all of the six datasets no matter under which testing schema. The classification accuracy of LIBSVM is also improved for most of the cases,

except for the *Leukemia* data under the CV1 testing scenario. As mentioned in the previous chapters, LIBSVM has the issue of overfitting, especially for the small dataset. It can achieve higher classification accuracy under CV1, which is why there is no obvious improvement after the feature selection step.

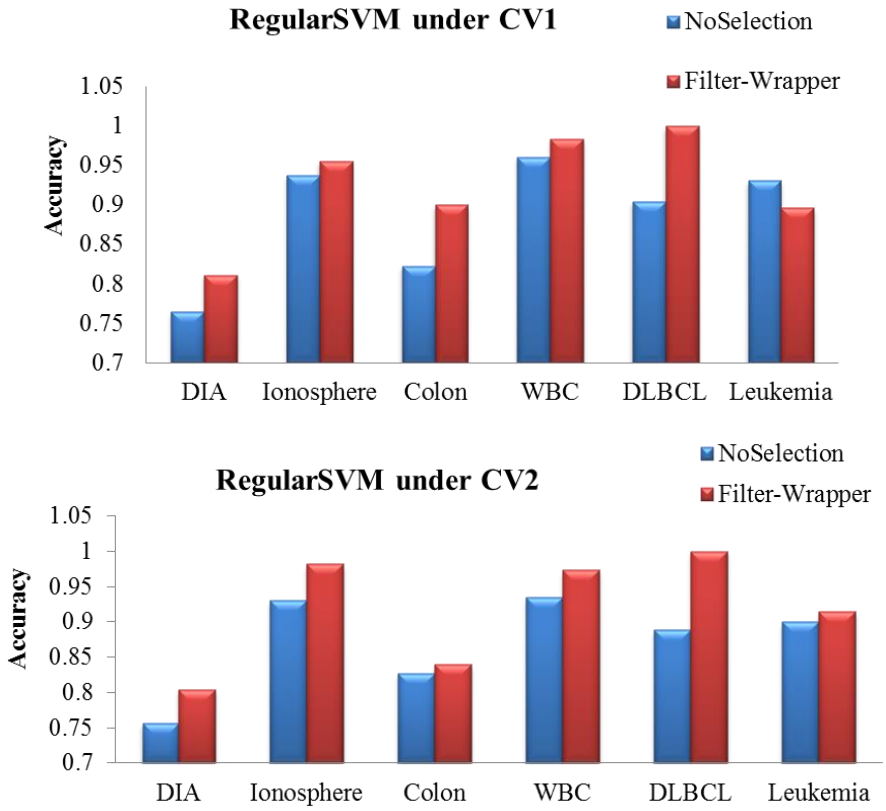


Figure 6.2. Classification accuracy improvement of LIBSVM by FSIFW.

Table 6.2. Classification accuracy improvements.

	PrivacySVM CV2	***PrivacySVM CV1***	***RegularSVM CV2***	***RegularSVM CV1***
DIA	3.39%	2.10%	4.86%	4.56%
Ionosphere	0.35%	3.42%	5.18%	1.78%
Colon	3.08%	8.00%	1.21%	7.70%
WBC	2.47%	1.12%	3.82%	2.26%
DLBCL	5.57%	10.95%	11.05%	9.54%
Leukemia	8.57%	3.45%	1.43%	-3.45%
Sum	23.43%	29.04%	27.55%	22.39%

Table 6.2 also lists the improvements in accuracy achieved by PAN-SVM and LIBSVM under CV1 and CV2 scenario, from which we can observe that the classification accuracy can be improved as high as 11.05%, which demonstrates that the proposed algorithm of FSIFW is not only workable but also effective to enhance the classifier's ability to predict unknown samples.

6.3.4 Comparison with Other Methods

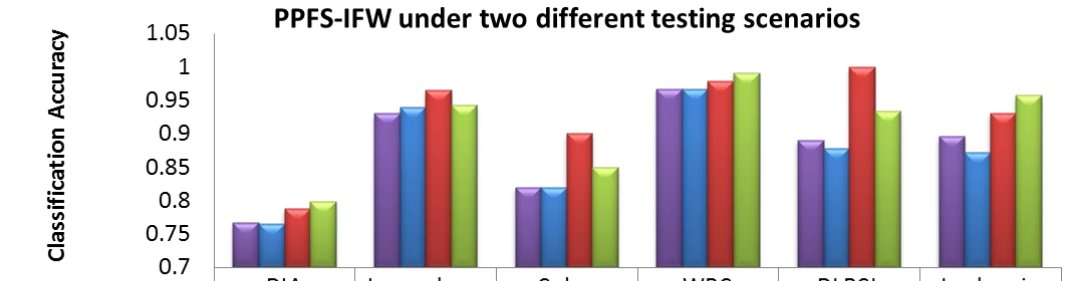
6.3.4.1 Performance Comparison under CV1 and CV2

