THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

HEALTH CARE PREDICTIVE ANALYTICS USING ARTIFICIAL INTELLIGENCE TECHNIQUES

GUANJIN WANG

PhD

The Hong Kong Polytechnic University

This programe is jointly offered by The Hong Kong Polytechnic University and University of Technology Sydney

2018

The Hong Kong Polytechnic University
School of Nursing
University of Technology Sydney
Faculty of Engineering and Information Technology

Health Care Predictive Analytics Using Artificial Intelligence Techniques

Guanjin Wang

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

February, 2018

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signed _____

Guanjin Wang

Abstract

In recent years, advances in Artificial Intelligence (AI) are opening the door for intelligent health care data prediction and decision making. Machine learning, as an increasingly popular approach to AI, has been widely used to learn directly from data, adapt independently, and produce predictive outcomes, which support doctors when encountering complex health care predictive analytics. However, traditional machine learning methods are not always perfectly working in the health field, intrinsically due to little consideration for characteristic problems within health care data. For example, the small sample size problem is common due to complex data collection procedures and privacy concerns. Missing data is also widely encountered since most data are collected as a second-product of patient-care activities instead of following systematic research protocols. The class imbalance is another inevitable problem in the medical data as the normal class usually predominates over the disease class. To solve aforementioned issues in health care predictive analytics, this study stands on the principles of machine learning and transfer learning to develop five advanced prediction models.

The first model is an output-based transfer least squares support vector machines (LS-SVMs) model which can leverage knowledge learned from the existing prediction model to facilitate the learning process on the target domain with insufficient data. This model overcomes the small sample

size problem and improves the health care data prediction by learning knowledge from the other domain.

The second model is a novel additive LS-SVMs model which can directly make predictions on missing data by simultaneously evaluating the influences on the classification error made by missing features. Moreover, this model can generate explanatory information for health professionals to improve the future data collection process.

The third model is a transfer-based additive LS-SVMs model which can deal with missing data from a transfer learning perspective. It leverages the model knowledge learned from the complete portion of the dataset to help the learning process on the whole dataset with missing data. The proposed model can provide supplementary information for health professionals to improve the data quality via data cleaning.

The forth model is a deep transfer additive LS-SVMs model called DTA-LS-SVMs and its imbalanced version called iDTA-LS-SVMs to enhance the prediction performance on the balanced and imblanced datasets. Enlightened by the deep architecture and transfer learning, the model stacks multiple additive LS-SVMs based modules layer-by-layer and embeds model transfer between adjacent modules to guarantee their consistency.

The fifth model is a deep cross-output transfer LS-SVMs model called DCOT-LS-SVMs and its imbalanced version called IDCOT-LS-SVMs to improve the prediction performance on the balanced and imbalanced datasets. The cross-output transfer is used to transfer the knowledge of outcomes from the previous module to the adjacent higher layer to achieve a better learning. Moreover, modules' parameters can be randomly

assigned in the proposed model which significantly simplifies the learning process.

The proposed models are verified using the public UCI datasets. Moreover, case studies are conducted to validate and integrate the proposed models with real world applications, including bladder cancer prognosis, prostate cancer diagnosis, and predictions of elderly quality of life (QOL). The experimental results have demonstrated that these models can enhance the prediction performances while taking the characteristic problems within health data into account, thus exhibiting potential to be widely used in the real world applications in future.

Publications during PhD study

The international refereed journal and conference papers relevant with my PhD study which have been submitted, accepted and published are list below:

Refereed International Journal Publications:

(1) **Wang, G.**, Lam, K.M., Deng, Z. and Choi, K.S., 2015. "Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques," *Computers in Biology and Medicine*, 63, pp.124-132.

(2) **Wang, G.**, Deng, Z. and Choi, K.S., 2016. "Tackling missing data in community health studies using additive LS-SVM classifier," *IEEE Journal of Biomedical and Health Informatics*, 22(2), pp. 579-587.

(3) **Wang, G.**, Choi, K.S. and Deng, Z., 2016. "Noise-benefit FRSDE for speedup of density estimation on large data," *Journal of Intelligent & Fuzzy Systems*, 30(1), pp.443-450.

(4) **Wang, G.**, Lu, J., Choi, K.S. and Zhang, G., "A transfer-based additive LS-SVM classifier for handling missing data," *IEEE Transactions on Cybernetics*, 2016 (Under second review)

(5) **Wang, G.**, Deng, Z. and Choi, K.S., 2017. "Detection of epilepsy with Electroencephalogram using rule-based classifiers," *Neurocomputing*, 228, pp.283-290.

(6) **Wang, G.**, Zhang, G., Choi, K.S. and Lu, J., "Deep additive Least Squares Support Vector Machines for classification with model transfer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017 (Accepted and on-line available)

(7) **Wang, G.**, Choi, K.S., Teoh, Y.C. and Lu, J., "Deep cross-output knowledge transfer using stacked-structure Least Squares Support Vector Machines," *IEEE Transactions on Cybernetics*, 2017 (Under review)

(8) **Wang, G.**, Zhang, G., Choi, K.S., Lam, K.M. and Lu, J., "Output-based transfer learning with Least Squares Support Vector Machine and its application in bladder cancer prediction," *Artificial Intelligence in Medicine*, 2017 (Under second review)

Refereed International Conference Publications:

(1) **Wang, G.**, Deng, Z. and Choi, K.S., "Detection of epileptic seizures in EEG signals with rule-based interpretation by random forest approach," in *Proceedings of 2015 International Conference on Intelligent Computing*, August. 2015, Fuzhou, China (pp. 738-744).

(2) **Wang, G.**, Zhang, G., Choi, K.S., Lam, K.M. and Lu, J., "An output-based knowledge transfer approach and its application in bladder cancer prediction," in *Proceedings of 2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, May. 2017, Anchorage, USA (pp. 356-363).

Acknowledgements

PhD study has been a challenging and memorial journey in the past three years. I would like to extend my warmest gratitude to the people who inspired and helped me along this journey.

First, I would like to express my earnest thanks to my chief supervisor in the Hong Kong Polytechnic University, Professor Choi Kup Sze, and my chief supervisor in the University of Technology Sydney, Distinguished Professor Jie Lu for their continuous support, precious guidance and enormous patience throughout my study. Without their excellent supervision and valuable suggestions, this Joint-PhD study between two universities could not have been finished. Thank you for all your detailed comments and suggestions on my research, manuscripts and presentations. Your strict academic attitude and hard working style have deeply influenced me and will alway inspire and motivate me in my future work and life. I also would like to address my sincere thanks to Professor Guangquan Zhang for his academic suggestions and advices, Dr Chiang Chung Lim Vico for his health care knowledge sharing and advices and Dr Lam Kin Man for his medical knowledge sharing and data support.

I am honored to have met all the great researchers and staffs in the Centre for Smart Health in the Hong Kong Polytechnic University and Centre for Artificial Intelligence in the University of Technology Sydney. I appreciate all their valuable and critical comments during my presentations, and the discussions with them were enlightening.

I am grateful to the School of Nursing in the Hong Kong Polytechnic University and the School of Software in the Faculty of Engineering and Information Technology in the University of Technology Sydney. This study was supported by the YC Yu Scholarship for the Centre for Smart Health, UTS Doctoral Scholarship, the Research Grants Council of the Hong Kong SAR and the Australian Research Council (ARC) discovery project.

Lastly, I would like to express my most grateful appreciation to my family for their unconditional love, encouragement and support. This journey would not have been possible without their help. I am especially grateful to my parents. Thank you for being the first teacher in my life and guiding me to be a better person. I always know you believe in me more than I do in myself.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the introduction to this study. Section 1.1 introduces the background of health care data analytics. Section 1.2 to Section 1.5 explain the challenges, contributions, significance and structure of this thesis.

1.1 Health Care Data Analytics

In the big data era, data have already become one of the most significant components in the health field. In a recent report on big data Manyika et al. [2011], the overall potential of health care data is estimated to reach around $300 billion in the United States. Thanks to the fast-growing sensing and data acquisition technologies, hospitals and institutions can nowadays easily collect and store large amounts of health care data in various forms, including sensor data, electronic health records (EHRs), medical images and clinical notes. To gain a better understanding and find underlying values from data, advanced data analytics techniques are required to dig deeply into the raw data and transform them into the meaningful knowledge. The successful implementations can gain new insights, leading to advances in patient care practices and health care operations, such as the prompt diagnosis and prognosis of cancers Afzal et al. [2013]; FitzHenry et al. [2013]; Gottlieb et al. [2013]; Makam et al. [2013]; Sylvester et al. [2006]; Van De Vijver et al. [2002], prediction of risk of readmission

Amarasingham et al. [2010]; Donzé et al. [2013]; Gildersleeve and Cooper [2013] and personalized medicine based on individual genomic profiles Chin et al. [2011a]; Ginsburg and McCarthy [2001]; Hamburg and Collins [2010].

Health care data can be stored in different forms. The structured electronic health records (EHRs) and biomedical images are collected in the clinical environment. EHRs describe patients' medical history, including demographics, medications, vital signs, laboratory test results, radiology reports, doctors' notes and billing data. EHRs store and manage personal health information in digital format which provide convenience to share information instantly across different health care organizations and institutions, motivate patients' participation and support better patient diagnostic and prognostic outcomes. Medical images also play an important role in the medical data analytics. Nowadays, there are various diagnostic techniques, such as magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET) and ultrasound (U/S), to look inside the patient body and gain a better understanding of the cause of an illness without making surgical cuts. However, analyzing such complex medical scans is very time consuming and expensive. Researchers and doctors recently have begun to benefit from the rise of machine learning, especially deep learning Shen et al. [2017] for the more effective computational medical image analysis. Sensor data is a type of health care data for real-time and retrospective analysis. They can be collected from electrocardiogram (ECG) and electroencephalogram (EEG) sensors on different parts of the human body. A typical application of sensor data is a real-time remote monitoring of the patient with the special medical condition in the intensive care units (ICUs) Vespa et al. [1999]. The analytical tools for the sensor data must make excellent observations on the big volume of data and be sensitive to any situational changes. Genomic data analysis also has been given lots of attention recently. Finding the relationships between different genetic markers, mutations and cancer conditions can considerably help the development of

future gene therapies. Another research trend on this is to transform the genomic knowledge into the personalized medicine practice. Many intelligent algorithms and bioinformatics tools are developed to deal with genomic data to serve the purpose of identification of disease biomarkers, therapeutic purposes and estimation of clinical outcomes Chin et al. [2011b]; Garnett et al. [2012]; Kim et al. [2012]; Samuels et al. [2004]; Welsh et al. [2001]. In addition, health care data can be stored in an unstructured form, such as clinical notes. Although these notes contain wealthy resources, it is very difficult to automatically extract information from textual clinical documents without human intervention Jagannathan et al. [2009]. In recent years, the computer-based methods such as Natural Language Processing (NLP) and machine learning techniques have been widely utilized to identify and extract information from the unstructured text Zheng et al. [2014].

1.2 Challenges

The advent of data era introduces big opportunities and challenges in the health care field. To gain a better understanding and extract the underlying knowledge from the complex health care data, traditional statistical and Artificial Intelligence (AI) prediction models are commonly employed for predictive analytics. Successful applications such as individualized diagnosis and prognosis, hospital readmission prediction and personalized medicine can lead to improvements in medical practices and health care experiences.

However, health care data has its uniqueness which deserves special attention when constructing prediction models. Delving into this, we find three common issues with the health care data. First, health care datasets are often characterized by limited samples due to the complexity and high-cost patient data collection procedures. However, most prediction models with superior performances require sufficient training data. When handling small datasets, their prediction performances

can be deteriorated. Second, missing data problems are also inevitable. They can be caused inadvertently, or intentionally due to privacy concerns. When health care data are collected as a byproduct of patient-care activities from the real world, it lacks the integrity required by research protocols. Thus, it is very common in the health care dataset that a patient record has missing values for certain features, or a feature loses values for several patient records. Inappropriate handling of missing data can easily cause bias or lead to loss of information which directly affect the prediction performance. Third, health care data usually have class imbalance problems, since there is a much larger number of samples in the normal group than those in the diseased group during data collection. The constructed prediction models consequently get influenced by the predominant classes and ignore the minor ones.

1.3 Research Contributions

To address the aforementioned challenges in the health data, this thesis proposes five novel prediction models using advanced AI techniques, including an output-based transfer LS-SVMs model, a novel additive LS-SVMs model, a transfer-based additive LS-SVMs model, a deep transfer additive LS-SVMs model and a deep cross-output transfer LS-SVMs model. The main contributions of this research are summarized as follows:

An output-based transfer LS-SVMs model is proposed to deal with classification with small data from a transfer learning perspective. It can efficiently leverage the output knowledge from the existing prediction model or on-line tool to facilitate the learning process on the current interest of domain with small data. This model does not require the data and modeling details of the existing model, which is applicable to be used in the real world health care scenarios (Chapter 3).

A novel additive LS-SVMs model is proposed to deal with missing data. It can directly perform classification simultaneously while taking the influences on the

classification error caused by missing features into consideration. The influence level of the missing feature can be quickly and autonomously determined using a fast leave-one-out cross validation strategy. Meanwhile, it can provide supplementary information to guide health professionals to improve the future data collection process in practice (Chapter 4).

This thesis also proposes a transfer-based additive LS-SVMs model from a transfer learning perspective to deal with missing data. It can leverage the model knowledge learned from the complete portion of the dataset to help the learning process on the whole dataset with missing data. Like the previous model, the influences on the classification error caused by incomplete samples can be determined using a fast leave-one-out cross validation strategy, which also provides distinct information for data cleaning to guarantee data quality (Chapter 5).

A deep transfer additive LS-SVMs model called DTA-LS-SVMs and its imbalanced version called iDTA-LS-SVMs is proposed to improve the classification performances on balanced and imbalanced datasets. We enhance the model using data argumentation via a deep stacked architecture to make the original data space more separable. Model transfer is employed between adjacent modules to guarantee their consistency and thus classification capability of the higher module is expected to be further improved (Chapter 6).

This thesis also proposes a deep cross-output transfer LS-SVMs model (DOCT-LS-SVMs) and its imbalanced version (IDCOT-LS-SVMs) to improve the classification performances on balanced and imbalanced datasets. It combines multiple LS-SVMs based modules layer-by-layer embedded with output knowledge transfer between adjacent modules. Moreover, model parameters, such as trade-off parameter and kernel width can be randomly selected in each module which greatly simplifies the learning process (Chapter 7).

The experiments and empirical studies in this thesis can be divided into two

categories: public UCI datasets and real-world health care data resources including the bladder cancer dataset for prognosis, the community health care dataset for predicting elderly QOL and the prostate cancer dataset for diagnosis. The empirical studies demonstrate the feasibility and effectiveness of the proposed prediction models using advanced AI techniques in the real world health care applications.

1.4 Research Significance

This thesis is expected to make contributions to the improvement of health care predictive analytics using advanced AI techniques. Particularly, it is a significant endeavor to customize AI techniques to handle the characteristic problems encountered in analyzing health data, such as small sample size, missing data and class imbalances. It will be an exemplar demonstrating the benefits of using AI in health care, thereby promoting practical and advanced applications in clinical practice. Moreover, this study is expected to make a practical and theoretical contribution in the areas of health informatics and machine learning.

1.5 Thesis Structure

This thesis consists nine chapters. Chapter 1 presents the research background, research challenges, contributions and significance related to this thesis. Chapter 2 presents the literature relevant to this research, including a review of traditional health care data prediction models, transfer learning, deep architectures and common issues in health care data and their solutions. Chapter 3 proposes an output-based transfer LS-SVMs model to solve the small sample size problem. Chapter 4 proposes a novel additive LS-SVMs model to handle classification with missing data. Chapter 5 develops a transfer-based additive LS-SVMs model to perform classification with missing data from a transfer learning perspective. Chapter 6 proposes a deep transfer

additive LS-SVMs model and its imbalanced version. Chapter 7 proposes a deep cross-output transfer LS-SVMs model and its imbalanced version. Chapter 8 presents the conclusions and future work directions. Lastly, Chapter lists the publications during PhD study. The thesis structure is clearly shown in Fig. 1.1.

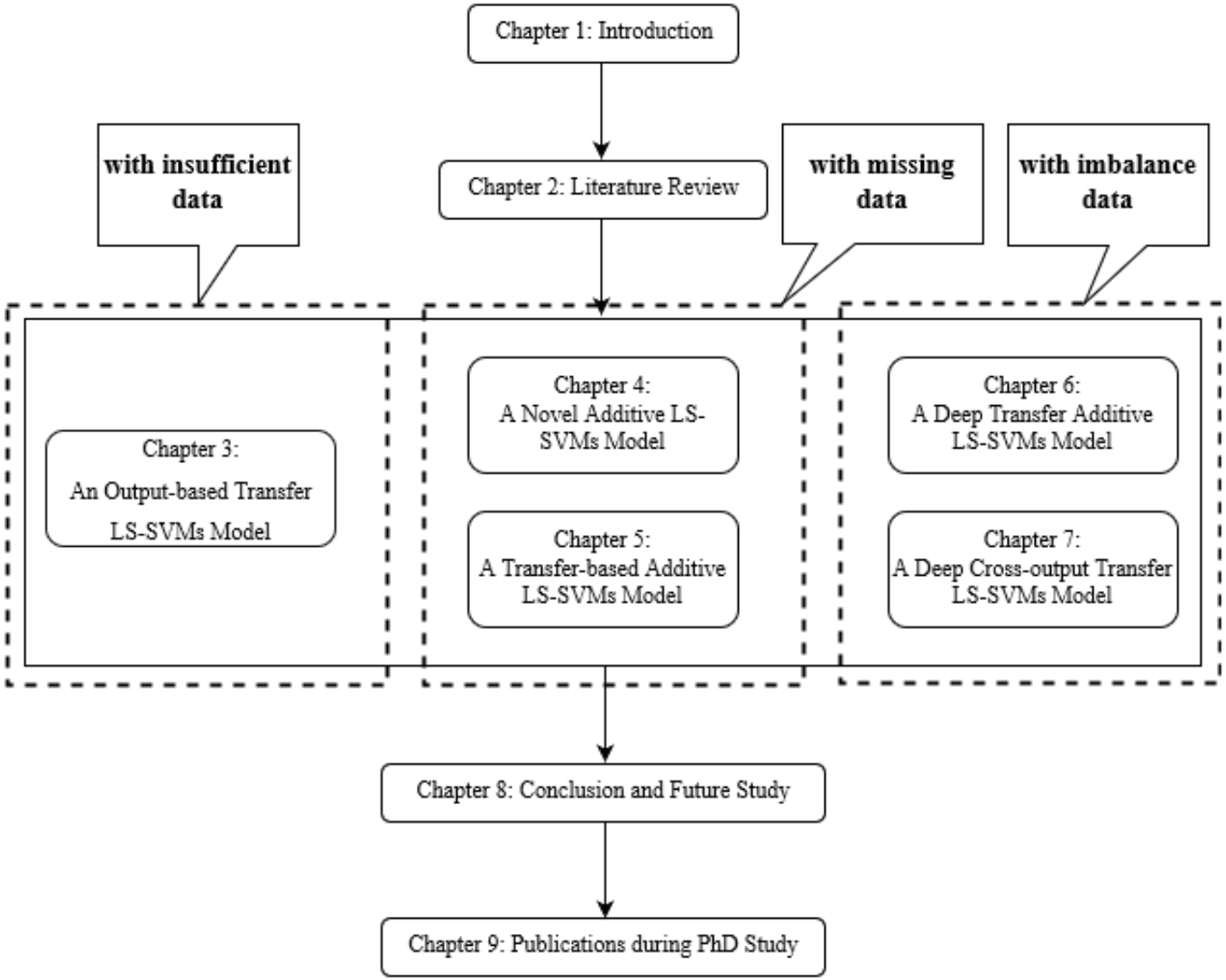


Figure 1.1: THESIS STRUCTURE

Chapter 2

Literature Review

This chapter gives the background of the study. Sections 2.1 and 2.2 review the traditional statistical and AI models for health care prediction, perspectively. Transfer learning, missing data problem, class imbalance problem and deep architecture are introduced from Section 2.3 to Section 2.6, respectively.

2.1 Health Care Data Prediction: Classical Statistical Models

Various statistical models have been used to classify the patients' status according to their health status. In this section, we review three basic statistical methods which have been widely investigated in health care.

- **Linear Regression** Regression analysis is a process of fitting prediction models to data by investigating the relationship between independent variables (predictors) and the dependent variable (outcome). Linear regression Hastie et al. [2009] is a type of regression techniques which assumes that there is a linearity between independent and continuous dependent variables in the case of corresponding estimated regression parameters. The independent variable(s)

can be continuous or discrete. *Least squares* method is the most frequently used coefficient estimation method by minimizing the sum of the squared deviations between the data points and the curve.

Although linear regression prediction models are very simple to understand, they are based on the assumption that independent and dependent variables are linearly associated, which is, in fact, very difficult to satisfy in many health care applications. Moreover, linear regression is very sensitive to outliers, which may strongly influence the fitting model and the predicted values. Multiple regression also suffers from multicollinearity which results in unstable coefficient estimates. These disadvantages limit the prediction performances of linear regression models in complex health care predictive analytics compared with computational-intelligence oriented AI techniques Catto et al. [2003]; Sousa et al. [2007].

- **Logistic Regression** Logistic regression is widely used for classification in health care where the dependent variable is binary (e.g., 0/1, alive/dead, normal/diseased) or ordinal (e.g, 'poor', 'neutral' and 'good') Dreiseitl and Ohno-Machado [2002]; Pregibon [1981]. The goal of logistic regression is to estimate probabilities using a logistic function. The model coefficients are usually approximated using maximum likelihood estimation. Logistic regression can handle various types of relationship between dependent and independent variables and have the interpretation capability of model parameters. However, it carries a higher risk of over-fitting issues which may influence the prediction performance on the unseen data Dreiseitl and Ohno-Machado [2002]. Moreover, it requires big sample size such that the maximum likelihood estimates can

be guaranteed to be powerful. In literature, logistic regression is frequently compared with computational-intelligence oriented AI techniques in terms of generalization performances Hanai et al. [2003]; Jefferson et al. [1997]; Marchevsky et al. [1998, 1999]; Singson et al. [1999].

- **Survival Models** Survival analysis is a set of methods to model time to event data where the outcome is the time until an event occurs Klein and Zhang [2005]; Miller Jr [2011]. This event can be death, discharge from the hospitalization, cancer recurrence or any other incident of interest happening during the observation. The starting point can be the diagnosis of cancer, hospitalization admission and the first time to have a specific treatment. The survival time can be measured in hours, days, years, etc. To model the survival time associated with variables such as patient histological, pathological and clinical characteristics, survival models are particularly effective in handling censored observations. The observations with incomplete information of their survival time is called censored. For example, the participant who drops out from the study before the end of the study is right censored. The participant who does not experience the event of interest in the whole study is censored. Unlike ordinary regression analysis, survival models can combine information from censored and uncensored observations to estimate model parameters. The nonparametric Kaplan-Meier method and the semiparametric Cox proportional hazards regression model are the most common approaches for survival analysis Ohno-Machado [2001].

2.2 Health Care Data Prediction: Artificial Intelligence Models

Although statistical methods are widely used for analyzing health care data, there are a number of pitfalls which may influence their performances and feasibilities in the real-world scenarios. For example, most statistical methods are criticized regarding their explicit assumptions, which are highly likely to be violated in clinical practice Egner [2010]. On the other hand, the rapid growth of AI has been giving more and more attention in the health field and motivate researchers to use advanced intelligent techniques as alternative methods for predicting health care data.

The concept of Artificial Intelligence (AI) Russell and Norvig [2003] was proposed in 1956 at Dartmouth College in Hanover. With the increasing computational power and advances in big data over the past decades, AI has been growing very fast and become one of the most significant research topics in computer science in the 21st century. It imitates human being's perception, learning, and reasoning to solve various complex problems.

Machine learning as an important approach to AI can produce reliable prediction outcomes by discovering hidden patterns from examples and experiences, which has been regarded as a promising alternative to statistical models. Machine learning methods have been extensively applied in different health care applications, such as personalized and predictive medicine Bassi et al. [2007], cancer diagnosis and detection Millan-Rodriguez et al. [2000], recommendations for treatments and therapies Rubin and Reisner [2009]. A detailed review of the most well-known machine learning methods are introduced below.

2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are originally inspired by the structure of the biological neural network in neuroscience, in which billions of neurons are connected with each other in a human brain through axons. Fig. 2.1 demonstrates the structure of a biological neuron. The dendrites receive signals from the external environment. If the sum of signals in this neuron exceeds a threshold, the neuron is activated and the signal is sent through the axon to other neighbors, otherwise, the delivery stops.

In 1943, McCulloch and Pitts [1943] proposed the first artificial neuron called McCulloch-Pitts (MCP) model, which performs like a linear threshold gate. In 1957, the simplest ANNs - perceptron was invented by Rosenblatt [1958] at the Cornell Aeronautical Laboratory. It consists of two layers of nodes to learn a binary classifier, in which the nodes on the second layer functionally process the inputs. In order to solve more complex non-linear problems, the traditional ANNs usually contains several hidden layers to adequately model the underlying behavior of the input data. The common type of ANNs include back-propagation feedforward neural networks (BPNN) and radial basis function (RBF) networks.

ANNs are one of the most extensively used machine learning methods for health care prediction Dreiseitl and Ohno-Machado [2002]; Solanki et al. [2016]. For example, Bassi et al. [2007] constructed ANNs to predict the 5-year overall mortality of bladder cancer patients undergoing radical cystectomy. Its prognostic performance was comparable with that using logistic regression analysis. Er et al. [2010] evaluated several ANNs on the diagnosis of chest disease and drew a conclusion that ANNs can be successfully used to assist clinicians with the detection of the disease. Ecke et al. [2012] evaluated an ANNs-based program 'ProstateClass' for prostate cancer detection. The experimental results showed that it has potential to be used in clinical practice to increase the cancer detection accuracy and reduce unnecessary biopsies.

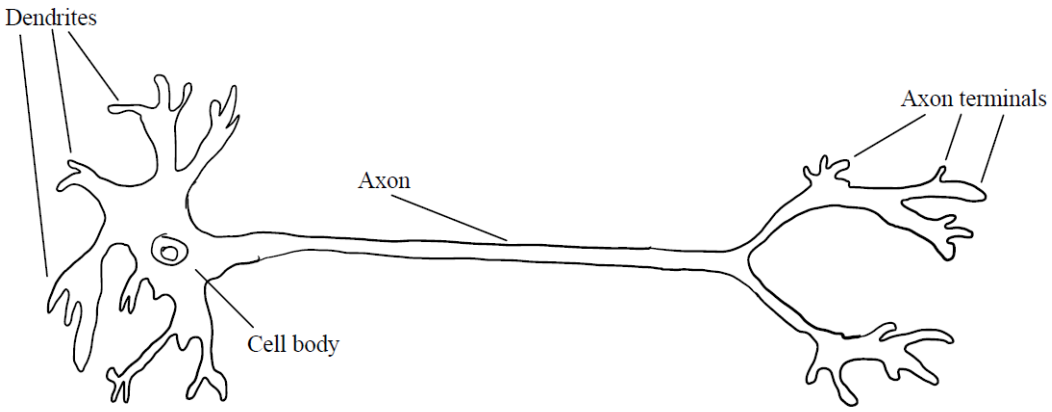


Figure 2.1: THE STRUCTURE OF A BIOLOGICAL NEURON

2.2.2 Support Vector Machines

Support Vector Machines (SVMs) was proposed by Cortes and Vapnik [1995]. It can project the original data to a higher dimensional feature space where an optimal hyperplane can be found to maximize the margin between different classes. As demonstrated in Fig. 2.2, several hyperplanes can be found between two classes. Of them, only the solid line is the optimal hyperplane which keeps the largest distances between the closest data points of two classes to this hyperplane. These data points which help to identify the boundary are called support vectors. Once the boundary is decided, the other data except support vectors become redundant. Therefore, the performance of SVMs does not reply on the sample size of the training dataset. Moreover, kernel trick can be used to implicitly transform the input data to a higher dimensional space by simply computing the proper inner product of data in that transformed space, which significantly reduces the computational burden. Hence, SVMs is known as the famous kernel-based machine learning method for pattern recognition. Other advantages of SVMs include no overfitting and local minima issues.

SVMs have been widely applied in health care applications especially with small

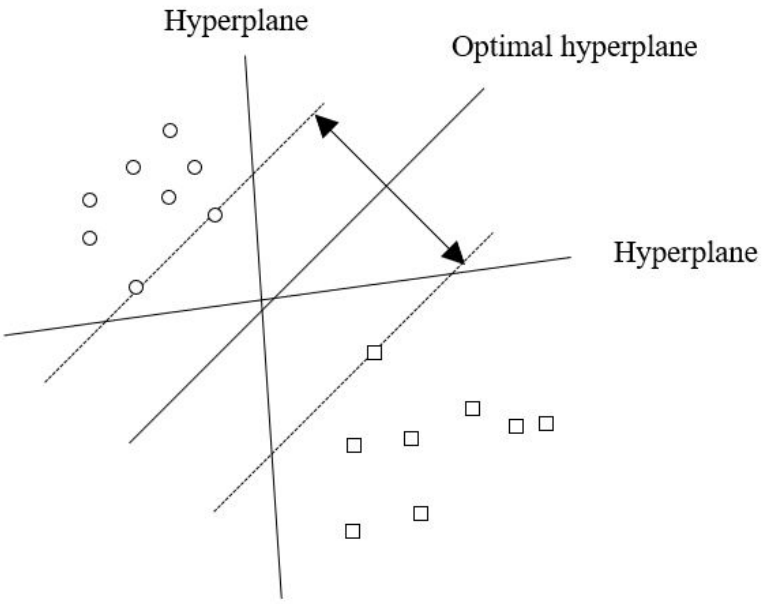


Figure 2.2: OPTIMAL HYPERPLANE SEPARATES TWO CLASSES WITH THE MAXIMUM MARGIN.

datasets due to their excellent generalization performances. Sanchez-Carbayo et al. [2006] used SVMs for predicting the outcome of advanced bladder cancer associated with genes which help patients gain the maximum benefit from more aggressive therapeutic intervention. Huang et al. [2008] built an accurate diagnostic model of breast cancer and fibroadenoma using SVMs for young women in Taiwan, showing that SVMs had a better prediction ability than that using Linear Discriminate Analysis (LDA). In Çınar et al. [2009], SVMs were used to construct the prediction model for the diagnosis of early prostate cancer, and achieved a better outcome compared with ANNs.

2.2.3 Least Square Support Vector Machines

Least Squares Support Vector Machines (LS-SVMs) is a variant of the standard SVMs, which was proposed by Suykens et al. [2002] in 1999. Its training process is much

more simplified than SVMs, which is to solve a system of linear equations instead of a quadratic programming (QP) problem in SVMs. Its objective function is modified by converting the inequality constraints in SVMs to the equality one and changing the empirical risk error from 1-norm to 2-norm. Several empirical studies Van Gestel et al. [2004] Zhang and Peng [2004] have shown that LS-SVMs are comparable to SVMs in terms of the generalization performance. Moreover, the analytical solution of LS-SVMs can help formulate the fast leave-one-out cross validation strategy for model selection, which lead to the significant improvement on the learning speed Cawley [2006].

Selvaraj et al. [2007] proposed an advanced classification technique based on LS-SVMs for brain image slices classification. Results showed the proposed approach outperformed SVMs, RBF classifier, multilayer Perceptron classifier and k-nearest neighbors (k-NN) classifier. Polat et al. [2008] applied the LS-SVMs on the ECG dataset to detect patients with arrhythmia. 100% classification accuracies can be achieved on the testing datasets, showing that LS-SVMs are more promising than previously reported classifiers in the computer-aided diagnosis system of ECG data. In Li et al. [2009], LS-SVMs were also used to classify normal people with eye open and epileptic patients during epileptic seizure activity using EEG signals and achieved 80.05% accuracy, proving to be a potential technique to classify EEG signals. Polat and Güneş [2007] used LS-SVMs for the diagnosis of breast cancer and evaluated the performance on the Wisconsin breast cancer dataset (WBCD) using different metrics. The results indicated that LS-SVMs had the superior advantages compared with the other previously reported machine learning techniques.

2.2.4 Naive Bayes Classifiers

Naive Bayes classifiers are a class of probability models in machine learning using Bayes' theorem with strong independence assumptions between variables. They

attempt to maximize the posterior probability to determine the class of the unseen data. The idea of naive Bayes has been studied since the 1950s, and been extensively introduced to solve problems such as automatic medical diagnosis Rish [2001]. Rajkumar and Reena [2010] constructed the Naive Bayes classifier to analyze a heart disease dataset for diagnosis and compared the classification performance with decision list algorithm and k-NN algorithm. The results showed that the naive Bayes classifier achieved the best performances. In Parthiban et al. [2011], the naive Bayes method was applied to diagnose heart disease for diabetic patients and performed well compared with other similar methods in literature.

2.2.5 Extreme Learning Machines

Extreme learning machines (ELMs) are simple and effective feed-forward neural networks with one or more hidden layers, where the parameters of hidden nodes can be randomly assigned or inherited from their ancestors with no change instead of being tuned. Therefore, the learning process of the output weights in ELMs is essentially equivalent to learning a linear model. Due to these characteristics, ELMs can achieve the good generalization performance at an extremely low running time, compared with the traditional BPNNs. Huang et al. [2006a] used ELMs to detect diabetes and compared the results with those using other popular machine learning methods. According to the experimental results, ELMs outperformed SVMs Rätsch et al. [1998], SAOCIF Romero and Alquezar [2002], Cascade-Correlation algorithm Romero and Alquezar [2002], bagging and boosting methodsFreund et al. [1996], C4.5 Freund et al. [1996], and Radial basis function network Wilson and Martinez [1996] reported in the previous studies, showing a promising potential in health care data prediction.

2.2.6 k-nearest Neighbors Algorithms

Unlike machine learning methods introduced above, the k-nearest neighbors algorithm (k-NN) Cover and Hart [1967] is a non-parametric machine learning method, which can directly classify the unseen data by voting from the selected k neighbors. The simplest example is that if $k = 1$, the new sample is assigned to the same class of the nearest neighbor. Fig. 2.3 illustrates an example of k-NN classification. The triangle represents a new incoming sample for labeling either as '+' class or to '-' class. If k is set to 1, the testing sample is assigned to '+' class since there are 1 '+' class and 0 '-' class within the solid black line circle. If k is set to 3, the testing sample is assigned to '-' class since there are 2 '-' classes and 1 '+' class within the dashed line circle. If k is set to 5, the testing sample is assigned to '+' class since there are 3 '+' classes and 2 '-' class within the outer circle. We can observe that the value of k needs to be set very carefully to reduce the effect of outliers and noise. Euclidean distance and Hamming distance are the common distance metrics for continuous and discrete variables respectively.

k-NN has been used to support health care prediction mainly due to its simple and easy implementation. Shouman et al. [2012] adopted k-NN to diagnose heart disease and found that it achieved higher accuracy compared with ANNs with easy implementation. Li et al. [2014] utilized principal component analysis (PCA) to image feature extraction followed by applying k-NN for diagnosing sperm health. This combination achieved the higher diagnostic accuracy compared with that using the combination of PCA and the propagation neural networks. In some health studies, k-NN was also used as the comparative method for performance evaluation Tama [2010]; Vijayan and Ravikumar [2014].

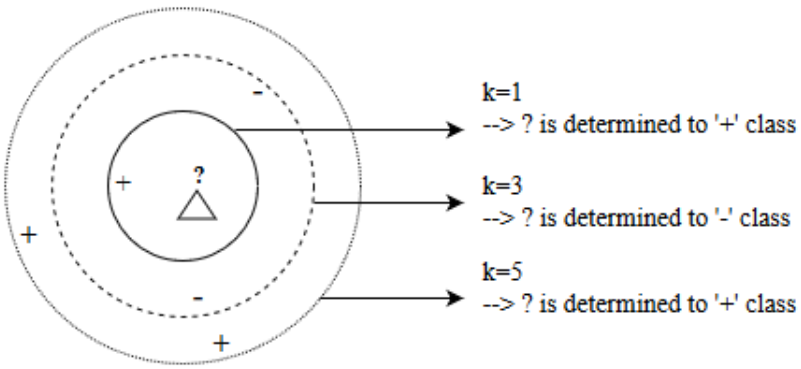


Figure 2.3: EXAMPLE OF k-NN CLASSIFICATION

2.3 Transfer Learning

Traditional machine learning methods for predictive analytics have many successful achievements in health care. However, most of them are developed based on an assumption that the training and testing data must have the same feature space under the same distribution. If either condition is not satisfied, the prediction model must be constructed from beginning using the newly collected training data. This process is very expensive and impractical in many real-world scenarios due to the complex data collection and patient privacy concerns. On the other hand, most machine learning methods require sufficient data for training. Small training data may substantially deteriorate the prediction performance of the constructed model.

These challenges make researchers wonder if there is a way that they can leverage the knowledge from a related but different domain (source domain) with sufficient data to help the learning process in the current interest of domain (target domain) with few data. For example, we want to construct a prediction model for predicting the 5-year mortality of bladder cancer in the Chinese population. However, there are only a few Chinese patient electronic records available for training (target domain). We have another big amount of bladder cancer data from American participants (source domain). Although the target populations are different, these two domains are still to

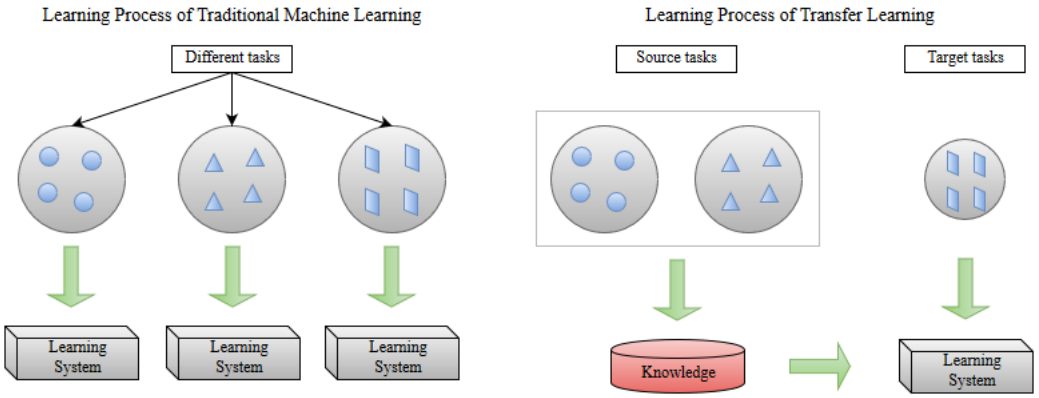


Figure 2.4: DIFFERENT LEARNING PROCESSES OF TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING

some extent similar due to the same cancer type. Therefore, if there is a way to find a bridge to share the learned knowledge between two domains, it can benefit many real-world small data scenarios.

Transfer learning is proposed to achieve this goal. It aims to leverage the knowledge from a source domain onto a target one to improve the performance of learning. It is inspired by how human being transfer knowledge between similar contexts in Psychology and Education. For example, learning Chinese can help a person later to learn Japanese more quickly, and learning repairing a smartphone can help a person to repair a tablet more easily. Fig. 2.4 demonstrates the difference between the learning processes of traditional machine learning and transfer learning techniques Pan and Yang [2010]. It can be clearly seen that traditional machine learning techniques must learn from scratch for every task or domain, while transfer learning can leverage previously learned knowledge from other tasks or domains to help the learning process on the current one.

2.3.1 Definition and Notations

In this section, the definition and notation of transfer learning is introduced.

Domain: A domain is defined as $D = \{\chi, P(X)\}$ Pan and Yang [2010], which includes:

(1) a feature space χ; and

(2) a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \cdots, x_d\} \in \chi$.

Task: A task is defined as $T = \{Y, f(\cdot)\}$ Pan and Yang [2010], which includes:

(1) a label space Y; and

(2) a prediction function $f(\cdot)$ learned from the training data containing pairs of $\{\vec{x}_i, y_i\}$, where $\vec{x}_i \in X$ and $y_i \in Y$. The function $f(\cdot)$ can be used to predict the label for the new incoming sample $\vec{x}$, and written as $P(y|\vec{x})$ from a probabilistic point of view.

Specifically, we denote the source domain as $D_S = \{(\vec{x}_{S_1}, y_{S_1}), \cdots, (\vec{x}_{S_N}, y_{S_N})\}$, where $\vec{x}_{S_i} \in \chi_S$, $y_{S_i} \in Y_S$, and the target domain as $D_T = \{(\vec{x}_{T_1}, y_{T_1}), \cdots, (\vec{x}_{T_N}, y_{T_N})\}$, where $\vec{x}_{T_i} \in \chi_T$, $y_{T_i} \in Y_T$. Take the prognosis of bladder cancer as an example, $\vec{x}_{S_i}$ is a patient instance from the United States and y_i is the corresponding outcome which is labeled as 'dead' or 'alive'. $\vec{x}_{T_i}$ is a patient instance from Hong Kong and y_{Ti} is its corresponding outcome.

Transfer learning: Given a source domain D_S, a source task T_S, a target domain D_T and target task T_T, transfer learning aims to help the learning of the target prediction function $f_T(\cdot)$ in D_T by leveraging the knowledge from the D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$ Pan and Yang [2010].

The condition $D_S \neq D_T$ implies that either (1) $\chi_S \neq \chi_T$, or (2) $P_S(X) \neq P_T(X)$. In the example of bladder cancer prognosis, case (1) may correspond to that the characteristics of two sets of patient records are different, and case (2) may correspond to that the source and target domains concentrate on different bladder cancer types. Transductive transfer learning refers to the situation that the source and target domains are different while the learning tasks are the same in both domains ($D_S \neq D_T$, $T_S = T_T$), which is also the focus of this study.

The condition $T_S \neq T_T$ implies that either (1) $Y_S \neq Y_T$, or (2) $f_S(x) \neq f_T(x)$. Take an example of prostate cancer diagnosis, case (1) may correspond to that the source dataset has class labels of 'benign' and 'prostate cancer', while the target one has three class labels of 'benign', 'insignificant prostate cancer' and 'significant prostate cancer', and case (2) may correspond to that the two sets are very imbalanced in terms of class labels. Inductive transfer learning refers to the situation that the learning tasks in the source and target domains are different ($T_S \neq T_T$).