Further comparison between the classification accuracies made by PAN-SVM is conducted to check whether there is any difference under CV1 and CV2 testing schema. The detail results are shown as in Figure 6.3 and Table 6.3, from which we can observe that before feature selection step the overall classification accuracies with all of the features are comparable for datasets *DIA, Ionosphere*, *Colon,* and *WBC*, but the classification performance of classifiers gets enhanced slightly under CV1 schema for datasets *DLBCL* and *Leukemia*.

After the feature selection step by PPFSIFW, the classification accuracies are improved significantly for all of the datasets, no matter under which testing situation, but there is no obvious pattern found. The results described in Table 6.3 show a wider picture and illustrate that the accuracy can be improved higher under CV1 test condition. It makes sense because there exists 'information leak' during the feature selection step under CV1 test schema.

Table 6.3. Classification accuracy improvement by PPFSIFW under CV1 and CV2.

	CV2	**CV1**	**CV2_# of feature**	**CV1_# of feature**
DIA	3.39%	2.10%	4	4
Ionosphere	0.35%	3.42%	2	8
Colon	3.08%	8.00%	34	157
WBC	2.47%	1.12%	10	4
DLBCL	5.57%	10.95%	394	444
Leukemia	8.57%	3.45%	537	631
Sum	23.43%	29.04%	981	1248



	DIA	Ionosphere	Colon	WBC	DLBCL	Leukemia
NoSelection_CV1	76.76%	93.03%	82.00%	96.69%	89.05%	89.65%
NoSelection_CV2	76.48%	93.94%	81.92%	96.64%	87.76%	87.14%
Selected_CV1	78.86%	96.44%	90.00%	97.81%	100.00%	93.10%
Selected_CV2	79.87%	94.29%	85.00%	99.11%	93.33%	95.71%

Figure 6.3. Classification Accuracy comparison achieved by PPFSIFW.

6.3.4.2 Comparison of Classification Accuracy

We also compare the proposed method in this chapter with other state-of-the-art methods, such as Fisher-SVM, FSV, RFE-SVM and KP-SVM [74, 78], the result data are selected from [78] and listed as in Table 6.4, where *DIA, WBC* and *Colon* are three common datasets in [78] and the current work. From Table 6.4, we can observe that the proposed method PPFSIFW outperforms the other methods for datasets *DIA* and *WBC*, which own few features and defeated by Fisher-SVM, RFE-SVM, and KP-SVM on *Colon* data with high dimensionality. The proposed feature selection algorithm of PPFSIFW is based on our previous work PAN-SVM [43], which encrypted the data to protect individual information and approximated the kernel matrix for reducing communication and computation cost, it is reasonable for accuracy sacrificed for some data.

Table 6.4. Classification accuracy comparison among different methods.

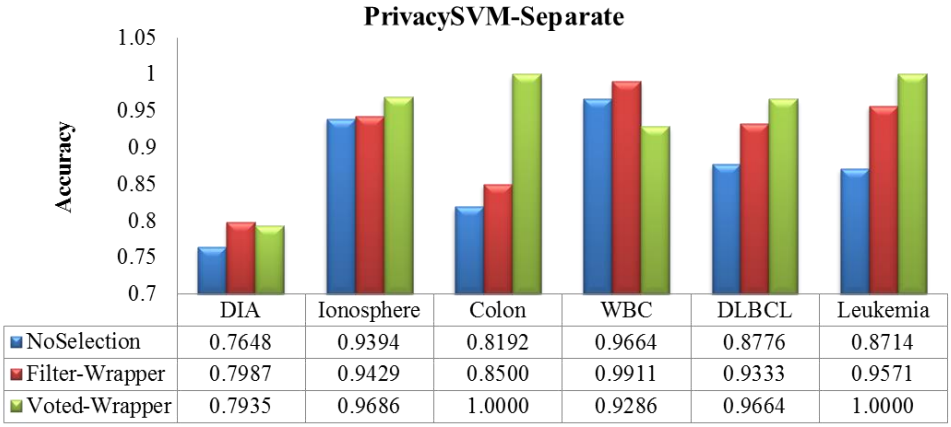
Dataset	**Fisher SVM**	**FSV**	**RFE SVM**	**KP SVM**	**PPFSIFW (CV2)**	**PPFSIFW (CV1)**
DIA	76.42	76.58	76.56	76.74	**79.87**	**78.86**
WBC	94.7	95.23	95.25	97.55	**99.11**	**97.81**
Colon	87.46	92.03	92.52	96.57	85.00	**90.00**