In transfer learning, if the feature spaces in source and target domains share some commonality explicitly or implicitly, we imply that the source and target domains are related. It must be noticed that if the source and target domains are the same ($D_S = D_T$) and their tasks are the same ($T_S = T_T$), it becomes the traditional machine learning problem.

2.3.2 A Categorization of Transfer Learning Techniques

To have a better understanding of transfer learning, we have to consider three questions Pan and Yang [2010]: (1) what to transfer; (2) how to transfer; (3) when to transfer.

'What to transfer' concerns which part of knowledge and how much knowledge should be shared across domains. Regarding this, transfer learning techniques in literature can be broadly categorized into four groups.

Instance transfer Techniques under this category assume that a certain amount of source data can be reused for learning in the target domain using instance re-weighting and importance sampling techniques. In Huang et al. [2006b], a nonparametric method was proposed to directly get re-sampling weights without estimating distribution. Liu et al. [2002] presented a novel method to re-evaluate the training samples using in-target-domain probability using positive and unsupervised learning.

Feature representation transfer Techniques under this category aim to learn a

proper common feature representation to decrease the differences between the source and target domains and their classification errors. The knowledge to transfer is embedded in the new feature representation. Jebara proposed a transfer learning based framework for common feature and kernel selection in multiple SVMs constructed using different but related datasets. In Pan et al. [2011], a feature representation method called transfer component analysis (TCA) was proposed. By using TCA, the distance between domains can be reduced in a latent space for domain adaptation. Duan et al. [2012] proposed a method to first perform heterogeneous feature augmentation across different domains using two novel feature mapping functions. Then the newly generated feature representation is used for classification using the SVMs. Zuo et al. [2015a] developed a method using Stacked Denoising Autoencoder (SDA) to extract several feature spaces from the related domains. Then two fuzzy sets are introduced to analyze the variation of prediction accuracies using different feature spaces.

Relational knowledge transfer Techniques under this category does not assume that the data drawn from each domain be independent and identically distributed (i.i.d.) as traditionally assumed. It tends to transfer the relationship among data from a source domain to a target domain. In this context, statistical relational learning techniques are commonly used to solve these problems. For example, Mihalkova et al. [2007] proposed a Markov logic networks (MLN) based transfer system which can map the predictions in the source MLN to the target domain to further revise the mapping structure for the performance enhancement.

Parameter transfer Techniques under this category assume that the source and target domains share parameters or priors of the models at some extent. The knowledge to transfer is embedded into the shared parameters or priors. For example, Bonilla et al. [2007] proposed a novel model to study a shared covariance function on input

dependent features and a "free-form" covariance matrix among tasks. Schwaighofer et al. [2004] presented a method under a hierarchical Bayesian architecture which can learn a common prior to mean and covariance across domains. Gao et al. [2008] proposed a locally weighted ensemble framework to learn the information from multiple models for knowledge transfer. The weights are autonomously determined according to each constructed model's prediction ability. In Deng et al. [2013] and Deng et al. [2016], a knowledge-leverage-based Takagi-Sugeno-Kang fuzzy system (KL-TSK-FS) and an advanced version were proposed for parameter knowledge transfer on the target domain based on the traditional TSK-FS model.

After determining the knowledge to transfer, *'how to transfer'* is another problem to consider. Numerous techniques in computational intelligence have been applied to this area, including neural network transfer learning, Bayes transfer learning, fuzzy transfer learning Lu et al. [2015]. Liu et al. [2009] employed neural network to initialize the weights of labeled data in the source domain. Then the source data are placed into the trained neural network on the small target data to determine their contribution levels based on errors. In Luis et al. [2010], a novel aggregation method was designed for transfer learning which can estimate and weight the average confidence probability of the source task on its similarity to the target task. Zuo et al. [2015b] proposed a deep transfer learning method to extract hierarchical feature representations, such that the source domain knowledge in various feature spaces with different levels of abstraction can be investigated and transferred to the target domain. Behbood et al. [2014] et al. developed a fuzzy refinement domain adaptation method with the application of long-term bank failure prediction using similarity/dissimilarity concepts to modify the sample label information in the target domain.

'When to transfer' concerns the circumstances in which knowledge transfer can

or cannot be done. For example, in some scenarios there is no relation between the source and target domains, brute force transfer then may not work or even hurt the performance of learning in the target domain. This is also known as 'negative transfer' Rosenstein et al. [2005]. An ideal transfer learning method should benefit from related domains or tasks but avoid the negative transfers.

From literature, transfer learning has already been applied to some health-related applications Caruana [1998]; Silver and Mercer [2002, 2007]; Zhou et al. [2011, 2013]. However, there was little literature on using transfer learning methods for health care data prediction with the consideration of common problems encountered in the health care datasets, such as small sample size, missing data and class imbalances. This presents the opportunity for the applications of transfer learning in this specific area in this thesis.

2.4 Missing Data Problem and Solutions

Missing data is a common problem in the health field, which may be attributed to various causes. For example, participants may skip questions in surveys or drop out of experiments. Patients may not qualify for certain medical tests, or operators may take incorrect measurements during data acquisition. Any inappropriate treatment of missing data may consequently deteriorate classification performance and as such, the ability to appropriately handle missing data in classification problems has always been an essential demand. Numerous methods were proposed in literature to handle the classification with missing data. Generally, we can summarize the current solutions into four categories García-Laencina et al. [2010].

Methods in the first category simply remove incomplete samples and use complete

samples for classifier construction Little and Rubin [2014]. However, deleting samples may cause loss of information and introduce bias into the analysis, particularly when the missing values are not entirely randomly distributed Batista and Monard [2003]; Little and Rubin [2014].

Methods in the second category impute missing values and construct classifiers using the recovered dataset. The statistical imputation methods used include mean imputation Donders et al. [2006a], regression imputation Little and Rubin [2014] and so on. Mean imputation is the simplest: a missing value is estimated using the average value of the same feature. In regression imputation, a missing feature is estimated using a regression model constructed using non-missing features. The former method does not consider the correlations between missing and non-missing features Donders et al. [2006b] while the latter method only follows a single regression curve limited by the inherent variation in the data Little and Rubin [2014]. Imputation can also use machine learning techniques such as k-nearest neighbor (k-NN). In this method, the k-nearest neighboring complete samples are used to estimate the missing values. However, the performance of k-NN imputation is dependent on parameter settings, such as the value of k, the distance function, and the weighting function which no theoretical approaches can perfectly determine them. Moreover, the search for the nearest neighbors, i.e. the most similar complete samples is usually computationally expensive.

Methods in the third category estimate the data distributions of the complete and incomplete data portions in the dataset and make use of them for pattern classification. In this approach, an expectation maximization (EM) algorithm is commonly used to estimate the data distribution, and Bayesian decision theory is applied for classification Dempster et al. [1977]. However, the methods in this category have massive computational costs. The calculation of standard errors for the estimates Horton and

Kleinman [2012] and Monte Carlo implementation of the EM algorithm (MCEM) to model joint distribution of the covariates Ibrahim et al. [1999] are complicated procedures which restrict the practicality of these methods.

Methods in the fourth category handle missing data and construct the classifier at the same time. An increasing number of studies in this category have attempted to improve the generalization ability, and many have demonstrated satisfactory results García-Laencina et al. [2010]. In recent years, some works have concentrated on SVMs for handling missing data Chechik et al. [2006]; Pelckmans et al. [2005]; Shivaswamy et al. [2006]; Smola et al. [2005]; Zhang [2005]. Pelckmans et al. [2005] presented an idea to integrate the uncertainty caused by missing values into an appropriate risk function, and an extension of this work was based on a formulation of an SVMs and LS-SVMs classifier. In Smola et al. [2005], SVMs was incorporated into a Gaussian process to handle missing data. In this approach, how to estimate missing values is equivalent to finding efficient optimization methods such as the EM algorithm. Chechik et al. [2006] proposed a max-margin learning framework using a geometrically-inspired objective function to directly classify incomplete data with lower computational costs. Zhang [2005] were also inspired by the probability modeling approach, and proposed a new SVMs classification formulation, which handles missing data with an intuitive geometric interpretation. In Shivaswamy et al. [2006], a SVM based classifier was proposed for classification with missing data by using probabilistic classification constraints instead of linear ones.

Most existing solutions apply classifiers after the missing data has been preprocessed, such as case deletion and imputation. However, methods in the last category goes beyond the traditional. They use machine learning techniques to work directly with the missing data instead of hypothetically predicting missing values. Research

work on this topic is rapidly growing and many intelligent methods have achieved satisfactory performance. Nevertheless, so far there are no reports of using transfer learning as part of an approach. Additionally, most machine learning methods focus on improving generalization performance on missing data, but little attention has been given to how to detect corrupt and/or meaningless incomplete samples or features from the dataset simultaneously and quickly with an unbiased estimation guarantee to further improve the data quality.

2.5 Class Imbalance Problem and Solutions

Class imbalances refer to the problem that one class is represented by a large number of samples while the other class is only represented by a few. This is especially common in cancer diagnosis and prognosis, since there are a larger number of normal cases compared with the diseased ones in the clinical practice. Most machine learning methods which do not concern the class imbalance issue, have a tendency to be overwhelmed by the majority class and thus lead to the poor classification performance Chawla et al. [2004]. In literature, many relevant techniques and methods have been proposed to deal with this issue. They can be broadly summarized into two main categories, re-sampling, and case-sensitive learning.

Methods in the re-sampling category are designed to balance the class distribution via re-sampling the data input space, such as random over-sampling, focused over-sampling, random under-sampling and focused under-sampling. The main advantage of re-sampling methods is that they can work independently and be easily adapted to most prediction models. Random over-sampling aims to balance class distribution by randomly duplicating samples from minority class. Focused oversampling only

duplicates samples from minority class which are close to the boundaries between two classes. However, these methods may have the higher risk of over-fitting. Random under-sampling aims to balance class distribution by randomly removing samples from majority class. Focused under-sampling only discards samples from majority class which are further away from boundaries between two classes. The main disadvantage is that this may also delete some cases which are useful for classifier construction. Other advanced guided sampling methods, such as Tomek-Link (T-Link) Tomek [1976], Synthetic Minority Oversampling Technique (SMOTE) Chawla et al. [2002], Neighborhood Cleaning Rule (NCR) Laurikkala [2001], Edited Nearest Neighbor (ENN) Wilson [1972] and Condensed Nearest Neighbor (CNN) Angiulli [2005] were proposed to improve the performance of basic re-sampling.

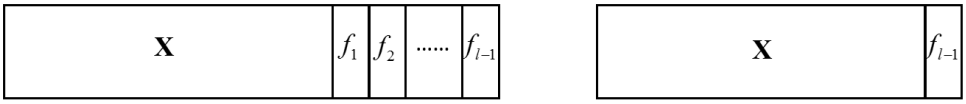
Methods falling in the cost-sensitive learning category assume that the actual misclassification costs vary with different kinds of errors in real-world applications. These methods, therefore, need to determine a cost matrix to embed the penalty of classifying a sample to a wrong class based on real situations in the learning process. For example, for the diagnosis of a disease, the significance level of recognizing a patient with disease is supposed to be higher than that of recognizing a normal case. Therefore, the cost of misclassifying a disease case is supposed to outweigh that of misclassifying a healthy one. After defining the cost matrix, the goal of cost-sensitive learning methods is to minimize the total misclassification cost when constructing the prediction model. Methods under this category include the modification of training data, moving the decision thresholds Elkan [2001]; Zhou and Liu [2006] or assigning weights to the training samples with different classes proportional to their corresponding misclassification costs Elkan [2001]; Ting [2002]. Another group of cost-sensitive learning methods can directly construct the prediction model by revising

the learning process or the objective function. For example, several work in literature studied on modifying the objective function of SVMs or extreme learning machine (ELMs) using a weighting strategy Casañola-Martin et al. [2016]; Maldonado and López [2014]; Phoungphol et al. [2012]; Wu et al. [2016]. Compared with re-sampling methods, cost-sensitive learning is more computationally efficient to handle a large amount of data with class imbalance issues Haixiang et al. [2017], and can work readily with the classifier learning algorithms, which provides the opportunities to further expand the work on imbalanced datasets.

2.6 Deep and Shallow Architectures

Most traditional machine learning methods, such as SVMs and LS-SVMs, fall under the shallow architecture in which only one processing layer exists. Although these shallow machines have well-performed in a wide range of applications, they experience difficulties in representing complex functions between the input and output. Conversely, deep architecture containing several processing layers of non-linear functions can learn a better underlying representation via a hierarchical structure, thus is more suitable to handle complex data.

There are different types of deep architectures, such as convolutional neural networks (CNNs) Krizhevsky et al. [2012], deep belief networks (DBNs) Hinton [2009], deep Boltzmann machines (DBM) Salakhutdinov and Hinton [2009] and deep auto encoders Wang et al. [2016b]. In this research, we are interested in a deep stacked architecture proposed by Deng and Yu [2011]. It is trained in a supervised and module-wise framework. Unlike other deep architectures such as DBNs that use back propagation over all modules, this hierarchy architecture stacks multiple modules



(a) (b)

Figure 2.5: TWO WAYS TO EXPAND THE FEATURE SPACE IN DEEP STACKED ARCHITECTURE

in a chain, and the predicted outputs from the previous adjacent module are fed into the data input of the higher module. Comparatively, deep stacked architecture is relatively simple and easy to implement. Moreover, the appended feature space can open the original data manifold to make it more separable. Following the philosophy of 'stacked generalization' Wolpert [1992], such architecture is expected to learn complex mappings between the input and output to further enhance the classification performance. Deep stacked architecture was first introduced in 2011 Deng and Yu [2011]. After that, a kernel version DSNs (K-DSNs) was proposed by Deng et al. [2012]. By using the kernel trick, the hidden neurons in each DSN layer becomes infinity. Another novel DSNs, which is called tensor-DSNs (T-DSNs) was presented by Hutchinson et al. [2013]. In this method, each module has a bilinear mapping from two hidden layers to the output layer by combining higher order statistics of the hidden binary features through a weight tensor. Vinyals et al. [2012] proposed a recursive perceptual representation using layers of linear SVMs and incorporating with random projections of weak predictions from each layer.

Through the deep stacked architecture, the appended feature space using the predictions from the module(s) of the previous layers open the original data manifolds such that the generalization performances may be improved. There are two ways to append the feature space with the increase of the depth in deep stacking architecture.

As depicted in Fig. 2.5 (a), the new feature space for the module in the l-th layer ($l \geqslant 2$) comes from the concatenation of the predicted outputs from all the modules of the previous layers and the original input features. Fig 2.5 (b) illustrates a different type which concatenates the predicted outputs from the previous module only. So far, little attention has been paid to embed transfer learning into the deep stacked architecture and apply it in the health analytics.

Chapter 3

An Output-based Transfer LS-SVMs Model for Bladder Cancer Prognosis with Insufficient Data

*The content of this Chapter was published in Wang et al. [2017a]:

Wang, G., Zhang, G., Choi, K.S., Lam, K.M. and Lu, J., "An output-based knowledge transfer approach and its application in bladder cancer prediction," in *Proceedings of 2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, May. 2017, Anchorage, USA (pp. 356-363).

3.1 Introduction

Accurate prediction and prognosis play an important role in heath care which can help doctors to make prompt treatment decisions on individual patients Knaus et al. [1991]; Ohno-Machado [2001]. For example, advanced bladder cancer patients with poor prognosis are advised not to take the ultra major surgery such as radical cystectomy.

There are two dilemmas frequently occur in many health prediction applications. First, the on-hand data cannot be completely put into the existing prediction model or on-line tool, since features in the new data do not perfectly match those required in the models or tools. As a result, some unique patient features collected in the current domain of interest might be wasted. Another significant dilemma is the lack of data due to the complex data collection procedures and privacy concerns, which may substantially deteriorate the prediction performance. To solve these problems, in this chapter, the output-based transfer LS-SVMs model with two versions is proposed. It stands on transfer learning mechanism which can leverage the probabilistic output information from an existing model or on-line tool (source domain) to help train a prediction model on the current domain of interest (target domain) with few samples. The influence level of the leveraged knowledge onto the target domain can be autonomously and quickly determined using the fast leave-one-out cross validation strategy.

The output-based transfer LS-SVMs model is evaluated on a real world small bladder cancer dataset for predicting 5-year mortality after radical cystectomy. The proposed model effectively handles the small sample size problem and produces better performance than traditional machine learning methods.

This chapter is organized as follows: Section 3.2 presents the proposed output-

based transfer LS-SVMs model with two versions. Section 3.3 shows the experimental evaluations and results. Finally, summary of this chapter are discussed in Section 3.4.

3.2 Output-based Transfer LS-SVMs Model with Insufficient Data

In this section, an output-based transfer LS-SVMs model is proposed to deal with small data from a transfer learning perspective. The proposed model can effectively learn a target domain with insufficient data by utilizing the output knowledge learned from the existing model or on-line tool in a related source domain. It can autonomously and rapidly determine the extent of output knowledge to transfer from the source domain to the target one using a proposed fast leave-one-out cross validation strategy. The output-based transfer LS-SVMs model is given as follows.

3.2.1 Inverted Pyramid Dataset

Given a small dataset in the target domain as $D_T = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_i, y_i), \cdots, (\vec{x}_N, y_N)\}$, where $\vec{x}_i = (x_1^i, x_2^i, \cdots, x_d^i) \in \mathbf{X}_T \subset \mathbf{R}^d$ and $y_i \in \mathbf{Y}_T = \{-1, 1\}$, $\mathbf{X}_T$ is the input dataset and $\mathbf{Y}_T$ is the corresponding output dataset. Each instance $\vec{x}_i$ has d features, i.e., f_1, f_2,$\cdots$, f_d. Since the existing model only requires a subset of features from the target domain, we project D_T to the source domain D_S using the common features, to fit into the existing model. $D_S = \{(\vec{x}_1', y_1), \cdots, (\vec{x}_i', y_i), \cdots, (\vec{x}_N', y_N)\}$, where $\vec{x}_i' = (x_1^i, x_2^i, \cdots, x_{d'}^i) \in \mathbf{X}_S \subset \mathbf{R}^{d'}$ and $y_i \in \mathbf{Y}_S = [0, 1]$. $\mathbf{X}_S$ is the input dataset and $\mathbf{Y}_S$ is the probabilistic outputs predicted from the existing model. Each instance $\vec{x}_i'$ contains d' features, i.e., f_1, f_2, $\cdots$, $f_{d'}$ ($d' < d$). We want to find a

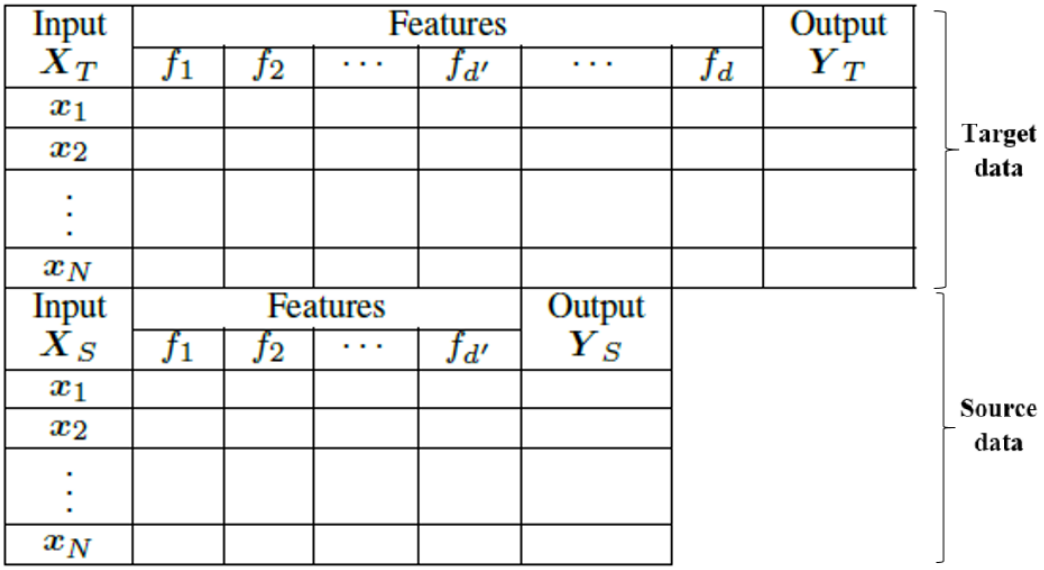


Figure 3.1: THE INVERTED PYRAMID DATASET IN WHICH $d' < d$

decision function $F : \mathbf{X}_T \rightarrow \mathbf{Y}_T$, such that the matching y for any new incoming instance $\vec{x}$ can be determined.

If we stack D_T onto D_S as demonstrated in Fig. 3.1, it shapes like an inverted pyramid. Thus we call the adopted dataset in the proposed model *inverted pyramid dataset*.

3.2.2 Framework of the Proposed Model

Fig. 3.2 illustrates the framework of the proposed model. The whole on-hand dataset is the target data D_T. Its subset which only contains the common features with the existing model or on-line tool is the source data D_S. After inputting the source data into the existing prediction model, we can obtain the predicted probabilistic outputs. These are the knowledge we attempt to learn and leverage to facilitate the learning process in the target domain with small data.