In chapter 5, we propose another privacy preserving feature selection via voted wrapper methods, named PPFSVW. In chapter 5, PPFSVW is assessed on six benchmark datasets and compared with other state-of-the-art methods of SVM-RFE, RSVM, and SVM-t. In this chapter, the PPFSIFW is proposed, and it shares the common classifier of PAN-SVM with PPFSVW aiming to protect individual privacy during the feature selection procedure. Different from PPFSVW, it integrates three popular filter methods of *Fisher score*, *Welch-t-test* and *between verses within class scatter ratio* together, and votes which features should be kept. The voted features then will be ranked according to their contributions to the classification accuracy of the classifiers. PPFSIFW integrates the advantages of filter methods, so it is much faster than traditional wrapper methods. In the voting step, PPFSIFW can filter and eliminate a significant number of features according to user-defined thresholds.

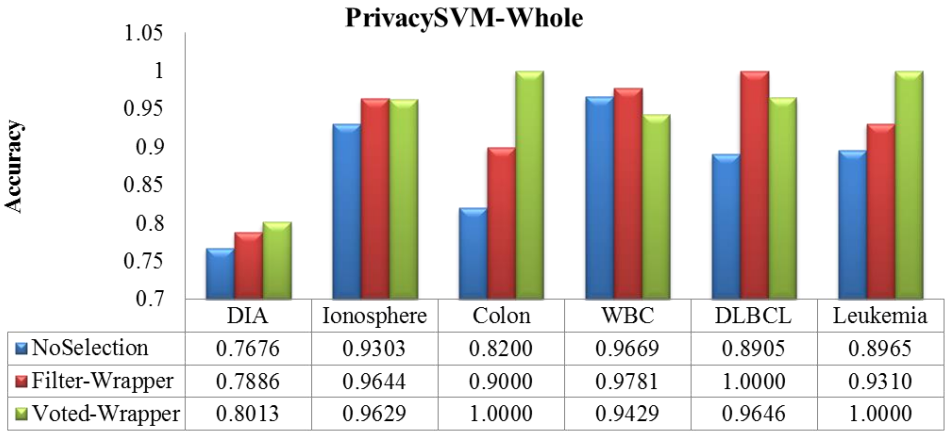
In this chapter, we compare the selected features number by PPFSVW and PPFSIFW on the six benchmark datasets; details are shown as in Figure 6.4, which shows the classification accuracy of PAN-SVM after executing PPFSVW and PPFSIFW under CV2 (top figure) and CV1 (bottom figure), respectively. We can observe that both feature selection method can significantly enhance the classifier's ability to predict, but they behave differently on different datasets under different test situations. Under CV2, PPFSIFW works better than PPFSVW on datasets *DIA, WBC,* and *Leukemia.* PPFSIFW works a little better for a dataset with enough sample size and few features, but a little poor on microarray dataset which normally contain small sample size but have high dimensionality.

Table 6.5 lists more details about the classification accuracy improvement achieved by PPFSIFW and PPFSVW under CV1 and CV2 scenarios. From Table 6.5, it can be seen clearly that PPFSVW outperforms PPFSIFW in most cases. The classification accuracy achieved by

PPFSVW can be an improvement as high as 41.82% in total under CV2 and 39.99% in total under CV1, and they are 23.43% for PPFSIFW under CV2 and 29.04% under CV1. They are much lower than those obtained by PPFSVW.



	DIA	Ionosphere	Colon	WBC	DLBCL	Leukemia
NoSelection	0.7648	0.9394	0.8192	0.9664	0.8776	0.8714
Filter-Wrapper	0.7987	0.9429	0.8500	0.9911	0.9333	0.9571
Voted-Wrapper	0.7935	0.9686	1.0000	0.9286	0.9664	1.0000



	DIA	Ionosphere	Colon	WBC	DLBCL	Leukemia
NoSelection	0.7676	0.9303	0.8200	0.9669	0.8905	0.8965
Filter-Wrapper	0.7886	0.9644	0.9000	0.9781	1.0000	0.9310
Voted-Wrapper	0.8013	0.9629	1.0000	0.9429	0.9646	1.0000

Figure 6.4. Accuracy comparison of PAN-SVM after executing PPFSIFW and PPFSVW.

Table 6.5. Classification accuracy improvement achieved by PPFSIFW and PPFSVW.

	Separate (CV2)		**Whole (CV1)**	
	IFW	*VW*	*IFW*	*VW*
DIA	3.39%	2.86%	2.10%	3.37%
Ionosphere	0.35%	2.91%	3.42%	3.27%
Colon	3.08%	18.08%	8.00%	18.00%
WBC	2.47%	0.00%	1.12%	-0.23%
DLBCL	5.57%	12.24%	10.95%	10.95%
Leukemia	8.57%	5.72%	3.45%	4.64%
Sum	23.43%	41.82%	29.04%	39.99%

6.3.4.3 Comparison of Selected Feature Number

From the previous section, we know that PPFSVW outperforms PPFSIFW in most cases, which is reasonable, since filter methods are always independent of the classifier, and they usually a pre-process procedure; whereas, the wrapper methods usually involve in the training and classification process. Thus the classification accuracy can be improved more. In this section, the selected feature numbers by PPFSIFW and PPFSVW are compared, details are shown as in Table 6.6. Here the selected features mean a subset of features that together can make the classifier achieve highest classification accuracy. From Table 6.6 we can observe that PPFSIFW usually selected a subset with fewer features in most cases under CV2 and CV1, such as for datasets *DIA, Ionosphere, Colon, WBC,* and *Leukemia*. The sizes of selected features are much smaller for *Leukemia* data, which are 537 versus 4826 under CV2 and 631 versus 6039 under CV1. Besides, the classification accuracy achieved by PPFSIFW is higher than that achieved by PPFSVW under CV2. It is hard to conclude that which method is better, PPFSIFW is faster and usually selects fewer features, while PPFSVW can achieve higher classification accuracy but with more features and slower.

Table 6.6. Comparison of selected features by PPFSIFW and PPFSVW.

	Separate		**Whole**	
	IFW	*VW*	*IFW*	*VW*
DIA	4	5	4	5
Ionosphere	2	17	8	10
Colon	34	53	157	55
WBC	10	11	4	18
DLBCL	394	121	444	105
Leukemia	537	4826	631	6039

6.4 Conclusions

In this chapter, we proposed a privacy preserving feature selection method via integrating the state-of-the-art methods of filter and wrapper (PPFSIFW). PPFSIFW is based on our previous

work named PAN-SVM, which is a framework for privacy preserving for binary classification using SVM. Therefore, PPFSIFW inherits the privacy preserving property and workable for distributed data mining tasks which have the privacy issues.

PPFSIFW integrates three filter methods, *Fisher score*, *Welch-t-test* and *between versus within class scatter ratio*. Features are firstly evaluated by the three measurements and then selected according to their scores. Features with scores that meet the predetermined thresholds of the three measurements will be voted, selected and kept for next step. Therefore, users can adjust the thresholds and decide about how many features should be chosen, a huge number of features can be removed at this step and sharply reduce the computation cost. The remained features will further be ranked according to the wrapper method of classification accuracy. PPFSIFW uses classification accuracy as the ranking criterion. During the ranking step, features will be picked up from the voted feature subset in order according to their negative affection to classifiers. If eliminating feature *i* can make classifier obtain higher classification accuracy than removing feature *j*; we say feature *i* has higher negative affection than feature *j*, therefore, feature *i* will be eliminated from the remaining list and put into the ranking list before feature *j*. At last, PPFSIFW will return a feature ranking list, with the most important feature on the top, and then the final selected feature subset can be decided from the ranking list, from top to bottom, how many features can make the classifier achieve highest classification accuracy.

PPFSIFW is tested on six benchmark datasets, including three microarray datasets under two testing schema, CV1, and CV2. The experimental results show that PPFSIFW can significantly improve the classification performance by selecting informative features no matter under which testing environment. The classification performance improvement has no obvious pattern for different datasets when tested under CV1 and CV2, in overall, higher improvement can be

obtained under the CV1 testing scheme, but fewer features are selected under the CV2 testing situation when using a separate dataset as testing samples.