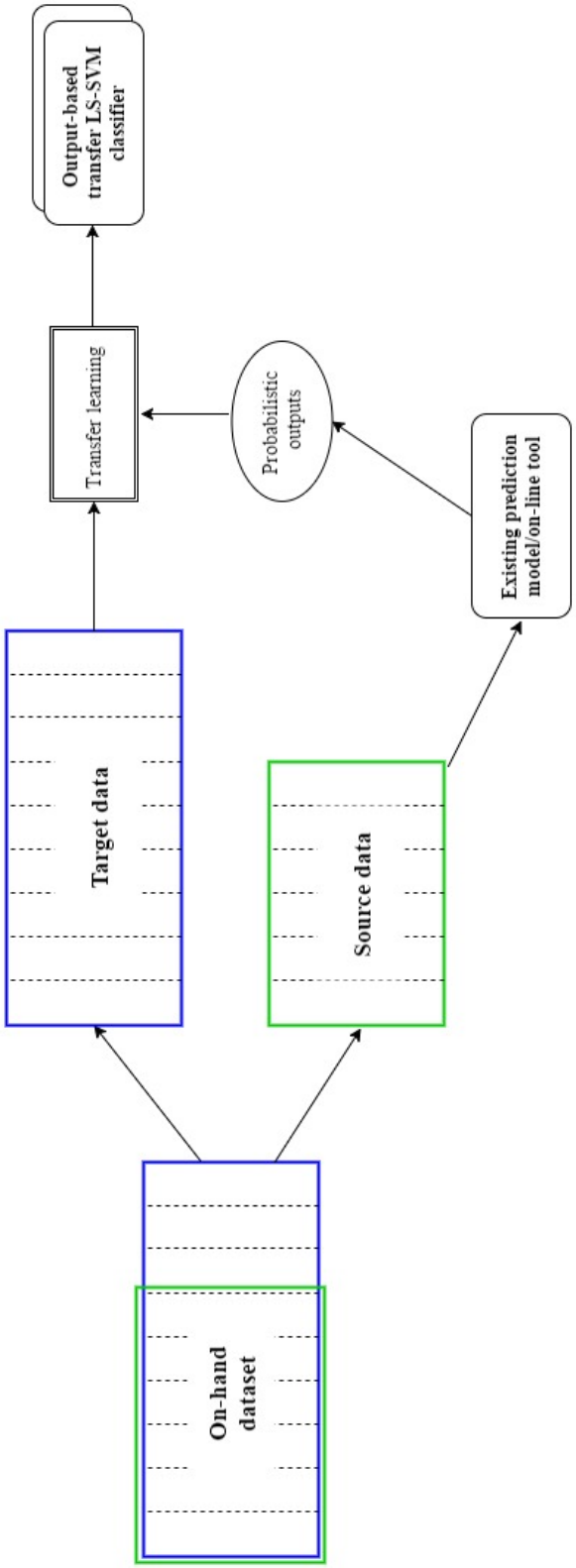


Figure 3.2: THE FRAMEWORK OF THE PROPOSED MODEL

3.2.3 Handle Probabilistic Outputs From the Existing Model

Most of the prediction models used in the medical field are built using the traditional statistical methods which produce probabilistic outputs. Hence, we design the proposed model to directly handle the probabilistic outputs from the existing model for the convenient use.

We put the source data D_S into the existing prediction model and obtain the corresponding probabilistic output p_i $(0 \leq p_i \leq 1,\ i = 1, 2, ..., N)$. p_i and $1 - p_i$ are the probabilities of $\vec{x}_i$ to be categorized into the positive and negative classes. We set a threshold θ to 0.5. If the output probability is greater than 0.5, $\vec{x}_i$ is classified into the positive class, otherwise, the negative class. For example, if an instance $\vec{x}_i$ obtains the probabilistic output of 0.65 from an existing prediction model, the probabilities of $\vec{x}_i$ being classified into the positive class and negative class are 0.65 and 0.35 $(1 - 0.65 = 0.35)$ respectively. According to threshold, $\vec{x}_i$ is classified into the positive class $(0.65 - 0.5 = 0.15 > 0)$ instead of the negative class $(0.35 - 0.5 = -0.15 < 0)$. Here, the processed probabilistic output $2p_i - 1 (i = 1, \cdots, N)$ are the knowledge we want to learn from the existing model or on-line tool.

3.2.4 Output-based Transfer LS-SVMs in Target Domain

The proposed model in the target domain is based on the LS-SVMs framework. LS-SVMs have two simplifications compared with the traditional SVMs Suykens et al. [2002]. First, the inequality constraints in the SVMs are replaced by the equality constraints. Second, the hinge loss function in the SVMs is replaced by a squared loss function. Therefore the LS-SVMs can be solved easily using a system of linear equations instead of quadratic programming in the SVMs. Besides, LS-

SVMs' analytical solution can help form the leave-one-out cross validation strategy for parameter tuning, which can reduce the high computational cost. The traditional LS-SVMs classifier formulation is formulated as follows:

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{3.1}$$

The input $\vec{x}_i$ can be classified based on the decision function:

$$\vec{w}^T\varphi(\vec{x}_i) + b \begin{cases} > 0 & \text{positive class} \\ < 0 & \text{negative class} \end{cases}$$

Therefore, to keep the sign of y_i to be the same of $(2p_i - 1)$ $(i = 1, 2..., N)$, $\sum_{i=1}^{N}(y_i - \xi_i)(2p_i - 1)$ should be as large as possible.

A weighting parameter μ is added to reflect the overall influence level of all the processed outputs from the existing model or on-line tool onto the constructed model. In the circumstances, all the output results from the existing model are utilized to make the maximum use to facilitate the learning process on the current domain. A fast leave-one-out cross-validation strategy is proposed to determine the optimal value of μ, which is discussed in Section 3.2.5. The proposed model with two versions are presented.

First version: The objective function of the LS-SVMs is modified into:

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \mu\sum_{i=1}^{N}(y_i - \xi_i)(2p_i - 1) \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{3.2}$$

After rewriting, we get

$$\begin{aligned} & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \mu\sum_{i=1}^{N}(y_i - \xi_i)(2p_i - 1) \\ =& \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 + \mu\sum_{i=1}^{N}\xi_i(2p_i - 1) - \mu\sum_{i=1}^{N}y_i(2p_i - 1) \end{aligned} \tag{3.3}$$

We exclude $\mu\sum_{i=1}^{N}y_i(2p_i - 1)$ since it is a constant value. We can then get the equivalent optimization problem using

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}(\xi_i + \frac{\mu}{2C}(2p_i - 1))^2 \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{3.4}$$

where $\frac{\mu}{2C}$ indicates the influence level of the processed probabilistic outputs onto the target domain. If μ is set to 0, Eq. (3.4) becomes the objective function of the traditional LS-SVMs. The Lagrangian L for this problem is

$$L = \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}(\xi_i + \frac{\mu}{2C}(2p_i - 1))^2 + \sum_{i=1}^{N}\alpha_i(y_i - \vec{w}^T\varphi(\vec{x}_i) - b - \xi_i) \tag{3.5}$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathbf{R}^N$ is the vector of all Lagrangian multipliers. The optimality conditions with respect to $\vec{w}$, ξ_i, b, α_i can be represented as follows:

$$\frac{\partial L}{\partial \vec{w}} = 0 \quad \Rightarrow \vec{w} = \sum_{i=1}^{N} \alpha_i \varphi(\vec{x}_i) \tag{3.6}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \xi_i = \frac{1}{C}[\alpha_i - \frac{\mu}{2}(2p_i - 1)] \tag{3.7}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \sum_{i=1}^{N} \alpha_i = 0 \tag{3.8}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \quad \Rightarrow y_i = \vec{w}^T \varphi(\vec{x}_i) + b + \xi_i \tag{3.9}$$

Combining Eq. (3.6) and Eq. (3.7) with Eq. (3.9), we obtain

$$\sum_{j=1}^{N} \alpha_i \varphi(\vec{x}_j)^T \varphi(\vec{x}_i) + b + \frac{\alpha_i}{C} = y_i + \frac{\mu}{2C}(2p_i - 1) \tag{3.10}$$

Using the kernel trick, $\varphi(\vec{x}_j)\varphi(\vec{x}_i)$ is replaced by $K(\vec{x}_j, \vec{x}_i)$, and the system of linear equations can be written in matrix as

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\Lambda & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \vec{y} + \frac{\mu}{2C}\vec{M} \\ 0 \end{bmatrix} \tag{3.11}$$

where Λ is a matrix in which each diagonal entry is 1 and all other entries are 0, $\vec{y}$ is the output vector of all the training samples, and $\vec{M} = (2p_1 - 1, 2p_2 - 1, ..., 2p_N - 1)^T$.

Lastly, the model parameters can be calculated simply by using matrix inversion:

$$\begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \mathbf{P} \begin{bmatrix} \vec{y} + \frac{\mu}{2C}\vec{M} \\ 0 \end{bmatrix} \tag{3.12}$$

where $\mathbf{P} = \mathbf{V}^{-1}$ and $\mathbf{V}$ is the first matrix on the left in Eq. (3.11). Once we determine the value of μ, $\vec{\alpha}$ and b can be easily calculated from Eq. (3.12). The resulting decision function is

$$\begin{aligned} F_1(\vec{x}_t) &= \vec{w}^T\varphi(\vec{x}_t) + b \\ &= \sum_{i=1}^{N} \alpha_i K(\vec{x}_i, \vec{x}_t) + b \end{aligned} \tag{3.13}$$

Second version: We replace $(y_i - \xi_i)$ in the first version with $(\vec{w}^T\varphi(\vec{x}_i) + b)$, therefore the objective function of the LS-SVMs becomes:

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \lambda\sum_{i=1}^{N}(2p_i - 1)(\vec{w}^T\varphi(\vec{x}_i) + b) \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{3.14}$$

The Lagrangian J of Eq. (3.14) gives the unconstrained minimization problem:

$$J = \tfrac{1}{2}\vec{w}^2 + \tfrac{C}{2}\textstyle\sum_{i=1}^{N}\xi_i^2 - \lambda\sum_{i=1}^{N}(2p_i - 1)(\vec{w}^T\varphi(\vec{x}_i) + b) + \sum_{i=1}^{N}\alpha_i(y_i - \vec{w}^T\varphi(\vec{x}_i) - b - \xi_i) \tag{3.15}$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathbf{R}^N$ is the vector of all Lagrangian multipliers. The optimality conditions for this problem, respect to $\vec{w}$, ξ_i, b, α_i, can be expressed as follows

$$\frac{\partial L}{\partial \vec{w}} = 0 \quad \Rightarrow \vec{w} = \mu\sum_{i=1}^{N}(2p_i - 1)\varphi(\vec{x}_i) + \sum_{i=1}^{N}\alpha_i\varphi(\vec{x}_i) \tag{3.16}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \xi_i = \frac{\alpha_i}{C} \tag{3.17}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \mu \sum_{i=1}^{N} (1 - 2p_i) = \sum_{i=1}^{N} \alpha_i \tag{3.18}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \quad \Rightarrow y_i = \vec{w}^T \varphi(\vec{x}_i) + b + \xi_i \tag{3.19}$$

Combining Eq. (3.16) and Eq. (3.17) with Eq. (3.19), we obtain

$$\sum_{i=1}^{N} \alpha_i \varphi(\vec{x}_i)^T \varphi(\vec{x}_j) + b + \frac{\alpha_i}{C} = y_i - \lambda \sum_{i=1}^{N} (2p_i - 1) \varphi(\vec{x}_i)^T \varphi(\vec{x}_j) \tag{3.20}$$

Using the kernel trick, $\varphi(\vec{x}_j)\varphi(\vec{x}_i)$ is replaced with $K(\vec{x}_j, \vec{x}_i)$, and the system of linear equations can be written in matrix form:

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\vec{\Lambda} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \vec{y} - \lambda \vec{Z} \\ \lambda \sum_{i=1}^{N} (2p_i - 1) \end{bmatrix} = \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \lambda \begin{bmatrix} \vec{Z} \\ \sum_{i=1}^{N} (1 - 2p_i) \end{bmatrix} \tag{3.21}$$

where $\vec{\Lambda}$ is a matrix in which each diagonal entry is one and all other entries are zero, $\vec{y}$ is the output vector of all the samples in the training dataset, and $\vec{Z} = (\sum_{j=1}^{N} (2p_i - 1) \varphi(\vec{x}_j)^T \varphi(\vec{x}_1), \sum_{j=1}^{N} (2p_i - 1) \varphi(\vec{x}_j)^T \varphi(\vec{x}_2), \cdots, \sum_{j=1}^{N} (2p_i - 1) \varphi(\vec{x}_j)^T \varphi(\vec{x}_N))$. The model parameters can be calculated simply by using the matrix inversion:

$$\begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \mathbf{P} \begin{bmatrix} \vec{y} - \lambda \vec{Z} \\ \lambda \sum_{i=1}^{N} (1 - 2p_i) \end{bmatrix} = \mathbf{P} \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \lambda \mathbf{P} \begin{bmatrix} \vec{Z} \\ \sum_{i=1}^{N} (1 - 2p_i) \end{bmatrix} \tag{3.22}$$

where $\mathbf{P} = \mathbf{V}^{-1}$ and $\mathbf{V}$ is the first matrix on the left in Eq. (3.21). Once we have obtained μ, $\vec{\alpha}$ and b can be calculated from Eq. (3.22). Combined with Eq. (3.16), the decision function for the new sample $\vec{x}_t$ becomes

$$\begin{aligned} F_2(\vec{x}_t) &= \vec{w}^T \varphi(\vec{x}_t) + b \\ &= \mu \sum_{i=1}^{N} (2p_i - 1) K(\vec{x}_i, \vec{x}_t) + \sum_{i=1}^{N} \alpha_i K(\vec{x}_i, \vec{x}_t) + b \\ &= \sum_{i=1}^{N} (\mu(2p_i - 1) + \alpha_i) K(\vec{x}_i, \vec{x}_t) + b \end{aligned} \tag{3.23}$$

3.2.5 Fast Leave-one-out Cross Validation Strategy for Parameter Tuning

From Section 3.2.4, we can see that the classification performance of the proposed model replies on the value of parameter μ (version 1) and λ (version 2). Traditionally, leave-one-out cross-validation strategy is commonly used as an almost unbiased estimator for parameter tuning, particularly on the small datasets; however, it is very time-consuming. In this section, we propose a fast leave-one-out cross validation for the proposed model to determine the optimal value of μ in Eq. (3.12) and λ in Eq. (3.22) respectively.

First version: We decompose $\mathbf{V}$ into block presentation with the isolation of the first row and column as follows:

$$\mathbf{V} = \begin{bmatrix} \mathbf{K} + \frac{1}{C}\vec{\Lambda} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} = \begin{bmatrix} v_{11} & \mathbf{v}_1^T \\ \mathbf{v}_1 & \mathbf{V}_{(-1)} \end{bmatrix} \tag{3.24}$$

$\vec{\alpha}_{(-i)}$ and $b_{(-i)}$ are denoted as the model parameters in the i-th iteration of the leave-one-out cross validation. In the first iteration, we have:

$$\begin{bmatrix} \vec{\alpha}_{(-1)} \\ b_{(-1)} \end{bmatrix} = \mathbf{P}_{(-1)} \left(\vec{y}_{(-1)} + \frac{\mu}{2C} \vec{M} \right) \tag{3.25}$$

where $\mathbf{P}_{(-1)} = \mathbf{V}_{(-1)}^{-1}$ and $y_{(-1)} = [y_2, y_3, ..., y_N, 0]^T$. The predicted label of i-th sample is denoted as $\tilde{y}_i$. The predicted label of the first training sample can be represented using

$$\tilde{y}_1 = \mathbf{v}_1^T \begin{bmatrix} \vec{\alpha}_{(-1)} \\ b_{(-1)} \end{bmatrix} + \frac{\mu}{2C} \vec{M}_{(-1)} = \mathbf{v}_1^T \mathbf{P}_{(-1)} \left(\vec{y}_{(-1)} + \frac{\mu}{2C} \vec{M}_{(-1)} \right) + \frac{\mu}{2C} \vec{M}_{(-1)} \tag{3.26}$$

Considering the last N equations in Eq. (3.11), we obtain $\left[\mathbf{v}_1 \, \mathbf{V}_{(-1)}\right] \left[\vec{\alpha}^T, \, b\right]^T = \left(\vec{y}_{(-1)} + \frac{\mu}{2C} \vec{M}_{(-1)} \right)$. Eq. (3.26) can be rewritten into

$$\begin{aligned} \tilde{y}_1 &= \mathbf{v}_1^T \mathbf{P}_{(-1)} \left[\mathbf{v}_1 \, \mathbf{V}_{(-1)}\right] \left[\alpha_1, \cdots, \alpha_N, b\right]^T - \frac{\mu}{2C} \vec{M}_{(-1)} \\ &= \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1 \alpha_1 + \mathbf{v}_1^T \left[\alpha_2, \cdots, \alpha_N, b\right]^T - \frac{\mu}{2C} \vec{M}_{(-1)} \end{aligned} \tag{3.27}$$

In Eq. (3.11), the first equation of the system is $y_1 + \frac{\mu}{2C} \vec{M}_{(-1)} = v_{11}\alpha_1 + \mathbf{v}_1^T \left[\alpha_2, \alpha_3, \cdots, \alpha_N, b\right]^T$. Combined with Eq. (3.27), we get $\tilde{y}_1 = y_1 - \alpha_1 (v_{11} - \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1)$. By using $\mathbf{P} = \mathbf{V}^{-1}$ and the block matrix inversion lemma, we obtain

$$\mathbf{P} = \begin{bmatrix} u^{-1} & -u^{-1} \mathbf{v}_1 \mathbf{P}_{-1} \\ \mathbf{P}_{(-1)} + u^{-1} \mathbf{P}_{(-1)} \mathbf{v}_1^T \mathbf{v}_1 \mathbf{P}_{(-1)} & -u^{-1} \mathbf{P}_{(-1)} \mathbf{v}_1^T \end{bmatrix} \tag{3.28}$$

where $u = v_{11} - \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1$. Since the system of linear equations in Eq. (3.11) is not sensitive to permutations of the ordering of the equations, we obtain

$$\tilde{y}_i = y_i - \alpha_i / P_{ii} \tag{3.29}$$

By defining $\left[\vec{\alpha}^{'T}, b^{'}\right]^T = \mathbf{P}\left[\vec{y}^T, 0\right]$, $\left[\vec{\alpha}^{''T}, b^{''}\right] = \mathbf{P}\left[\vec{M}^T, 0\right]$, and $\vec{\alpha} = \vec{\alpha}^{'} + \frac{\mu}{2C}\vec{\alpha}^{''}$, we obtain

$$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{P_{ii}} - \frac{\frac{\mu}{2C}\alpha_i^{''}}{P_{ii}} \tag{3.30}$$

It can be seen that we only need to calculate the matrix inversion for $\mathbf{P}$ once, the leave-one-out cross-validation estimate can be calculated, which is much less computational expensive. It is assumed that the optimal μ will retain the sign of $\tilde{y}_i$ to be the same of y_i for all the samples in the training dataset. However, this might result in local minima issues due to its non-convex formulation. Thus, we use the following loss function, which is similar to hinge loss:

$$l(\tilde{y}_i, y_i) = |1 - \tilde{y}_i y_i|_+ = \left| y_i \frac{\alpha_i^{'} + \frac{\mu}{2C}\alpha_i^{''}}{P_{ii}} \right|_+ \tag{3.31}$$

where $|x|_+ = \max\{0, x\}$. This is a convex upper bound to the leave-one-out misclassification loss. It prefers the solutions in which $\tilde{y}_i$ has an absolute value that is equal to or bigger than 1 and retain the same sign of y_i. The objective function becomes:

$$\begin{aligned} &\min_{\mu} \sum_{i=i}^{N} l(\tilde{y}_i, y_i) \\ &\text{s.t} \qquad 0 \leq \mu \leq D \end{aligned} \tag{3.32}$$

where D is a constant. This optimization process can be implemented by using a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 3.1.

Second version: Similar to the first version, by defining $\left[\vec{\alpha}'^T, b'\right]^T = \mathbf{P}\left[\vec{y}^T, 0\right]$, $\left[\vec{\alpha}''^T, b''\right] = \mathbf{P}\left[\vec{M}^T, 0\right]$, and $\vec{\alpha} = \vec{\alpha}' - \lambda\vec{\alpha}''$, the predicted label of the i-th training sample can be represented as

$$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{P_{ii}} + \frac{\lambda\alpha_i^{''}}{P_{ii}} \tag{3.33}$$

The same loss function in the first version is adopted to avoid local minima issues:

$$l(\tilde{y}_i, y_i) = |1 - \tilde{y}_i y_i|_+ = \left| y_i \frac{\alpha_i^{'} - \lambda\alpha_i^{''}}{P_{ii}} \right|_+ \tag{3.34}$$

where $|x|_+ = \max\{0, x\}$. Finally, the objective function is:

$$\begin{aligned} &\min_{\lambda} \sum_{i=i}^{N} l(\tilde{y}_i, y_i) \\ &\text{s.t} \qquad 0 \leq \lambda \leq D \end{aligned} \tag{3.35}$$

where λ is a constant. This optimization process can be implemented by a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 3.2.