The ability of PPFSIFW to enhance the classification performance of a classifier is also compared with other state-of-the-art methods of Fisher-SVM, FSV, RFE-SVM and KP-SVM [74, 78]. PPFSIFW outperforms the other four methods on two datasets that have few features but defeated on the microarray dataset of *Colon*. Slight sacrifice on accuracy is acceptable for PPFSIFW, since it is based on PAN-SVM, which employs the Nystrom technique to approximate the kernel matrix in order to speed up the computation and communication for distributed mining. The most important difference of PPFSIFW from the other methods is that it can preserve individual privacy during the procedure of significantly improving the classifier accuracy. In future work, we will deploy PPFSIFW on cloud and scale it to much bigger and complex datasets.

7 CONCLUSIONS AND FUTURE WORK

The privacy is a major concern for the distributed data mining applications. New sophisticated distributed data mining approaches with privacy issue aware are urgently needed. This work aims to design a framework with a serial of algorithms to keep a good state of equilibrium between privacy, accuracy, and efficiency of data mining algorithms. The new privacy preserving framework named PAN-SVM is proposed in chapter three. It is workable and efficient compared with other state-of-the-art methods based on testing results on 12 benchmark datasets. In chapter four, a privacy preserving multi-class classification method for horizontally distributed data, named PPM2C, is proposed. It follows the One-versus-All schema and employs PAN-SVM as the binary classifier; therefore it can guarantee the privacy. In chapter 5 and 6, two privacy preserving feature selection methods, say PPFSVW and PPFSIFW, are developed. PPFSVW integrates there popular wrapper methods of SVM-RFE, RSVM and SVM-t to achieve higher classification accuracy, and protect individual privacy. PPFSIFW integrates three popular filter methods of *Fisher*, *Welch-t-test* and *between versus within class scatter ratio*, and selects a temporary feature set via voting according to the three measurement scores to eliminate a large number of features and then rank the remaining features according to their contributions to classification accuracy. PPFSIFW works effectively on six benchmarks. It select a small relatively good feature subset more quickly than PPFSVW, but has lower classification accuracy than PPFSVW.

The proposed algorithms PAN-SVM, PPM2C, PPFSVW, and PPFSIFW are workable and efficient. However, there are still a number of limitations. Such as in the proposed model of PAN-SVM, the *secure sum protocol* is used to transmit landmarks while maintaining data privacy; however, this protocol can be compromised if multiple data sources collude. For

example, data source S_{i-1} and S_{i+1} can determine the exact value sent from S_i by comparing the values that they send and receive. Also, the version of PAN-SVM used in testing is implemented in Matlab; memory limitations in Matlab limit it to datasets with millions of samples and features, so do the other three methods. To scale the proposed algorithms to real big data, we will deploy PAN-SVM, PPM2C, PPFSVW and PPFSIFW on the cloud. Besides, the *k-secure sum protocol* with zero probability of data leakage proposed in [88] will be used instead of the original secure sum protocol to make PAN-SVM more secure and efficient, thus no data will be disclosed.

However, not all individuals equally concern about their privacy. For example, people may not regard the disease of flu as private, but HIV as very sensitive information. As a result, we may wish to treat the information in a given dataset very differently for an anonymous purpose. Besides this personalized anonymity, another interesting model [89] allows a person to specify the level of privacy for his or her sensitive information and has the advantage for direct protection of the sensitive values of individuals.

To a nutshell, the needs for distributed data mining with privacy preserving concerns are increasing. Examples may include that collaboration among corporations or agencies without divulging individual trade secrets. Even within a single multi-national corporation, sharing information may be restricted due to different cultures or national laws. This increasing need for privacy preserving distributed data mining will also require flexible solutions that can balance privacy, efficiency, and accuracy, as well as can be tailed for individual privacy needs for different distributed data mining tasks. The current PPDDM algorithms only assume that each party is honest, semi-honest or malicious. In fact, there are many real world scenarios where parties that participate in the secure protocol are “rational”. This assumption may affect the

PPDDM protocols and may make the algorithms more complex. Clearly, further research is needed to explore the effectiveness of the rational behavior in PPDDM.

REFERENCES

[1] R. Gavison, "*Privacy and the limits of law Philosophical Dimensions of Privacy*," Cambridge University Press, 1984.

[2] B. Pinkas, "Cryptographic techniques for privacy-preserving data mining," *SIGKDD Explor. Newsl.,* vol. 4, pp. 12-19, 2002.

[3] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *Sigmod Record,* vol. 29, pp. 439-450, Jun 2000.

[4] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, 2005, pp. 217-228.

[5] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "*l*-diversity: Privacy beyond *k*-anonymity," *ACM Trans. Knowl. Discov. Data,* vol. 1, p. 3, 2007.

[6] A. C.-C. Yao, "How to generate and exchange secrets," in *Foundations of Computer Science, 1986., 27th Annual Symposium on*, 1986, pp. 162-167.

[7] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*: Cambridge University Press, 2004.

[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," presented at the Proceedings of the 17th international conference on Theory and application of cryptographic techniques, Prague, Czech Republic, 1999.

[9] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *SIGKDD Explor. Newsl.,* vol. 4, pp. 28-34, 2002.

[10] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *Ieee Transactions on Knowledge and Data Engineering,* vol. 16, pp. 1026-1037, Sep 2004.

[11] D. Wenliang and M. J. Atallah, "Privacy-preserving cooperative statistical analysis," in *Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual*, 2001, pp. 102-110.

[12] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On Private Scalar Product Computation for Privacy-Preserving Data Mining," in *Information Security and Cryptology – ICISC 2004*. vol. 3506, C.-s. Park and S. Chee, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 104-120.

[13] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Advances in Cryptology-Crypto 2000, Proceedings,* vol. 1880, pp. 36-54, 2000.

[14] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology,* vol. 15, pp. 177-206, Sum 2002.

[15] G. Mathew and Z. Obradovic, "A privacy-preserving framework for distributed clinical decision support," in *Computational Advances in Bio and Medical Sciences (ICCABS), 2011 IEEE 1st International Conference on*, 2011, pp. 129-134.

[16] G. Mathew and Z. Obradovic, "Distributed Privacy Preserving Decision Support System for Predicting Hospitalization Risk in Hospitals with Insufficient Data," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, 2012, pp. 178-183.

[17] S. E. Fienberg, Y. Nardi, and A. B. Slavkovic, "Valid Statistical Analysis for Logistic Regression with Multiple Sources," *Protecting Persons While Protecting the People,* vol. 5661, pp. 82-94, 2009.

[18] W. W. Fang, C. S. Zhou, and B. R. Yang, "Privacy preserving linear regression modeling of distributed databases," *Optimization Letters*, vol. 7, pp. 807-818, Apr 2013.

[19] B. N. Keshavamurthy and D. Toshniwal, "Privacy Preserving Naive Bayes Classification Using Trusted Third Party Computation over Distributed Progressive Databases," *Advances in Computer Science and Information Technology, Pt I*, vol. 131, pp. 24-32, 2011.

[20] J. Vaidya, M. Kantarcioglu, and C. Clifton, "Privacy-preserving Naive Bayes classification," *Vldb Journal*, vol. 17, pp. 879-898, Jul 2008.

[21] X. Yi and Y. C. Zhang, "Privacy-preserving naive Bayes classification on distributed data via semi-trusted mixers," *Information Systems*, vol. 34, pp. 371-380, May 2009.

[22] H. Yu, X. Jiang, and J. Vaidya, "Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data," presented at the Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France, 2006.

[23] J. Que, X. Jiang, and L. Ohno-Machado, "A collaborative framework for Distributed Privacy-Preserving Support Vector Machine learning," *AMIA Annu Symp Proc*, vol. 2012, pp. 1350-9, 2012.

[24] H. Yu, J. Vaidya, and X. Q. Jiang, "Privacy-preserving SVM classification on vertically partitioned data," *Advances in Knowledge Discovery and Data Mining, Proceedings*, vol. 3918, pp. 647-656, 2006.

[25] J. Vaidya, H. J. Yu, and X. Q. Jiang, "Privacy-preserving SVM classification," *Knowledge and Information Systems*, vol. 14, pp. 161-178, Feb 2008.

[26] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," presented at the Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, Chicago, Illinois, USA, 2005.

[27] X. D. Lin, C. Clifton, and M. Zhu, "Privacy-preserving clustering with distributed EM mixture modeling," *Knowledge and Information Systems*, vol. 8, pp. 68-81, Jul 2005.

[28] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45-66, 2002.

[29] Y. M. Lu, Y. H. Gao, Z. B. Cao, J. Cui, Z. N. Dong, Y. P. Tian, *et al.*, "A study of health effects of long-distance ocean voyages on seamen using a data classification approach," *BMC medical informatics and decision making*, vol. 10, Mar 10 2010.

[30] W. Yu, T. Liu, R. Valdez, M. Gwinn, and M. J. Khoury, "Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes," *BMC medical informatics and decision making*, vol. 10, p. 16, 2010.

[31] F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J. P. Vert, "Classification of microarray data using gene networks," *Bmc Bioinformatics*, vol. 8, p. 35, Feb 1 2007.