3.2.6 Computational Complexity

The computational cost of the proposed fast leave-one-out cross validation can be represented using $O(N^3 + N)$. The first part $O(N^3)$ is the computational cost of the matrix inversion for $\mathbf{P}$, which is related to the sample size of the training dataset.

Algorithm 3.1: Projected Sub-gradient Descent Algorithm for version 1
Input: $\vec{\alpha}^{'}$, $\vec{\alpha}^{''}$
Initialize: $\mu \leftarrow 0$ and $t \leftarrow 1$
Repeat
$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{P_{ii}} - \frac{\frac{\mu}{2C}\alpha_i^{''}}{P_{ii}}$, $i = 1, 2, ..., N$
$d_i \leftarrow \vec{1}\{\tilde{y}_i y_i > 0\}$, $i = 1, 2, ..., N$
$\mu \leftarrow \mu - \frac{1}{\sqrt{t}} d_i y_i \frac{\alpha_i^{''}}{P_{ii}}$
If $\mu > D$ then $\mu \leftarrow D$
End if
$\mu \leftarrow \max(\mu, 0)$
$t \leftarrow t + 1$
Until convergence
Output: μ

Algorithm 3.2: Projected Sub-gradient Descent Algorithm for version 2
Input: $\vec{\alpha}^{'}$, $\vec{\alpha}^{''}$
Initialize: $\lambda \leftarrow 0$ and $t \leftarrow 1$
Repeat
$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{P_{ii}} + \frac{\lambda\alpha_i^{''}}{P_{ii}}$, $i = 1, 2, ..., N$
$d_i \leftarrow \vec{1}\{\tilde{y}_i y_i > 0\}$, $i = 1, 2, ..., N$
$\lambda \leftarrow \lambda - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i y_i \frac{\alpha_i^{''}}{P_{ii}}$
If $\lambda > D$ then $\lambda \leftarrow D$
End if
$\lambda \leftarrow \max(\lambda, 0)$
$t \leftarrow t + 1$
Until convergence
Output: λ

The second part $O(N)$ is the computational cost of optimizing Eq. (3.32) in Algorithm 3.1 or Eq. (3.35) in Algorithm 3.2.

In terms of the traditional leave-one-out cross-validation with grid search from $[\mu_1, \mu_2, ..., \mu_T]$ for μ in the first version and from $[\lambda_1, \lambda_2, ..., \lambda_T]$ for λ in the second version, the whole computational cost can be represented using $T * O(N^3 * N) = T * O(N^4)$, which is much more computationally expensive.

3.3 A Case Study on a Real World Bladder Cancer Dataset

3.3.1 Data Collection and Existing Prediction Model

The dataset employed in this study originated from a retrospective review on the 5-year survival of patients treated with radical cystectmy for bladder cancer Chan et al. [2013]. It consists of 117 patient clinical records after radical cystectomy within the period from 2003 to 2011 in a urology unit in Hong Kong. The data include the results from preoperative evaluations, such as transurethral resection (TUR) of the bladder tumor and computed tomography (CT). Tumor stages were classified based on the 2002 American Joint Committee on Cancer guidelines. There was no loss of follow-up during the study. The 30-day mortality, 5-year cancer-specific mortality, 5-year other-cause mortality, and the 5-year overall mortality were 3%, 33%, 22% and 55% respectively. The baseline characteristics of this cohort are described in Table. 3.1. Referring to the results of statistical analysis reported in Chan et al. [2013] and doctors' experience, the inputs and output to construct the prediction model are selected and listed in Table. 3.2.

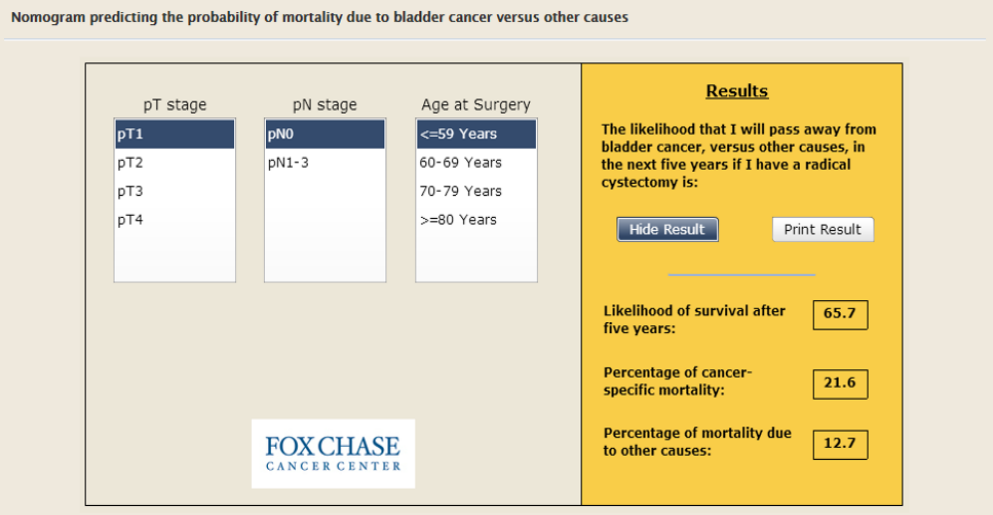


Figure 3.3: ONLINE NOMOGRAM PREDICTING THE PROBABILITY OF MORTALITY DUE TO BLADDER CANCER VERSUS OTHER CAUSES

The existing prediction model can be accessed from CancerNomograms.com Nomograms. It was created on 11,260 bladder cancer patients treated with radical cystectomy between 1988 and 2006 within 17 Surveillance, Epidemiology, and End Results registries in the United States Lughezzani et al. [2011]. Patients were stratified into 20 groups based on patient age, tumour stage and lymph node stage following radical cystectomy. A smoothed Poisson regression model was constructed to predict the probability of overall mortality, cancer-specific mortality and mortality due to other causes after five years. In this study, we only focus on the five-year overall mortality outcome. The user interface of the existing model is shown in Fig. 3.3.

It can be observed that a subset of the clinical dataset could be fit into this on-line tool which contains the features 'age at operation', 'tumour stage' and 'lymph node stage', and the corresponding probabilistic outputs could thus be obtained. This subset and the whole clinical dataset form the *inverted pyramid dataset* which is appropriate to be used in the proposed output-based transfer LS-SVMs.

Table 3.1: BASELINE CHARACTERISTICS OF THE COHORT

Demographics/Characteristics	No. (%) of patients or mean ± standard deviation		
	Overall	Age<=75 years	Age>75 years
No. of patients	117(100)	83(71)	34(29)
Mean age (years)	68±10	64±9	80±4
Gender			
Male	99 (85)	72 (87)	27 (79)
Female	18 (15)	11 (13)	7 (21)
Cystectomy			
Open	71 (61)	52 (63)	19 (56)
Laparoscopic/ robotic-assisted	46 (39)	31 (37)	15 (44)
Urinary diversion			
Ileal conduit	96 (82)	62 (75)	34 (100)
Neo-bladder/ continence diversion	21 (18)	21 (25)	0
Hospital stay duration (mouths)			
Mean	22±17	23±18	22±15
Median	18	17 (14-26)	18 (12-24)
Preoperative serum albumin level (g/L)	38±6	39±6	36±7
CCI			
0	77 (66)	60 (72)	17 (50)
1-2	38 (32)	22 (27)	16 (47)
>=3	2 (2)	1 (1)	1 (3)
Tumour grade			
G0	5 (4)	5 (6)	0
G2	24 (21)	17 (20)	7 (21)
G3	69 (59)	48 (58)	21 (62)
CIS	4 (3)	4 (5)	0
N/A	15 (13)	9 (11)	6 (18)
Tumour stage			
NMIBC	34 (29)	25 (30)	9 (26)
T0	11	6	5
Tis	7	7	0
Ta	4	3	1
T1	12	9	3
MIBC	82 (70)	57 (69)	25 (74)
T2	32	23	9
T3	32	23	9
T4	18	11	7
N/A			
Lymph node			
N0	88 (75)	65 (78)	23 (68)
N1	6 (5)	5 (6)	1 (3)
N2	14 (12)	8 (10)	6 (18)
N3	1 (1)	0	1 (3)
N/A	8 (7)	5 (6)	3 (9)
Follow-up (months)			
Mean	31±29	34±31	24±23
Range	(0-100)	(0-100)	(0-77)

Table 3.2: THE INPUTS AND OUTPUT OF THE PREDICTION MODEL

Input	Value(s)
Gender	1 (femele) 2 (male)
Age at operation	Normalized to [0, 1]
Surgery Type	1 (open surgery) 2 (laparoscopic surgery) 3 (robotic surgery)
Preoperative serum albumin level	Normalized to [0,1]
Tumor stage	1 (T1) 2 (T2) 3 (T3) 4 (T4)
Lymph node stage	0 (N0) 1 (N1) 2 (N2) 3 (N3)
Overall cancer stage	1 (Stage I) 2 (Stage II) 3 (Stage III) 4 (Stage IV)
Follow up period	Normalized to [0,1]
Grade	1 (Grade 1) 2 (Grade 2) 3 (Grade 3)
Type of diversion	1 (ideal conduit) 2 (neo bladder)
Ouput	**Value(s)**
5-year overall mortality	1 (dead) 0 (alive)

3.3.2 Experimental Design

The main purpose of the experiment is to evaluate the performances of the proposed output-based transfer LS-SVMs with two versions for bladder cancer prognosis and compare the results with those using traditional machine learning methods, including LS-SVMs, SVMs, BPNNs and k-NN.

The subset of the real world dataset was fed into the existing on-line tool to obtain the corresponding probabilistic outputs for the proposed method. This is the knowledge we want to leverage to assist the target domain construction. Both versions, called proposed model-v1 and proposed model-v2, were applied to the whole clinical dataset to train a prediction model with the help of the learned knowledge from the previous step. The comparative methods were directly applied on the whole clinical dataset for model construction.

To make our comparison fair, we used grid search with cross validation to discover the optimal parameters during the training process. We used the polynomial kernel Wang et al. [2015] and chose the trade-off parameter C and the degree parameter γ by searching $C \in \{1, 10, 50, 100, 150, 200, 250\}$ and $\gamma \in \{2e-5, 2e-4, 2e-3, 2e-2, 2e-1, 1\}$ for the proposed method, LS-SVMs and SVMs. For BPNNs, the number of hidden neurons, the momentum and the learning rate were selected from $\{3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29\}$, $\{0, 0.2, 0.5, 0.9\}$ and $\{0.01, 0.05, 0.09\}$ respectively. For k-NN, the value of the neighbouring parameter k was experimentally selected from $\{10, 12, 15, 18, 20\}$. The parameter settings are summarized in the Table 3.3. All the experiments are implemented using 64 bit MATLAB on a computer with Intel Core i5-6300 2.40 GHz CPU and 8.00GB RAM.

Table 3.3: PARAMETER SETTINGS OF THE PROPOSED AND COMPARATIVE METHODS

Models	Proposed model v1 & v2	LS-SVMs	SVMs	BPNNs	k-NN
Parameter settings	C=150 $\gamma = 2e-2$	C=150 $\gamma = 2e-2$	C=200 $\gamma = 2e-2$	number of hidden neurons=15 learning rate=0.05 momentum=0.9	k=18

3.3.3 Results Analysis

In the experiments, we compare the performance of the proposed model with four traditional machine learning methods for predicting the five-year overall mortality of bladder cancer patients after radical cystectomy. From the experimental results presented in Table 3.4, it can be seen that two versions of our proposed model achieved the highest classification accuracy of 0.7697 and 0.7618 respectively. Their performances also stood out in terms of sensitivity, specificity, precision and area under the curve (AUC) compared to those using other methods. BPNNs and k-NN obtained comparatively low performance, with a mean classification accuracy of 0.6758 and 0.7061 respectively. The standard LS-SVMs and SVMs exhibited better performance than BPNNs and k-NN with a mean accuracy of 0.7424 and 0.7485 respectively. The ROC curves of the proposed model with two versions and the comparison methods are shown in Fig. 3.4 and Fig. 3.5 respectively. In addition, we also directly use the existing model to predict the mortality outcome on the same dataset and obtain an accuracy of 0.6330. The low classification performance is mainly due to the fact that the existing model uses limited number of features for prediction.

The experimental results demonstrate the proposed models v1 and v2 have the superior advantages over directly using the existing on-line tool and can achieve comparatively better classification performance than traditional machine learning

Table 3.4: PERFORMANCE RESULTS OF THE PROPOSED MODELS AND COMPARATIVE METHODS

Performance		Proposed model v1	Proposed model v2	LS-SVMs	SVMs	BPNNs	k-NN
Accuracy	Mean	**0.7697**	**0.7618**	0.7424	0.7485	0.6758	0.7061
	SD	0.0456	0.0621	0.0860	0.0655	0.0516	0.0516
	Max	0.8788	0.8824	0.8182	0.7879	0.7879	0.8182
	Min	0.6970	0.6176	0.6364	0.6364	0.6061	0.6061
Sensitivity	Mean	0.7848	0.7829	0.7805	0.7551	0.6542	0.6940
	SD	0.0678	0.0993	0.1365	0.0806	0.0900	0.0876
Specificity	Mean	0.7579	0.7433	0.7216	0.7507	0.7119	0.7287
	SD	0.0625	0.1121	0.1315	0.1105	0.1368	0.0688
Precision	Mean	0.7805	0.7834	0.7462	0.7672	0.7485	0.7497
	SD	0.0612	0.0901	0.1261	0.1154	0.1124	0.0767
AUC	Mean	0.8385	0.8369	0.8016	0.7757	0.6667	0.7869
	SD	0.0720	0.0839	0.0676	0.0746	0.1210	0.0698

methods without knowledge transfer. Exploring knowledge from the probabilistic output using the existing on-line model can benefit model construction on the on-hand clinical dataset, showing that it has great potential for real-world implementation. The main advantage of this study is that by using either version of the proposed model, we are able to construct a reliable model on a small number of samples. The improved accuracy of prognosis could assist doctors to advise the best treatment plans for patients. Moreover, the proposed model has the capability to work readily with the existing model or on-line tool in the medical field by leveraging their probabilistic output knowledge to help the learning process in the current domain of interest. Importantly, it is not necessary to know the details of the existing model and its training data, which is very practical in most real-world scenarios where the data and its modeling are private. In other words, the proposed approach can be regarded as a module-based model which has the capacity be extended to various medical problems and situations.

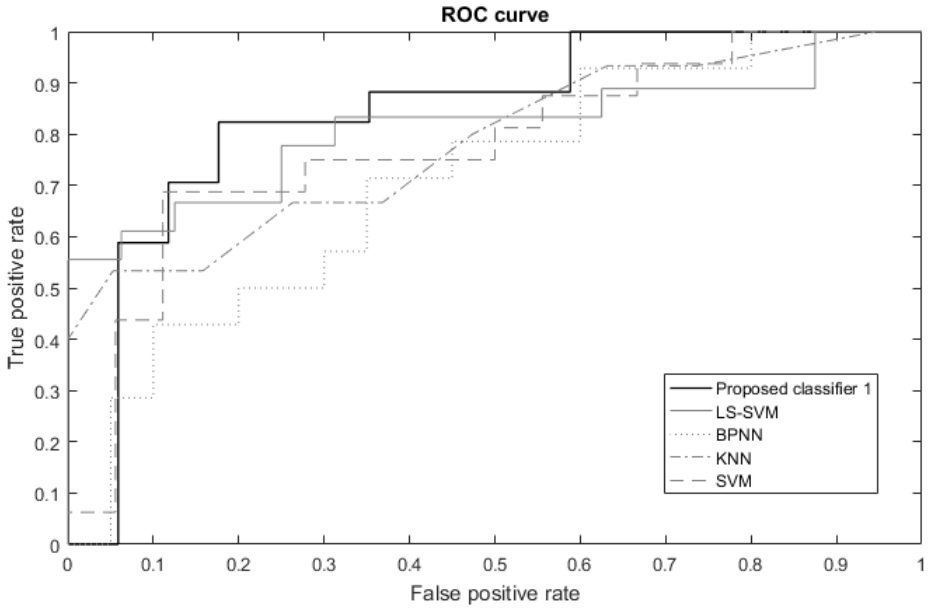


Figure 3.4: ROC CURVE OF THE PROPOSED MODEL-V1 AND COMPARATIVE METHODS

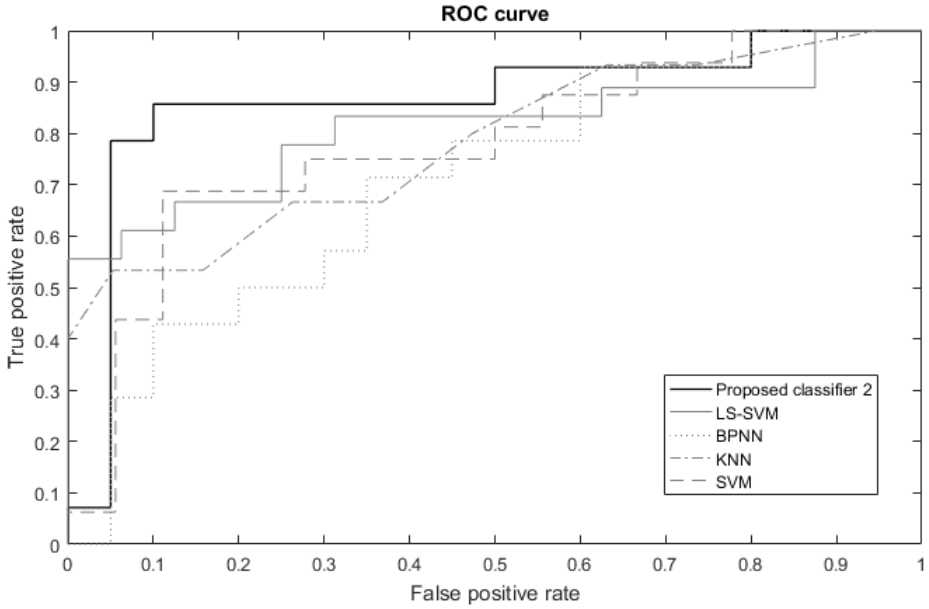


Figure 3.5: ROC CURVE OF THE PROPOSED MODEL-V2 AND COMPARATIVE METHODS

3.4 Summary

This chapter proposes a novel output-based transfer LS-SVMs model with two versions to deal with small sample problems in health care data prediction. The proposed model can leverage the probabilistic outcomes from the existing model or on-line tool to make the maximum use of small data and guarantee an enhanced generalization capability. The proposed model can autonomously and quickly decide the influence level on the target domain caused by the leveraged output knowledge using a fast leave-one-out cross validation strategy. Importantly, the output-based transfer LS-SVMs model works readily with the statistical prediction software or on-line tool where the data and and modeling details are usually private. The proposed model is evaluated on a real world bladder cancer dataset for prognosis and compared with traditional machine learning methods. The experimental results show that the proposed model has good performances with insufficient data and can be well implemented in the real world health care applications.

Chapter 4

A Novel Additive LS-SVMs Model for Predicting Elderly QOL with Missing Data

*The content of this Chapter was published in Wang et al. [2016a]:

Wang, G., Deng, Z. and Choi, K.S., 2016. "Tackling missing data in community health studies using additive LS-SVM classifier," *IEEE Journal of Biomedical and Health Informatics*, 22(2), pp. 579-587.

4.1 Introduction

Missing data is a common issue in health care data, and is attributed to various causes. For example, participants may skip questions in the survey or drop out of the study. Patients may not quality for certain medical tests, or operators take the incorrect measurements during the data collection. Any inappropriate treatment of missing data may consequently deteriorate prediction performance and, as such, the ability to appropriately handle classification with missing data has always been an essential demand.

In this chapter, a novel additive LS-SVMs model is proposed to deal with classification with missing data. Instead of handling missing data in the data pre-processing process separately, such as imputation, the proposed model can perform classification simultaneously with the evaluation of influences on the classification error caused by missing features using the fast leave-one-out cross validation strategy. Moreover, the significance levels of missing features can further guide health professionals to improve the future data collection process.

The novel additive LS-SVMs model is evaluated on a real-world community health care dataset with missing data for predicting the elderly quality of life (QOL). The proposed model outperforms four conventional missing data treatment methods, including case deletion, feature deletion, mean imputation and k-NN imputation, showing a promising potential for tackling missing data in health care predictive analytics.

This chapter is organized as follows. Section 4.2 presents the proposed novel additive LS-SVMs model. Section 4.3 shows the experimental evaluations and results on the real world community health care dataset. Section 4.4 concludes the chapter.

4.2 Novel Additive LS-SVMs Model with Missing Data

The proposed novel additive LS-SVMs model and its fast leave-one-out cross-validation strategy to estimate the influence levels on the classification error caused by missing features are given below.

4.2.1 Problem Description

Given a training dataset $\mathbf{T}$ with N samples, an input dataset $\mathbf{X}$ and the corresponding output dataset $\mathbf{Y}$, where $\mathbf{T} = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_N, y_N)\}$, $\vec{x}_i = (x_1^i, x_2^i, \cdots, x_l^i, \cdots, x_d^i) \in \mathbf{X} \subset \mathbf{R}^d$ and $y_i \in \mathbf{Y} = \{+1, -1\}$. Each sample $\vec{x}_i$ contains d features. Fig. 4.1(a) shows the scenario of a normal classification problem, where all the features have values. On the contrary, Fig. 4.1(b) shows the scenario of the classification with missing data problem, where data for certain features are missing and denoted by the symbol '?'.

4.2.2 Novel Additive LS-SVMs Model

The upper bound of the classification error caused by the l-th missing feature is considered and denoted as c_l ($l = 1, 2, \cdots, d$). Therefore, for the i-th sample, the upper bound of the classification error caused by all the missing features can be represented as $\sum_{l=1}^{d} c_l I_l^i$, where I_l^i is an indicator defined as

$$I_l^i = \begin{cases} 1 & \text{if the value of the } l\text{-th feature in } \vec{x}_i \text{ (i.e. } x_l^i\text{) is missing} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

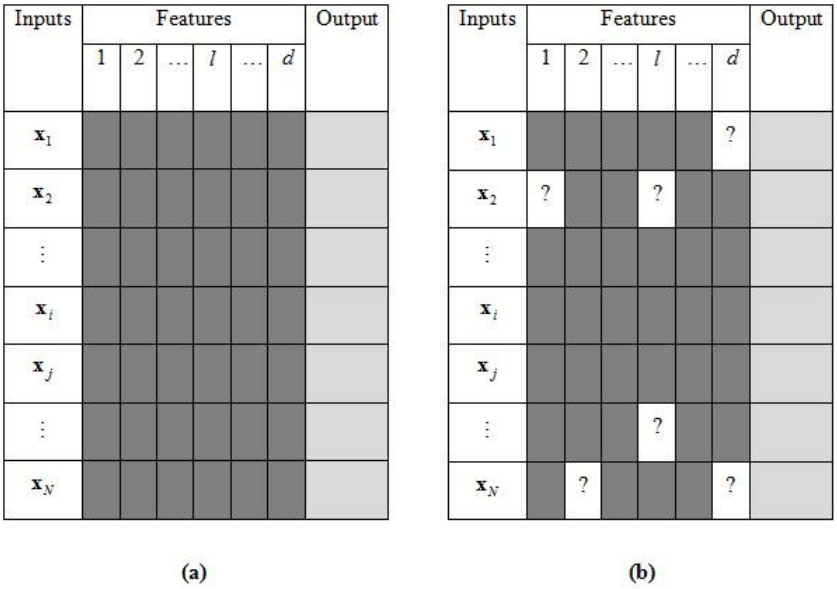


Figure 4.1: PATTERN CLASSIFICATION ON (A) COMPLETE AND (B) INCOMPLETE DATASET

For classification with missing data, by introducing the upper bound c_l defined above, the objective function of the LS-SVMs can be modified into

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \sum_{l=1}^{d} c_l I_l^i + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{4.2}$$

which is mathematically equivalent to

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\vec{w}^2 + \frac{C}{2}\sum_{i=1}^{N}(\xi_i - \sum_{l=1}^{d} c_l I_l^i)^2 \\ \text{s.t} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{4.3}$$

where $\varphi(\vec{x}_i) = (\tilde{\varphi}(x_1^i), \tilde{\varphi}(x_2^i), \cdots, \tilde{\varphi}(x_d^i))$ and $\tilde{\varphi}(x_l^i)$ is a feature mapping such that the kernel function $\mathbf{K}$ can be adopted in Eq. (4.3). That is

$$\mathbf{K}(\vec{x}_i, \vec{x}_j) = \varphi(\vec{x}_i)^T \varphi(\vec{x}_j) = \sum_{l=1}^{d} k(x_l^i, x_l^j) \tag{4.4}$$

where

$$k(x_l^i, x_l^j) = \begin{cases} \tilde{k}(x_l^i, x_l^j) & \text{both } x_l^i \text{ and } x_l^j \text{ are not missing} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Gaussian kernel is adopted in this study, i.e., $\tilde{k}(x_l^i, x_l^j) = e^{\frac{-(x_l^i - x_l^j)^2}{\delta^2}}$, where δ is the kernel width. Obviously, $\mathbf{K}(\vec{x}_i, \vec{x}_j)$ in Eq. (4.4) is an additive Gaussian kernel, which can be calculated depending on whether each feature contains missing values or not.

It can be observed that after subtracting the classification error caused by missing features in the training dataset, the optimization problem in Eq. (4.3) is essentially to minimize the total classification error caused by the features without missing data. If there is no missing data in the training dataset, i.e., $I_l^i (i = 1, \cdots, N, l = 1, \cdots, d)$ is zero, Eq. (4.3) becomes the objective function of the standard LS-SVMs. The Lagrangian J of Eq. (4.3) gives the unconstrained minimization problem:

$$J = \tfrac{1}{2}\vec{w}^2 + \tfrac{C}{2}\textstyle\sum_{i=1}^{N}(\xi_i - \sum_{l=1}^{d} c_l I_l)^2 + \sum_{i=1}^{N} \alpha_i (y_i - \vec{w}^T \varphi(\vec{x}_i) - b - \xi_i) \tag{4.6}$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathbf{R}^N$ is the vector of all Lagrangian multipliers. The system of linear equations can be obtained

$$\sum_{j=1}^{N} \alpha_i \varphi(\vec{x}_j)^T \varphi(\vec{x}_i) + b + \frac{\alpha_i}{C} = y_i - \sum_{l=1}^{d} c_l I_l^i \tag{4.7}$$

Using the kernel trick, $\varphi(\vec{x}_j)^T\varphi(\vec{x}_i)$ is replaced by $\vec{K}(\vec{x}_i, \vec{x}_j)$, Eq. (4.7) can be further written in matrix form as

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\vec{\Lambda} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \vec{y} - \sum_{l=1}^{d} c_l \vec{I}_l \\ 0 \end{bmatrix} \tag{4.8}$$

where $\vec{y}$ is the actual label vector of all the training samples, i.e., $\vec{y} = (y_1, y_2, \cdots, y_N)$, $\vec{\Lambda}$ is a diagonal matrix with unity diagonal entries, and $\vec{I}_l = (I_l^1, I_l^2, \cdots, I_l^N)^T$.

Lastly, the model parameters can be calculated simply by using matrix inversion:

$$\begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \vec{y} - \sum_{l=1}^{d} c_l \vec{I}_l \\ 0 \end{bmatrix} \tag{4.9}$$

where $\mathbf{Q} = \mathbf{H}^{-1}$ and $\mathbf{H}$ is the first matrix from the left in Eq. (4.8). Once c_l is determined, $\vec{\alpha}$, $\vec{w}$ and b can be readily obtained. Therefore, we can easily obtain the decision function for the new sample $\vec{x}_t$ as below:

$$f(\vec{x}_t) = \begin{cases} \vec{w}^T\varphi(\vec{x}_t) + b & \text{if } \vec{x}_t \text{ has no missing value} \\ \vec{w}^T\varphi(\vec{x}_t) + b + \sum_{l=1}^{d} c_l I_l^t & \text{otherwise} \end{cases} \tag{4.10}$$

The proposed model can be used to solve multi-class classification problems by using the one-against-all strategy. Thus, the predicted output of the new sample $\vec{x}_t$ is determined by $\max_{k=1,\cdots,M} y_k(\vec{x}_t)$, where M denotes the number of the classes.

4.2.3 Fast Leave-one-out Cross Validation Strategy

4.2.3.1 Fast Leave-one-out Cross Validation for Parameter Tuning

The classification performance of the proposed model depends on the value of the parameter c_l. The fast leave-one-out cross validation strategy introduced in Chapter 3.2.5 is employed to determine the optimal value of c_l.

Similarly, by defining $\left[\vec{\alpha}'^T, b'\right]^T = \mathbf{Q}\left[\vec{y}^T, 0\right]^T$, $\left[\vec{\alpha}''^T, b''\right]^T = \mathbf{Q}\left[\vec{I}_l^T, 0\right]^T$, and $\vec{\alpha} = \vec{\alpha}' - \sum_{l=1}^{d} c_l \vec{\alpha}_l''$, the leave-one-out output $\tilde{y}_i$ of the i-th training sample can be represented as

$$\tilde{y}_i = y_i - \frac{\alpha_i'}{Q_{ii}} + \frac{\sum_{l=1}^{d} c_l \alpha_{li}''}{Q_{ii}} \tag{4.11}$$

The loss function below is adopted to avoid local minima issues:

$$l(\tilde{y}_i, y_i) = |1 - \tilde{y}_i y_i|_+ = \left| y_i \frac{\alpha_i' - \sum_{l=1}^{d} c_l \alpha_{li}''}{Q_{ii}} \right|_+, \tag{4.12}$$

where $|x|_+ = \max\{0, x\}$. Finally, the objective function becomes:

$$\begin{aligned} &\min_{c_l} \sum_{i=i}^{N} l(\tilde{y}_i, y_i) \\ &\text{s.t} \qquad ||\vec{c}||_2 \leq D \end{aligned} \tag{4.13}$$

where D is a constant and the L_2 norm constraint is added on the vector $\vec{c}$ as a form of regularization. This optimization process can be implemented by a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 4.

Algorithm 4: Projected Sub-gradient Descent Algorithm
Input: $\vec{\alpha}^{'}$, $\vec{\alpha}_l^{''}$
Initialize: $\vec{c} \leftarrow \vec{0}$ and $t \leftarrow 1$
Repeat
$\tilde{y_i} = y_i - \frac{\alpha_i^{'}}{\vec{Q}_{ii}} + \frac{\sum_{l=1}^{d} c_l \alpha_{li}^{''}}{\vec{Q}_{ii}}, i = 1, 2, ..., N$
$d_i \leftarrow \vec{1}\{\tilde{y_i} y_i > 0\}, i = 1, 2, ..., N$
$c_l \leftarrow c_l - \frac{1}{\sqrt{t}} d_i y_i \frac{\alpha_{li}^{''}}{Q_{ii}}, l = 1, 2, \cdots, d$
If $\|\|\vec{c}\|\|_2 > D$ then $\vec{c} \leftarrow \frac{\vec{c}}{\|\vec{c}\|_2} D$
End if
$t \leftarrow t + 1$
Until convergence
Output: $\vec{c}$

4.2.3.2 Interpretation of Influences of Missing Features

The parameter c_l represents the influence level on the classification error caused by the missing feature l, which consequently provides the guidance for the future data collection.

Given a multi-class classification task, after obtaining the corresponding value of c_k for M classes ($k = 1, 2, \cdots, M$), two cases are considered as below:

Case 1: If c_l^k for each class equals 0 or $\max\limits_{k=1,2,\cdots,M} |c_l^k|$ is less than a given small positive threshold, the upper bound of the influence of the l-th feature can be regarded as negligible.

Case 2: If $\min\limits_{k=1,2,\cdots,M} |c_l^k|$, denoted as *Inf,* is greater than 0 or a given small positive threshold, the l-th feature has a certain extent of influence on the classification performance. The greater the value of *Inf is,* the more significantly the influence on the classification performance are caused by this missing feature.

4.3 A Case Study on a Real World Community Health Care Dataset

4.3.1 Data Collection

The real-world dataset adopted in this study was collected from the PolyU-Henry G. Leong Mobile Integrative Health Centre (MIHC), which is a nurse-led mobile clinic in Hong Kong providing free health screening services for elderly people at the age of 60 years or above Choi et al. [2013]. The data were collected in August 2013 from two communities, which include demographics, socioeconomic status, health history and the outcomes of several health assessments of 444 clients. The participants were asked to complete a questionnaire about their QOL.

Some data were missing from the dataset, which were mainly caused by (i) language barriers due to incomprehension of dialectical differences, (ii) physical frailty, hearing or cognitive impairment, (iii) clients lacking patience to finish the health assessments, (iv) time conflicts, and (v) reluctance to disclose personal information due to privacy concerns. There are 33 features in total, in which 14 of them are without missing data. 14 of them are with missing data in 5% of all the samples. 4 of them are with missing data in 5% to 10% of all the samples. One feature is with missing data in more than half (60.1%) of all the samples. Therefore, in total, 19 of 33 features in the dataset contain missing values. Table 4.1 shows the extent of missing data for certain features. In the dataset, demographic data including gender, age and marital status; socioeconomic data including the type of residency, relationships with roommates and social participation; and health history data including smoking and drinking habits and chronic health conditions are available. Data obtained from a series of health

Table 4.1: THE EXTENT OF MISSING DATA IN CERTAIN FEATURES ($N = 444$)

Features	No. (%) of missing values	Mean±SD	No. (%) of patients
Age	0	75.30±7.87	
Gender	0		
Male			136(30.6)
Female			308(69.4)
Mobility	0		
Wheel chair			8 (1.8)
Walking stick			53(11.9)
Independent			382 (86.0)
Walking frame			1 (.2)
Social participation	0		
Unengaged			99 (22.3)
Partial unengaged			119 (26.8)
Engaged			226 (50.9)
Marital status	7 (1.6)		
Single			25 (5.6)
Married			250 (56.3)
Widowed			138 (31.1)
Separated/divorced			24 (5.4)
Residence	7 (1.6)		
Private housing			95 (21.4)
Public housing			197 (44.4)
Elderly home			2 (0.5)
Nursing home			7 (1.6)
Others			136 (30.6)
Smoking habit	7 (1.6)		
Smoker			24 (5.4)
Non-smoker			413 (93.0)
Drinking habit	7 (1.6)		
Drinker			52 (11.7)
Non-drinker			385 (86.7)
Hypertension	0		
Without hypertension			185 (41.7)
With hypertension			259 (58.3)
Number of co-morbidities	267 (60.1)	2.16±1.62	

assessments are also available and described in Table. 4.2.

Table 4.2: HEALTH RELATED ASSESSMENTS AND QUESTIONNAIRE ON MIHC

Title	Description
Bio-measurements	Major vital signs of the patients were measured, e.g. body temperature, pulse rate, oxygen saturation (SpO2), blood pressure and waist-hip ratio (WHR).
Berg Balance Scale (BBS)	Patient balance ability measured using a metric established using 14 tests. These tests include having the patient stand up from a sitting position and other more taxing balance tests e.g. standing on one foot. From this, a score between 0 to 4, with 4 being the highest, is assigned to each completed test. This gives an overall rating ranging from 0 to 56 for the patients overall balance ability.
Timed Up and Go Test (TUG)	A physical test to measure basic functional mobility, this test is well known and has good reliability. This test required that the patient would, starting from a sitting position, rise from a chair and walk for 3 meters, then turn around and return to the original sitting position. Patients were required to repeat the task three times to establish a best time.
Visual Analogue Scale (VAS) for pain	A scale used to measure the extent of pain to which a client felt localized at the most painful part of the body. Using a 10 cm vertical line, patients indicate visually somewhere from the lower end to upper end of this line the extent of their experienced pain: ranging from no pain to unbearable pain from top to bottom of the line respectively.
The 30-second Chair Stand Test (30-s CST)	Designed to test lower body strength and endurance, specifically when undertaking demanding tasks found in daily life. Intended to measure the patients ability to perform daily tasks such as climbing stairs, getting up from chairs or out of the bath tub. The test required that the patient repeatedly rise from a chair to a fully standing position and then sit down again. The number of times that the patient was able to perform this task within a 30 second window was recorded.
Body Composition Analysis	Used to determine patient levels of fitness and obesity using body mass index (BMI), skeletal muscle mass, body fat mass and body fat percentage (BFP).
Handgrip Strength	Using a dynamometer the grip strength of both the dominant and non-dominant hand were measured taking the average over three trials. This test required that the patient squeeze the device with maximum effort from a standing position. A total of six trials, three for each hand, was performed and the average strength for both were recorded.
QOL	The QOL of the clients was measured using the WHOQOL-BREF(HK).

Table 4.3: RE-CATEGORIZATION OF THE RESPONSES TO OVERALL QOL

Original score	Re-categorized score	QOL description
1,2	1	poor
3	2	neutral
4,5	3	good

4.3.2 Data pre-processing

The scores measured using the WHO quality of Life-BREF questionnaire (Hong Kong version) (WHOQOL-BREF (HK)) are re-categorized into three classes, namely, (i) '1' indicating poor QOL, (ii) '2' for neutral QOL and (iii) '3' for good QOL, by grouping participants choosing '1' and '2' in the original 5-point scale into the first class and those choosing '4' and '5' into the third class, as shown in Table 4.3.

4.3.3 Results Analysis

In the experiments, the novel additive LS-SVMs model is verified and evaluated on the real-world community dataset with missing data for predicting the elderly QOL. The performance is compared with those using methods (A) to (D), referring to (A) case deletion, (B) feature deletion, (C) mean imputation and (D) k-NN imputation respectively. For the proposed model, the missing data were handled simultaneously with the construction of the classification model. For methods (A) to (D), missing data are handled in the data pre-processing stage followed by the traditional SVMs for classification. The experimental results are shown in Table 4.4. The classification performance of method (A) is also compared with those using other methods via t-test. We find that the proposed method can achieve a better classification performance. Moreover, the influence level on the classification error caused by the missing feature can be represented using *Inf*. Table 4.5 lists the values of *Inf* with missing features in

Table 4.4: CLASSIFICATION ACCURACIES OF THE PROPOSED AND COMPARATIVE METHODS

Performances		Methods				
		proposed method	method (A)	method (B)	method (C)	method (D)
Accuracy	Mean	**0.7438**	0.7149	0.7187	0.6896	0.7052
	SD	0.0215	0.0441	0.0407	0.0163	0.0297
	Max.	0.7612	0.7447	0.7836	0.7164	0.7487
	Min.	0.7164	0.6783	0.6716	0.6643	0.6816
p-value		-	0.002	0.003	0.000	0.008

descending order.

From Table 4.4, we can observe that the proposed model achieved the best classification performance with the mean accuracy of 0.7438 and the maximum accuracy of 0.7612. The prediction models developed using methods (A), (B), (C) and (D) had comparatively lower performances. Their mean classification accuracies were 0.7149, 0.7187, 0.6896 and 0.7052 respectively. The performance of the proposed model was statistically better than those using other methods as evidence from the results of the t-tests. In general, in this real-world community health care application, the proposed additive LS-SVMs model outperformed the other methods that employ conventional treatments for handling missing data and SVMs for classification.

In terms of the running time, the proposed model with the fast leave-one-out cross validation strategy took around 60 seconds (on a computer equipped with a 3.4 GHz Intel Core i7-4930K processor and 16 GB RAM), which cannot be achieved if the missing data were handled using the standard leave-one-out cross-validation. Given d missing features in a training dataset of size N, suppose it takes t seconds to randomly assign a value from 0.0 to 1.0 at a step size of 0.001 to c_l $l = 1, 2, , d$, by using the standard leave-one-out cross-validation, the running time can be represented as $(1/0.001)^d \times N \times t = (1000)^d Nt$. We can observe that it is impractical when d is large. On the other hand, the proposed model only took around 60 seconds for classification

Table 4.5: INFLUENCES OF MISSING FEATURES

Missing features	***Inf***
Number of co-morbidity	0.5837
Duration of doing exercise (each time)	0.0859
Skeletal muscle mass	0.0585
Body fat mass	0.0585
BFP	0.0585
BMI	0.0311
Roommate	0.0173
Marital status	0.0173
Residence	0.0173
Smoking habit	0.0173
Drinking habit	0.0173
VAS for pain	0.0132
Day of doing exercise (per week)	0.0125
Blood glucose	0.0124
Body temperature	0.0124
Relation with roommate(s)	0.0111
Abbreviated mental test (AMT) score	0.0072
WHR	0.0000
30-s CST test	0.0000

with the community health care dataset although there were 19 missing features in it.

The proposed additive LS-SVMs model can also provide information about the influence on the classification performance caused by the missing feature. For the multi-class classification, if *Inf* of the missing feature is equal to zero or negligibly small, it can be inferred that this feature has little effect in the classification process. As shown in Table 4.5, features '30-s CST test' and 'WHR' are with *Inf* of zero. Therefore, they are not significant features for predicting QOL and the effect of missing data is minimal. On the other hand, the feature 'Number of co-morbidity' achieves the highest *Inf of* (0.5837), indicating that this feature has the high level of importance on the prediction performance and the effect of missing data was significant. Besides, the *Inf* of features associated with body composition analysis (e.g. skeletal muscle mass, body fat mass, BFP and BMI) were higher than those associated with socio-demographic characteristics (e.g. marital status, residence, roommate), which were in turn higher than those associated with health history (e.g. drinking habit and smoking habit) and health assessments (e.g. pain and AMT score). The results suggest that more attention should be paid to those features with high values of *Inf* so as to guarantee better data quality in the future data collection.

4.4 Summary

This chapter proposes a novel additive LS-SVMs model to tackle the classification with missing data in health care. Instead of handling missing data in the data pre-processing stage, the proposed model can directly perform the classification with missing data by simultaneously considering the influences on the classification error caused by missing features using the fast leave-one-out cross validation strategy. Moreover, the influence

levels of missing features can give the relative importance of those features and guide health professionals to further improve the data collection in future. The proposed model is evaluated on a real world community health care dataset for predicting the elderly QOL and compared with the traditional data imputation methods, namely, case deletion, features deletion, mean imputation and k-NN imputation, followed by the traditional SVMs. The experimental results show that the proposed model can achieve the good performance with missing data and provide insights into missing features.