[32] L. K. Goodwin and J. C. Prather, "Protecting patient privacy in clinical data mining," *J Healthc Inf Manag*, vol. 16, pp. 62-7, Fall 2002.

[33] L. Ohno-Machado, P. S. P. Silveira, and S. Vinterbo, "Protecting patient privacy by quantifiable control of disclosures in disseminated databases," *International Journal of Medical Informatics*, vol. 73, pp. 599-606, Aug 2004.

[34] D. M. Grzybowski, "Patient privacy: the right to know versus the need to access," *Health Manag Technol*, vol. 26, p. 54, Sep 2005.

[35] C. Boyens, R. Krishnan, and R. Padman, "Privacy-preserving data releases for health report generation," *Studies in health technology and informatics*, vol. 107, pp. 1268-72, 2004.
[36] "Standards for privacy of individually identifiable health information. Office of the Assistant Secretary for Planning and Evaluation, DHHS. Final rule," *Fed Regist*, vol. 65, pp. 82462-829, Dec 28 2000.
[37] H. H. S. Office for Civil Rights, "Standards for privacy of individually identifiable health information. Final rule; correction of effective and compliance dates," *Fed Regist*, vol. 66, p. 12434, Feb 26 2001.
[38] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, pp. 273-297, Sep 1995.
[39] P. Drineas and M. W. Mahoney, "On the Nystrom method for approximating a gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153-2175, Dec 2005.
[40] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, pp. 100-108, 1979.
[41] K. Zhang, Lan, L., Wang, Z., and Moerchen, F, "Scaling up Kernel SVM on Limited Resources: a Low-rank Linearization Approach," *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, vol. 22, pp. 1425-1434, 2012.
[42] V. Franc and S. Sonnenburg, "Optimized Cutting Plane Algorithm for Large-Scale Risk Minimization," *Journal of Machine Learning Research*, vol. 10, pp. 2157-2192, Oct 2009.
[43] Y. Lu, P. Phoungphol, and Y. Zhang, "Privacy Aware Non-linear Support Vector Machine for Multi-source Big Data," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 2014, pp. 783-789.
[44] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved Nystrom low rank approximation and error analysis," presented at the Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, 2008.
[45] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling methods for the Nyström method," *J. Mach. Learn. Res.*, vol. 13, pp. 981-1006, 2012.
[46] H. Harbrecht, M. Peters, and R. Schneider, "On the low-rank approximation by the pivoted Cholesky decomposition," *Applied Numerical Mathematics*, vol. 62, pp. 428-440, Apr 2012.
[47] http://www.mathworks.com/products/statistics/.
[48] K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository*. Available: http://archive.ics.uci.edu/ml
[49] (2016). *Gene Expression Omnibus*. Available: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE2990
[50] (2016). *LIBSVM Data: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/*
[51] H. C. Kim, S. Pang, H. M. Je, D. Kim, and S. Y. Bang, "Support vector machine ensemble with bagging," *Pattern Recogniton with Support Vector Machines, Proceedings*, vol. 2388, pp. 397-407, 2002.
[52] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-Based Distributed Support Vector Machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663-1707, May 2010.

[53] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1-122, 2011.

[54] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast support vector machine training and classification on graphics processors," presented at the Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, 2008.

[55] R. Ali and R. Ben, "Random features for large-scale kernel machines," *In Neural Infomration Processing Systems*, 2007.

[56] Y. Lu and Y. Zhang, "Privacy Preserving Multiclass Classification for Horizontally Distributed Data," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA 2017) (Accepted)*, Beijing, China, 2017.

[57] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, *et al.*, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, p. 531, 1999.

[58] Y. S. O. a. M. D. Zexuan Zhu, "Markov Blanket-Embedded Genetic Algorithm for Gene Selection," *Pattern Recognition*, vol. 49, 2007.

[59] X. Zhang, X. Lu, Q. Shi, X.-q. Xu, H.-c. E. Leung, L. N. Harris, *et al.*, "Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data," *Bmc Bioinformatics*, vol. 7, p. 197, 2006.

[60] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 6562-6, May 14 2002.

[61] L. B. Amir Ben-Dor, Nir Friedman, Iftach Nachman, Michèl Schummer, and Zohar Yakhini, *Journal of Computational Biology*, vol. 7, pp. 559-583, July 2007.

[62] C. Furlanello, M. Serafini, S. Merler, and G. Jurman, "Entropy-based gene ranking without selection bias for the predictive classification of microarray data," *Bmc Bioinformatics*, vol. 4, p. 54, 2003.