Chapter 5

A Transfer-based Additive LS-SVMs Model for Predicting Elderly QOL with Missing Data

5.1 Introduction

In this chapter, a transfer-based additive LS-SVMs model is proposed from a transfer learning perspective to handle classification with missing data. The LS-SVMs framework is adopted in the source and target domains, where the source domain represents the complete portion of the dataset, while the target domain represents the whole dataset with missing data. The proposed model aims to leverage the model-based knowledge from the source domain to the target domain by finding a correlation between weight parameters of two domains within the LS-SVMs framework. The proposed model can simultaneously evaluate the influence on the classification error caused by each incomplete sample using a fast leave-one-out cross validation strategy,

which provides distinct information for data cleaning to guarantee data quality. For example, incomplete samples with higher error influences can be discovered and removed from the dataset.

The novel transfer-based additive LS-SVMs model is evaluated on the public UCI datasets with various combinations of missing data rates and missing features, and compared with traditional missing data treatment methods, including case deletion, mean imputation and k-NN imputation followed by the traditional LS-SVMs. The proposed model is also conducted on the community heath care dataset for predicting the elderly QOL, which particularly highlights the contributions and benefits of the proposed model to the real world application.

We must notice that although the proposed model in this chapter and the novel additive LS-SVMs model presented in Chapter 4 are both based on the LS-SVMs framework using the additive kernel for missing data classification, there are two main differences between them. First, the former model introduces the transfer learning mechanism to minimize the disagreement between the source and target domains on the incomplete data, while the later one does not. Second, the former model introduces the influence levels of missing features in the dataset, while the latter one introduces the influence levels of incomplete samples in the dataset. The obtained information from the influence values thus have different meanings for health professionals in real-world applications.

This chapter is organized as follows. Section 5.2 presents the proposed transfer-based additive LS-SVMs model in detail. Section 5.3 shows the experimental evaluations on the UCI public datasets. Section 5.4 shows a case study on a real world community health care dataset. Section 5.5 concludes the chapter.

5.2 Transfer-based Additive LS-SVMs Model with Missing data

The proposed transfer-based additive LS-SVMs model and its fast leave-one-out cross-validation strategy to estimate the influence levels on the classification error caused by incomplete samples are given below.

5.2.1 Problem Description

Given a training dataset $\mathbf{S}$ with N samples, the input dataset is denoted as $\mathbf{X}$, the corresponding output dataset is denoted as $\mathbf{Y}$, where $\mathbf{S} = \{(\vec{x}_1, y_1), ..., (\vec{x}_N, y_N)\}$, $\vec{x}_l = (x_1^l, x_2^l, ..., x_d^l) \in \mathbf{X} \subset \mathbf{R}^d$ and $y_l \in \mathbf{Y} = \{+1, -1\}$. Each sample $\vec{x}_i$ contains d features. $\mathbf{S}$ consists of two data portions of data ($N = N_1 + N_2$), N_1 includes the complete data samples ($\vec{x}_1, \vec{x}_2, ..., \vec{x}_{N_1}$) and N_2 includes the incomplete data samples ($\vec{x}_{N_1+1}$,$\vec{x}_{N_1+2}$,...,$\vec{x}_{N_1+N_2}$). We want to find a decision function $f : \mathbf{X} \rightarrow \mathbf{Y}$, such that it can find the matching y for any new incoming sample $\vec{x}$. Fig. 5.1 demonstrates the dataset $\mathbf{S}$ where missing data are denoted by the symbol ?.

5.2.2 Framework of the Proposed Model

The framework of the proposed transfer-based additive LS-SVMs model is illustrated in Fig. 5.2. The source domain contains complete data N_1, and the target domain contains both N_1 and incomplete data N_2. The additive LS-SVMs classifier is first constructed on the source domain and then a transfer-based additive LS-SVMs model is constructed for classification in the target domain with missing data.

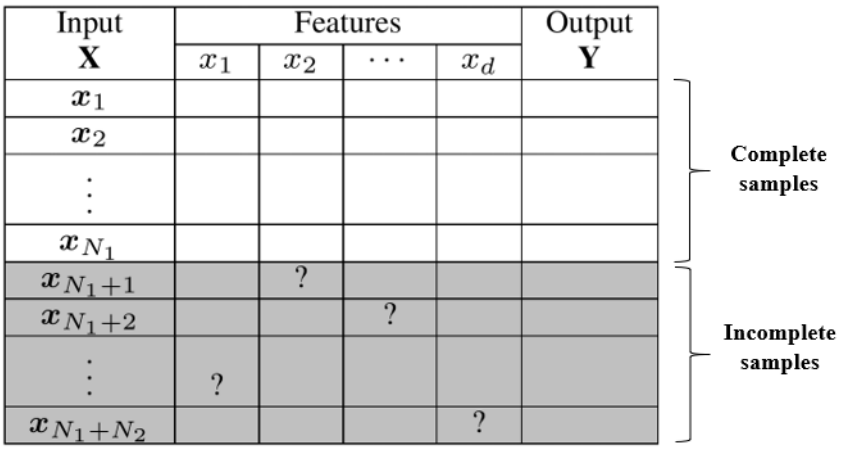


Figure 5.1: DATASET REPRESENTATION

5.2.3 Adaptive Regularization

In order to find the function $\mathcal{H}$ in the hypothesis space which approximates the unknown decision function f, the learning process can be formalized as an optimization problem to minimize the structural risk:

$$\eta\Omega(f) + R_{emp}(f(\vec{x_l}), y_l) \tag{5.1}$$

where $\eta > 0$ is a regularization parameter which balances good generalization performance with the smoothness or simplicity enforced by a small $\Omega(f)$. The empirical risk $R_{emp}(f)$ can be those using square loss or the -insensitive loss. To maximize the margin of classification in the feature space using the regularization term $\frac{1}{2}\|\vec{w}\|^2$, we get:

$$\frac{1}{2}\|\vec{w}\|^2 + R_{emp}(f(\vec{x_l}), y_l) \tag{5.2}$$

In the proposed model, the distribution P_s in the source domain and the distribution P_t in the target domain are related, and the model on each domain shares the similarity

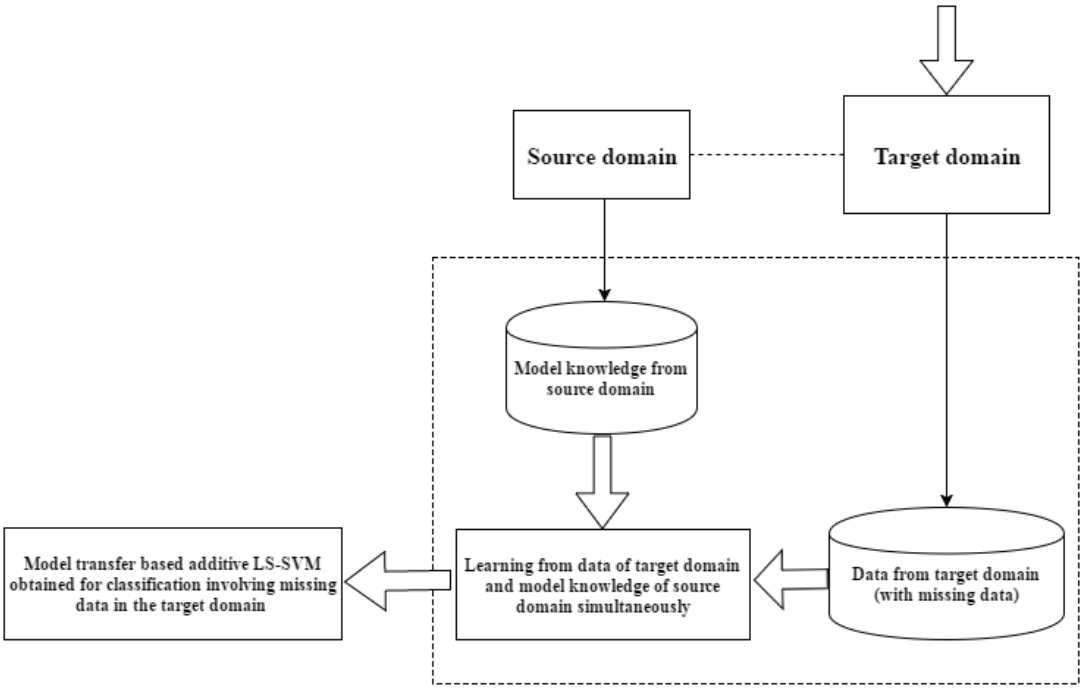


Figure 5.2: THE FRAMEWORK OF THE PROPOSED TRANSFER-BASED ADDITIVE LS-SVMS

to some extent. Thus, the model knowledge learned from the source domain can be leveraged to help the learning process in the target domain. For example, we can first find the optimal $\vec{w}_s$ by minimizing Eq. (5.2) in the source domain. When we encounter a new target domain, we can construct a model in which $\vec{w}_t$ gets as close as possible to the known $\vec{w}_s$. Through editing the regularization term, the optimization problem becomes to minimize

$$\frac{1}{2}\left\|\vec{w}_t - \vec{w}_s\right\|^2 + R_{emp}(f(\vec{x}_l), y_l) \tag{5.3}$$

where $f(\vec{x}_l)$ on the target domain is parameterized in terms of $\vec{w}_t$.

In addition, to evaluate the similarity between $\vec{w}_s$ and $\vec{w}_t$ in the optimization problem above, we can further edit the regularization term into $\|\vec{w}_t - \lambda\vec{w}_s\|$ by adding

the weighting factor λ.

5.2.4 Transfer-based Additive LS-SVMs Model

To construct the proposed transfer-based additive LS-SVMs model for missing data, we use $\lambda\vec{w}_s$ as reference in the regularization term in Eq. (5.3), and the square loss $R_{emp}(f(\vec{x}_l), y_l) = (f(\vec{x}_l) - y_l)^2$. Moreover, the upper bound of the classification error caused by each incomplete sample in the target domain is denoted as c_l. The learning parameters λ and c_l are selected by the fast leave-one-out cross validation strategy. Thus, the proposed model is obtained by reformulating the minimization problem of LS-SVMs as:

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}(\vec{w} - \lambda\vec{w}_s)^2 + \frac{C}{2}\sum_{l=1}^{N}(\xi_l - c_l)^2 \\ \text{s.t.} \quad & y_l = \sum_{j=1}^{d}\vec{w}_j^T\varphi(x_j^l)I_j^l + b + \xi_l \\ & l = 1, 2, ..., N_1 + N_2 (= N) \end{aligned} \tag{5.4}$$

where

$$I_j^l = \begin{cases} 1 & \text{if feature j of the } l\text{-th sample has value} \\ 0 & \text{if feature j of the } l\text{-th sample has no value} \end{cases}$$

Since $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_{N_1})$ is a group of the complete data, I_j^l $(l = 1, 2, ..., N_1)$ is set to 1, and c_l $(l = 1, 2, ..., N_1)$ is set to 0 accordingly. Also, $\varphi(\vec{x}_l) = (\tilde{\varphi}(x_1^l), \tilde{\varphi}(x_2^l), \cdots, \tilde{\varphi}(x_j^l), \cdots, \tilde{\varphi}(x_d^l))$ and $\tilde{\varphi}(x_j^l)$ is a feature mapping such that the kernel $\mathbf{K}$ below can be adopted in Eq. (5.4).

$$\mathbf{K}(\vec{x}_l, \vec{x}_k) = \varphi(\vec{x}_l)^T\varphi(\vec{x}_k) = \sum_{j=1}^{d} k(x_j^l, x_j^k) \tag{5.5}$$

where

$$k(x_j^l, x_j^k) = \begin{cases} \tilde{k}(x_j^l, x_j^k) & \text{both } x_j^l \text{ and } x_j^k \text{ are not missing} \\ 0 & \text{otherwise} \end{cases}$$

$\tilde{k}(x_j^l, x_j^k)$ is a kernel function. In this study, Gaussian function is adopted, i.e., $\tilde{k}(x_j^l, x_j^k) = e^{\frac{-(x_j^l - x_j^k)^2}{\sigma^2}}$, where σ is the kernel width. It is obvious that $\mathbf{K}(\vec{x}_l, \vec{x}_k)$ in Eq. (5.5) is an additive Gaussian kernel Duvenaud et al. [2011]. The Lagrangian L of Eq. (5.4) gives the unconstrained minimization problem:

$$L = \frac{1}{2}(\vec{w} - \lambda \vec{w}_s)^2 + \frac{C}{2}\sum_{l=1}^{N}(\xi_l - c_l)^2 + \sum_{l=1}^{N}\alpha_l(y_l - \sum_{j=1}^{d} w_j^T \varphi(x_j^l) I_j^l - b - \xi_l) \quad (5.6)$$

where $\vec{\alpha} \in \mathbf{R}^N$ is the vector of all Lagrangian multipliers. The system of linear equations can be obtained

$$\sum_{l=1}^{N}\sum_{j=1}^{d}\alpha_l I_j^k I_j^l \varphi(x_j^k)^T \varphi(x_j^l) + b + \vec{\alpha}_l/C = y_l - \lambda \sum_{j=1}^{d} w_j I_j^l \varphi(x_j^l) - c_l \quad (5.7)$$

Based on a kernel trick, we can replace $\varphi(\vec{x}_l)^T \varphi(\vec{x}_k)$ by $\mathbf{K}(\vec{x}_l, \vec{x}_k)$, Eq. (5.7) can be further written in matrix form as:

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\vec{\Lambda} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \vec{y} - \vec{\gamma} \\ 0 \end{bmatrix} \quad (5.8)$$

where $\vec{\Lambda}$ is a matrix in which each diagonal entry is one and all other entries are zero, $\vec{y}$ is the real label vector of all the samples in the training dataset and

$$\vec{\gamma} = \begin{pmatrix} \lambda \sum_{j=1}^{d} w_j^s I_j^1 \phi(x_j^1) \\ \lambda \sum_{j=1}^{d} w_j^s I_j^{N_1} \phi(x_j^{N_1}) \\ \lambda \sum_{j=1}^{d} w_j^s I_j^{N_1+1} \phi(x_j^{N_1+1}) + c_{N_1+1} \\ \lambda \sum_{j=1}^{d} w_j^s I_j^{N_1+2} \phi(x_j^{N_1+2}) + c_{N_1+2} \\ \vdots \\ \lambda \sum_{j=1}^{d} w_j^s I_j^{N} \phi(x_j^{N}) + c_{N} \end{pmatrix} = \lambda \begin{pmatrix} \sum_{j=1}^{d} w_j^s I_j^1 \phi(x_j^1) \\ \vdots \\ \sum_{j=1}^{d} w_j^s I_j^{N_1} \phi(x_j^{N_1}) \\ \sum_{j=1}^{d} w_j^s I_j^{N_1+1} \phi(x_j^{N_1+1}) \\ \sum_{j=1}^{d} w_j^s I_j^{N_1+2} \phi(x_j^{N_1+2}) \\ \vdots \\ \sum_{j=1}^{d} w_j^s I_j^{N} \phi(x_j^{N}) \end{pmatrix} + c_{N_1+1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + c_{N_1+2} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + c_N \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \tag{5.9}$$

Since $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_{N_1})$ is a group of the complete data, c_l $(l = 1, \cdots, N_1)$ should be $\vec{0}$. Thus, we do not represent them in the above formula. Our goal is to evaluate c_l $(l = N_1 + 1, \cdots, N)$ and λ using the proposed fast leave-one-out cross validation. Eq. (5.8) can be further rewritten into

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\vec{\Lambda} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \vec{y} - \lambda \vec{I}_1 - c_{N_1+1}\vec{I}_2 - c_{N_1+2}\vec{I}_3 - \cdots - c_N \vec{I}_{N_2+1} \\ 0 \end{bmatrix} = \begin{bmatrix} \vec{y} - \sum_{l=1}^{N_2+1} \beta_l \vec{I}_l \\ 0 \end{bmatrix} \tag{5.10}$$

where $\vec{\beta} = (\lambda, c_{N_1+1}, c_{N_1+2}, \cdots, c_N)$, and

$$\vec{I}_1 = \begin{pmatrix} \sum_{j=1}^{d} w_j^s I_j^1 \phi(x_j^1) \\ \sum_{j=1}^{d} w_j^s I_j^{N_1} \phi(x_j^{N_1}) \\ \sum_{j=1}^{d} w_j^s I_j^{N_1+1} \phi(x_j^{N_1+1}) \\ \sum_{j=1}^{d} w_j^s I_j^{N_1+2} \phi(x_j^{N_1+2}) \\ \vdots \\ \sum_{j=1}^{d} w_j^s I_j^{N} \phi(x_j^{N}) \end{pmatrix}, \ \vec{I}_2 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ \vec{I}_3 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ -1 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \ \vec{I}_{N_2+1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix}$$

Lastly, the model parameters can be calculated simply by using a matrix inversion:

$$\begin{bmatrix} \vec{\alpha} \\ b \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \vec{y} - \sum_{l=1}^{N_2+1} \beta_l \vec{I_l} \\ 0 \end{bmatrix} \tag{5.11}$$

where $\mathbf{Q} = \mathbf{H}^{-1}$ and $\mathbf{H}$ is the first matrix from the left in Eq. (5.10). $\vec{w}$ can be determined by

$$\vec{w} = \lambda \vec{w}_s + \sum_{l=1}^{N} \alpha_l (I_1^l \phi(x_1^l), I_2^l \phi(x_2^l), ..., I_d^l \phi(x_d^l)) \tag{5.12}$$

Therefore, the decision function for the new sample $\vec{x}_t$ is

$$\begin{aligned} f(\vec{x}_t) &= \sum_{j=1}^{d} \left(\lambda w_{sj} + \sum_{l=1}^{N} \alpha_l I_j^l \phi(x_j^l) \right) \phi(x_j^t) + b \\ &= \sum_{j=1}^{d} \left(\lambda w_{sj} \phi(x_j^t) + \sum_{l=1}^{N} \alpha_l I_j^l k(x_j^l, x_j^t) \right) + b \end{aligned} \tag{5.13}$$

The proposed model can be used to solve multi-class classification problems by using the one-against-all strategy. Thus, the predicted output of the new sample $\vec{x}_t$ is determined by $\max_{k=1,...,M} y_k(\vec{x}_t)$, where M denotes the number of the classes.

5.2.5 Fast Leave-one-out Cross Validation Strategy

5.2.5.1 Fast Leave-one-out Cross Validation for Parameter Tuning

The classification performance of the proposed model relies on the value of the parameter $\vec{\beta}$. The fast version of the leave-one-out cross-validation strategy introduced in Chapter 3.2.5 is employed to find the optimal value of $\vec{\beta}$.

Similarly, by defining $\left[\vec{\alpha}^{'T}, b^{'}\right]^T = \mathbf{Q}\left[\vec{y}^T, 0\right]$, $\left[\vec{\alpha}^{''T}, b^{''}\right] = \mathbf{Q}\left[\vec{I}_l^T, 0\right]$, and $\vec{\alpha} = \vec{\alpha}^{'} - \sum_{l=1}^{N_2+1} \beta \vec{\alpha_l}^{''}$, the predicted label of the i-th training sample can be represented as

$$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{\mathbf{Q}_{ii}} + \frac{\sum_{l=1}^{N_2+1} \beta_l \alpha_{li}^{''}}{\mathbf{Q}_{ii}} \tag{5.14}$$

The loss function below is adopted to avoid local minima issues:

$$l(\tilde{y}_i, y_i) = |1 - \tilde{y}_i y_i|_+ = \left| y_i \frac{\alpha_i^{'} - \sum_{l=1}^{N_2+1} \beta_l \alpha_{li}^{''}}{\mathbf{Q}_{ii}} \right|_+ \tag{5.15}$$

where $|x|_+ = \max\{0, x\}$. Finally, the objective function becomes:

$$\begin{aligned} &\min_{\beta_l} \sum_{i=i}^{N} l(\tilde{y}_i, y_i) \\ \text{s.t} \quad &\|\vec{\beta}\|_2 \leq D \end{aligned} \tag{5.16}$$

where D is a constant and the L_2 norm constraint is added on the vector $\vec{\beta}$. This optimization process can be implemented by a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 5.

5.2.5.2 Interpretation of Influences of Incomplete Samples

The parameter c_l provides the relative influence level on the classification error caused by of each incomplete sample, which consequently helps us to clean data and further improve the quality of data.

If $|c_l|$ or $\max_{k=1,\ldots,M} |c_l^k|$ of the l-th incomplete sample is greater than a given small positive threshold, the influence on the classification error from this incomplete sample is serious and should be cleaned from the dataset. Inversely, if $|c_l|$ or $\min_{k=1,\ldots,M} |c_l^k|$ of the

Algorithm 5: Projected Sub-gradient Descent Algorithm
Input: $\vec{\alpha}'$, $\vec{\alpha}_l''$
Initialize: $\vec{\beta} \leftarrow \vec{0}$ and $t \leftarrow 1$
Repeat
$\tilde{y}_i = y_i - \frac{\alpha_i'}{\vec{Q}_{ii}} + \frac{\sum_{l=1}^{N_2+1} \beta_l \alpha_{li}''}{\vec{Q}_{ii}}$, $i = 1, 2, ..., N$
$d_i \leftarrow \vec{1}\{\tilde{y}_i y_i > 0\}$, $i = 1, 2, ..., N$
$\beta_l \leftarrow \beta_l - \frac{1}{\sqrt{t}} \sum_{i=1}^{N} d_i y_i \frac{\alpha_{li}''}{Q_{ii}}$, $l = 1, 2, ..., N_2 + 1$
If $\|\vec{\beta}\|_2 > D$ then $\vec{\beta} \leftarrow \frac{\vec{\beta}}{\|\vec{\beta}\|_2} D$
End if
$\beta_1 \leftarrow max(\beta_1, 0)$
$t \leftarrow t + 1$
Until convergence
Output: $\vec{\beta}$

l-th incomplete sample is less than a given small positive threshold, the influence on the classification error caused by this incomplete sample is tolerable and this sample can be kept in the training dataset.