[63] I. Kholod, M. Kuprianov, and I. Petukhov, "Distributed data mining based on actors for Internet of Things," in *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, 2016, pp. 480-484.

[64] M. Bendechache and M. T. Kechadi, "Distributed clustering algorithm for spatial data mining," in *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on*, 2015, pp. 60-65.

[65] K. Parmar, D. Vaghela, and P. Sharma, "Performance prediction of students using distributed data mining," in *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on*, 2015, pp. 1-5.

[66] I. Guyon, Andr, #233, and Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157-1182, 2003.

[67] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," *Machine Learning*, vol. 46, pp. 389-422, 2002.

[68] R. Díaz-Uriarte and S. Alvarez de Andrés, "Gene selection and classification of microarray data using random forest," *Bmc Bioinformatics*, vol. 7, p. 3, 2006.

[69] A. Sharma, S. Imoto, and S. Miyano, "A Top-r Feature Selection Algorithm for Microarray Gene Expression Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, pp. 754-764, 2012.

[70] C.-H. H. Chen-An Tsai, Ching-Wei Chang, and Chun-Houh Chen, "Recursive Feature Selection with Significant Variables of Support Vectors," *Computational and Mathematical Methods in Medicine,* vol. 2012 2012.

[71] J. Miranda, R. Montoya, and R. Weber, "Linear Penalization Support Vector Machines for Feature Selection," in *Pattern Recognition and Machine Intelligence: First International Conference, PReMI 2005, Kolkata, India, December 20-22, 2005. Proceedings*, S. K. Pal, S. Bandyopadhyay, and S. Biswas, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 188-192.

[72] P. S. Bradley and O. L. Mangasarian, "Feature Selection via Concave Minimization and Support Vector Machines," presented at the Proceedings of the Fifteenth International Conference on Machine Learning, 1998.

[73] Y. Lu and Y. Zhang, "Privacy Preserving Feature Selection on Horizontally Distributed Datasets," in *2017 5th International Conference on Bioinformatics and Computational Biology (ICBCB 2017) (Accepted)*, Hong Kong, China, 2017.

[74] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning,* vol. 46, 2002.

[75] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, *et al.*, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 96, pp. 6745-6750, 1999.

[76] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, *et al.*, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature,* vol. 403, pp. 503-511, 02/03/print 2000.

[77] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1-27, 2011.

[78] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized support vector machines," *Information Sciences,* vol. 181, pp. 115-128, 2011.

[79] J. W. Isabelle Guyon, Stephen Barnhill, Vladimir Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," *Machine Learning* vol. 46, pp. 389-422, 2002.

[80] J. Xuan, Y. Wang, Y. Dong, Y. Feng, B. Wang, J. Khan, *et al.*, "Gene selection for multiclass prediction by weighted Fisher criterion," *EURASIP J Bioinform Syst Biol,* p. 64628, 2007.

[81] R. Diaz-Uriarte and S. A. de Andres, "Gene selection and classification of microarray data using random forest," *Bmc Bioinformatics,* vol. 7, Jan 6 2006.

[82] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *In Advances in Neural Information Processing Systems (NIPS) 13, 2000*, L. e. e. n. TK, D. i. Tg, and T. r. V, Eds., ed, 2001.

[83] Y. Lu and Y. Zhang, "Privacy Preserving Feature Selection via Integrating Filter and Wrapper Methods for Horizontally Distributed Data," in *2017 International Conference on Cryptography, Security and Privacy (ICCSP 2017) (Accepted)*, Wuhan, China, 2017.

[84] R. A. FISHER, "The use of multiple measurements in taxonomic problems," *Ann. Eugen. 7*, vol. 7, pp. 179-188, 1936.

[85] T. X. Jieping Ye, "Computational and Theoretical Analysis of Null Space and Orthogonal Linear Discriminant Analysis " *Journal of Machine Learning Research,* vol. 7, pp. 1183-1204, 2006.

[86] Y. H. Y. Sandrine Dudoit , Matthew J. Callow , Terence P. Speed, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," *STATISTICA SINICA,* vol. 12, pp. 111-139, 2002.

[87] (2013). *LIBSVM Data: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/*

[88] B. K. Rashid Sheikh, Durgesh Kumar Mishra "A Distributed k-Secure Sum Protocol for Secure Multi-Party Computations," *JOURNAL OF COMPUTING,* vol. 2, 2010.

[89] X. Xiao and Y. Tao, "Personalized privacy preservation," presented at the Proceedings of the 2006 ACM SIGMOD international conference on Management of data, Chicago, IL, USA, 2006.