5.2.6 Computational Complexity

One highlight in the proposed model is its fast computational ability. Its computational cost contains three parts, which can be represented as $O(N_1^3 + N^3 + N(N_2 + 1))$. The first part includes the model knowledge obtained using LS-SVMs on the source domain N_1. Therefore, the complexity of this part is $O(N_1^3)$, which is the complexity of LS-SVMs. The second part includes the calculation of the matrix $\mathbf{Q}$ by the inverse related to the training dataset on the target domain, and so the corresponding computational complexity becomes $O(N^3)$. The third part includes the computational complexity of each iteration in the Algorithm 5 to optimize Eq. (5.16), which can be represented as $O((N_2 + 1)N)$.

Let us consider the traditional cross-validation strategy. If a standard LS-SVMs is

Table 5.1: DATASET DESCRIPTIONS

Dataset	Number of samples	Features	Class	Class(%)
Surgery	470	17	F T	85.11 14.89
Diabetic	1151	19	0 1	46.92 53.08
Pima	769	8	0 1	65.02 34.98
Bupa	345	6	1 2	42.03 57.97
Breast	699	9	2 4	65.52 34.48
Titanic	887	6	0 1	61.44 38.56
German	1000	24	1 2	70.00 30.00

adopted and T (≥ 3) grid values for each parameter are simply considered, the whole time complexity would become $O(N_1^3 + (N^3 * N)^{T(N_2+1)}) = O(N_1^3 + N^{4T(N_2+1)})$ which is much more computationally expensive, and even impractical, than $O(N_1^3 + N^3 + N(N_2 + 1))$ occupied by the proposed fast cross-validation strategy.

5.3 Experiments

5.3.1 UCI Datasets

In the experiments, seven public datasets (*Surgery, Diabetic, Pima, Bupa, Breast, Titanic* and *German*) were adopted. The original breast dataset has missing values, which were removed during data processing in order to fully control the missing data in our experiments. The rest of datasets are complete with no missing data. Table 5.1 summarizes the datasets adopted in this work.

5.3.2 Experimental Design

The main purpose of the experiments conducted in this work is to evaluate the performance of the proposed transfer-based additive LS-SVMs model for missing data, compared to traditional missing data classification methods, denoted as follows:

(A) **Case deletion** all samples with missing values were removed.

(B) **Mean imputation** missing values for a certain feature were replaced with the mean of values of complete samples for that feature.

(C) k**-NN imputation** missing values were replaced with the weighted mean of the k nearest-neighbour columns.

Using the proposed method, missing data was assembled by constructing a classifier. Using the comparative methods (A), (B) and (C), missing data were first manipulated, and then both standard LS-SVMs and SVMs classifiers were used on the processed data for model construction. To make the comparison fair, we adopted the additive Gaussian kernel on both proposed and comparative methods. We first calculated the standard deviation of each feature in the dataset and then took their average value as $\overline{\sigma}$. Accordingly, we established a trade-off parameter C and a Gaussian kernel parameter σ by searching $C \in \{1, 10, 50, 100, 1000, 10000\}$ and $\sigma \in \{\overline{\sigma}/16, \overline{\sigma}/8, \overline{\sigma}/4, \overline{\sigma}/2, \overline{\sigma}, 2\overline{\sigma}, 4\overline{\sigma}, 8\overline{\sigma}, 16\overline{\sigma}\}$. Additionally, we obtained $\vec{w}_s$ from the source domain for the proposed model transfer method in advance. Finding an optimal value for the neighbouring parameter k for the method (C) was a major issue. The missing values were filled using estimated values from their 1, 3, 7, 9 and 10 nearest neighbours. Due to the space limitations, we only show results from the 3 and 10 nearest neighbours, identified as k-NN3 and k-NN10 respectively in this work. All the experiments were implemented using 64-bit MATLAB on a computer with an Intel

Core i5-6300 2.40 GHz CPU and 8.00GB RAM.

Missing data were artificially inserted in different features with different proportions into the public datasets. We first selected the first, second and third most relevant feature(s) using wrapper and filter techniques, then modified their values to unknown. Doing this allowed us to consider that less relevant or non relevant features might not contribute to classifier construction or even compromise the experimental analysis. We also inserted various proportions of missing data in the datasets (10%, 20%, 30%, 40%, 50%, 60%) such that we could analyse the corresponding performance of the classifiers.

5.3.3 Experimental Results Analysis

The 10-fold cross validation strategy was used in the experiments for performance evaluation, to ensure that every sample from the dataset had a chance to be used in the training and testing sets. Here, the dataset was randomly divided into ten subsets. The model was built using nine subsets and tested on the remaining one. This process was repeated 10 times, and the mean and standard deviation of accuracy in the 10-fold cross validation procedure was calculated.

Tables 5.2-5.8 display the numerical experimental results of the proposed and comparative methods on seven public datasets in terms of accuracy. Figure 5.5 use line graphs to further demonstrate the change tendencies of performances with different missing data rates. In order to detect significant differences among the performances of the proposed and comparative methods, we also carried out the Friedman ranking test followed by Holm post-hoc test Demšar [2006]; Garcia and Herrera [2008] for multiple comparisons on seven datasets. The Friedman ranking test was used to

evaluate whether there was a statistically significant difference among all the methods. If the p-value is smaller than 0.05, the null hypothesis is rejected and there is significant difference. The Holm post-hoc test was used to further verify if there was a statistical difference between the best Friedman ranking method and each of the rest, and the hypothesis of equivalence of the methods is rejected if $p < \alpha/i$. Tables 5.11 and 5.12 list the corresponding statistical results about Friedman ranking test and Holm post-hoc test, respectively. According to these results, we make the following observations:

(1) In most cases, our proposed classifier achieved better classification performances than those using other comparative methods. This indicates that our proposed classifier, by leveraging the knowledge learned from the model on the source domain to the target domain, has the ability to perform classification with missing data and achieve advantageous performances compared with the traditional missing data treatments followed by LS-SVMs or SVMs.

(2) In very few cases, with a specific combination of the missing data rate and missing feature(s), the performance results of our proposed method were lower than those using the case deletion method. For example, in Table 5.6, when there were 40% missing data in the *Breast* dataset, (case deletion + SVMs) achieved marginally higher accuracies than the other methods. The similar situation occurs in Table 5.3, when there were 20% missing data in the *Diabetic* dataset. This might be due to the reason that those randomly selected missing data coincidently had the noise and thus data removal enhanced the classification performance, particularly of the SVM which suffers from the noise sensitivity problem. Also, there are few cases in Tables 5.2 and 5.5 that the proposed method was beaten by (k-NN3+SVMs) and (k-NN10+SVMs). We noticed that these usually occurred when the missing data rate was comparatively higher ($\geq 30\%$), which may greatly fluctuate the classification performance. In terms

of Tables 5.11 and 5.12, there are significant differences between the proposed method and all the comparative methods except (case deletion + SVMs) ($0.171857 > 0.05$) in terms of accuracy, we must notice that the proposed classifier also has the advantage on data cleaning via the fast leave-one-out cross validation strategy, which case deletion and all other imputation methods cannot achieve. Further details are discussed in the case study of the real world dataset.

5.4 A Case Study on a Real World Community Health Care Dataset

5.4.1 Data Collection and Pre-processing

In this case study, the proposed model was evaluated on the same community health care dataset introduced in Chapter 4.3.1 for predicting the elderly QOL. The range of QOL scores recorded under the World Health Organization questionnaire on quality of life: short form Hong Kong version (WHOQOL-BREF(HK)) framework Leung et al. [1997, 2005] lacked extreme values for an overall quality of life score on a 1 to 5 scale. Therefore, some data pre-processing was required. To avoid unintended bias in the training set, these values were re-mapped to a scale of 3, where "1" indicates poor, "2" indicates neutral, and "3" indicates good quality of life.

5.4.2 Challenge

Using 33 features inherent in the community health care dataset, we intended to construct a classifier to predict the quality of life of elderly patients using the same scale mentioned above - poor, neutral, and good. However, in this dataset 19 of the 33

features, and 159 of the 444 patient records, contain missing values, which presents problems for constructing a prediction model.

5.4.3 Results Analysis

Table 5.9 and Fig. 5.3 demonstrate that the proposed model provided the best classification performance. The mean accuracy of the proposed method was 0.7258 among all the methods. The running time of the proposed method which had the fast leave-one-out cross validation was 4.03 seconds. Thus, in this practical application, the proposed transfer-based additive LS-SVM model outperforms both conventional methods and the standard LS-SVMs classifier for missing data classification.

Additionally, the influence of each incomplete sample in the training dataset can be determined by $|c_l|$ (binary classification) or $\max_{k=1,2,3} |c_l^k|$ (the multi-class classification) obtained during the classification process. We performed data cleaning on the community health care dataset and observed the corresponding classification results on the cleaned dataset. Fig. 5.4 shows $\max_{k=1,2,3} |c_l^k|$ of each incomplete sample in the community health care training dataset. We can observe that the $\max_{k=1,2,3} |c_l^k|$ ranged from 0 to 1.8499. In fact, the $\max_{k=1,2,3} |c_l^k|$ of all the samples were below 1 except one (1.8499), which indicated that this incomplete sample had a comparatively big influence on the classification error and must be removed. Based on the range of these values, the threshold was set to 0.6, 0.65, 0.80, 1.00. The l-th incomplete sample whose $\max_{k=1,2,3} |c_l^k|$ was bigger than the chosen threshold was then removed and the corresponding performance results were displayed in Table. 5.10. We observe that the performance after data cleaning was maintained or improved, to a certain extent, by given different thresholds. The best classification accuracy achieved was 0.7327

when the incomplete samples with $\max_{k=1,2,3} |c_l^k|$ greater than 0.8 were removed. This result shows that the proposed classifier has the ability to clean unnecessary incomplete samples in the real-world dataset based on their influences on the classification error.

5.4.4 Contribution

Due to the complex nature of the way the mobile integrative health centre (MIHC) acquires data, any automated predictive algorithm that could decrease the workload for staff nurses would be valuable. More importantly, given the patient type, it is highly likely that future datasets from the MIHC will contain missing data and any over-collection of data will only increase the likelihood of missing features. As previously mentioned, a common reason for missing features is a loss of patience by the patient or an inability to communicate. Using the proposed method, it is possible to perform classification directly on the missing data. Moreover, unnecessary samples are automatically removed by determining which samples have the least impact on the overall accuracy of the classification when they are missing from the dataset. Beyond assisting with data cleaning, determining a classifier that effectively handles the corrupt or missing data samples, may vastly improve the overall performance and effectiveness of the MIHC itself. If patients and practitioners are less concerned about fully complying with the rigors of the tests, it is likely that stress levels and therefore test times will decrease. As a result, interpersonal relationships improve, leading to increased participation and better overall accuracy, and the cycle perpetuates.

Table 5.2: PERFORMANCE RESULTS FOR THE *Surgery* DATASET

Missing rates	Missing feature - 1								
	proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.8582±0.0123**	0.8402±0.0356	0.8466±0.0270	0.8440±0.0123	0.8482±0.0146	0.8511±0.0208	0.8489±0.0174	0.8475±0.0250	0.8511±0.0177
20%	**0.8573±0.0217**	0.8378±0.0262	0.8374±0.0275	0.8434±0.0178	0.8426±0.0279	0.8434±0.0227	0.8433±0.0273	0.8463±0.0178	0.8539±0.0207
30%	0.8534±0.0291	0.8267±0.0298	0.8387±0.0343	0.8424±0.0139	0.8411±0.0174	0.8427±0.0244	**0.8539±0.0201**	0.8428±0.0071	0.8411±0.0253
40%	**0.8485±0.0108**	0.8282±0.0350	0.8373±0.0316	0.8427±0.0113	0.8455±0.0274	0.8440±0.0165	0.8376±0.0355	0.8400±0.0256	0.8439±0.0198
50%	0.8465±0.0261	0.8330±0.0351	0.8386±0.0390	0.8345±0.0225	0.8433±0.0218	0.8369±0.0188	0.8389±0.0272	0.8417±0.0108	**0.8467±0.0250**
60%	**0.8463±0.0178**	0.8388±0.0365	0.8377±0.0389	0.8364±0.0108	0.8355±0.0208	0.8407±0.0148	0.8404±0.0239	0.8418±0.0183	0.8454±0.0081
	Missing features - 1, 10								
10%	**0.8581±0.0191**	0.8365±0.0267	0.8457±0.0324	0.8434±0.0108	0.8440±0.0198	0.8463±0.0148	0.8553±0.0146	0.8322±0.0249	0.8546±0.0244
20%	**0.8582±0.0188**	0.8514±0.0353	0.8566±0.0250	0.8471±0.0259	0.8511±0.0278	0.8440±0.0225	0.8461±0.0307	0.8374±0.0227	0.8504±0.0223
30%	**0.8487±0.0148**	0.8364±0.0222	0.8374±0.0355	0.8418±0.0205	0.8454±0.0281	0.8369±0.0309	0.8447±0.0244	0.8274±0.0228	0.8482±0.0280
40%	**0.8463±0.0178**	0.8365±0.0311	0.8293±0.0427	0.8392±0.0235	0.8440±0.0229	0.8369±0.0188	0.8411±0.0215	0.8317±0.0148	0.8461±0.0232
50%	**0.8440±0.0142**	0.8352±0.0326	0.8349±0.0414	0.8363±0.0205	0.8404±0.0227	0.8360±0.0071	0.8405±0.0230	0.8298±0.0213	0.8433±0.0178
60%	**0.8436±0.0041**	0.8332±0.0377	0.8326±0.0345	0.8358±0.0269	0.8324±0.0227	0.8329±0.0219	0.8433±0.0165	0.8334±0.0164	0.8424±0.0370
	Missing features -1, 10, 11								
10%	**0.8576±0.0148**	0.8433±0.0249	0.8460±0.0272	0.8406±0.0217	0.8440±0.0227	0.8369±0.0142	0.8454±0.0200	0.8440±0.0188	0.8475±0.0246
20%	**0.8572±0.0256**	0.8538±0.0306	0.8496±0.0262	0.8347±0.0108	0.8482±0.0328	0.8274±0.0148	0.8418±0.0223	0.8369±0.0156	0.8433±0.0251
30%	**0.8481±0.0164**	0.8255±0.0364	0.8267±0.0318	0.8298±0.0071	0.8468±0.0161	0.8340±0.0195	0.8433±0.0253	0.8311±0.0275	0.8426±0.0402
40%	**0.8445±0.0188**	0.8299±0.0317	0.8400±0.0359	0.8180±0.0205	0.8389±0.0253	0.8337±0.0164	0.8440±0.0253	0.8394±0.0209	0.8400±0.0252
50%	**0.8500±0.0082**	0.8365±0.0370	0.8366±0.0345	0.8252±0.0123	0.8369±0.0284	0.8302±0.0157	0.8411±0.0255	0.8363±0.0148	0.8496±0.0271
60%	0.8419±0.0206	0.8329±0.0358	0.8312±0.0369	0.8234±0.0228	0.8354±0.0279	0.8323±0.0228	0.8355±0.0161	0.8382±0.0175	**0.8428±0.0227**

Table 5.3: PERFORMANCE RESULTS FOR THE *Diabetic* DATASET

Missing rates	Missing feature - 2								
	proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.7346±0.0061**	0.7325±0.0148	0.7330±0.0168	0.7119±0.0109	0.7188±0.0138	0.7133±0.0207	0.7194±0.0192	0.7235±0.0208	0.7254±0.0245
20%	**0.7298±0.0200**	0.7250±0.0314	0.7278±0.0287	0.7091±0.0234	0.7100±0.0226	0.7158±0.0174	0.7246±0.0228	0.7225±0.0145	0.7289±0.0148
30%	0.7268±0.0204	0.7194±0.0316	**0.7273±0.0335**	0.7074±0.0188	0.7048±0.0272	0.7093±0.0076	0.7159±0.0094	0.7126±0.0173	0.7179±0.0184
40%	**0.7241±0.0191**	0.7206±0.0293	0.7221±0.0199	0.7033±0.0017	0.7039±0.0304	0.6950±0.0152	0.7052±0.0288	0.7125±0.0252	0.7130±0.0121
50%	**0.7182±0.0229**	0.7102±0.0432	0.7165±0.0270	0.7000±0.0179	0.7025±0.0281	0.7008±0.0155	0.7110±0.0188	0.7062±0.0178	0.7142±0.0182
60%	**0.7170±0.0282**	0.7029±0.0302	0.7035±0.0298	0.6917±0.0093	0.6936±0.0225	0.7009±0.0261	0.7048±0.0236	0.7058±0.0188	0.7049±0.0229
	Missing features - 2, 3								
10%	**0.7338±0.0116**	0.7317±0.0189	0.7305±0.0219	0.7007±0.0106	0.7040±0.0116	0.7208±0.0150	0.7220±0.0211	0.7110±0.0161	0.7208±0.0255
20%	**0.7244±0.0207**	0.7237±0.0236	0.7216±0.0274	0.7033±0.0017	0.7083±0.0081	0.7119±0.0261	0.7127±0.0153	0.7106±0.0093	0.7142±0.0342
30%	0.7257±0.0020	0.7236±0.0283	**0.7281±0.0117**	0.7023±0.0017	0.7090±0.0251	0.7108±0.0226	0.7032±0.0171	0.7100±0.0184	0.7055±0.0192
40%	**0.7263±0.0225**	0.7216±0.0307	0.7218±0.0367	0.6715±0.0188	0.6969±0.0181	0.6985±0.0145	0.7055±0.0232	0.7007±0.0207	0.7032±0.0182
50%	**0.7232±0.0245**	0.7185±0.0208	0.7220±0.0321	0.6888±0.0271	0.6899±0.0231	0.6878±0.0224	0.6982±0.0234	0.6991±0.0149	0.6910±0.0186
60%	**0.7159±0.0261**	0.6964±0.0301	0.6949±0.0306	0.6869±0.0204	0.6879±0.0244	0.6875±0.0174	0.6827±0.0187	0.6840±0.0206	0.6884±0.0228
	Missing features - 2, 3, 9								
10%	0.7290±0.0184	0.7302±0.0194	**0.7330±0.0210**	0.7110±0.0104	0.7124±0.0156	0.7129±0.0209	0.7243±0.0290	0.7301±0.0185	0.7306±0.0229
20%	**0.7254±0.0204**	0.7234±0.0269	0.7112±0.0282	0.7105±0.0167	0.7090±0.0160	0.7119±0.0104	0.7150±0.0194	0.7113±0.0120	0.7208±0.0155
30%	**0.7211±0.0225**	0.7120±0.0284	0.7190±0.0256	0.6982±0.0213	0.6990±0.0139	0.7012±0.0060	0.7052±0.0170	0.7107±0.0253	0.7116±0.0271
40%	**0.7206±0.0225**	0.7058±0.0239	0.7179±0.0261	0.6900±0.0186	0.6935±0.0121	0.6965±0.0100	0.6997±0.0179	0.7004±0.0159	0.7000±0.0211
50%	**0.7199±0.0225**	0.7150±0.0419	0.7156±0.0248	0.6843±0.0177	0.6870±0.0246	0.6917±0.0217	0.6879±0.0236	0.7018±0.0058	0.7068±0.0276
60%	**0.7038±0.0016**	0.6891±0.0349	0.6906±0.0425	0.6840±0.0250	0.6824±0.0278	0.6850±0.0192	0.6827±0.0192	0.6802±0.0290	0.6856±0.0234

Table 5.4: PERFORMANCE RESULTS FOR THE *Pima* DATASET

Missing rates	Missing feature - 2								
	proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.7706±0.0087**	0.7628±0.0202	0.7635±0.0236	0.7489±0.0115	0.7515±0.0178	0.7549±0.0043	0.7572±0.0202	0.7570±0.0066	0.7580±0.0253
20%	**0.7576±0.0217**	0.7557±0.0269	0.7351±0.0271	0.7388±0.0109	0.7342±0.0328	0.7401±0.0132	0.7411±.0.0233	0.7446±0.0130	0.7472±0.0298
30%	**0.7588±0.0263**	0.7517±0.0304	0.7401±0.0349	0.7330±0.0238	0.7203±0.0248	0.7334±0.0174	0.7424±0.0218	0.7172±0.0100	0.7307±0.0230
40%	**0.7505±0.0254**	0.7261±0.0424	0.7355±0.0203	0.7304±0.0180	0.7325±0.0236	0.7204±0.0175	0.7212±0.0287	0.7274±0.0107	0.7238±0.0324
50%	**0.7330±0.0136**	0.7271±0.0388	0.7267±0.0187	0.7215±0.0222	0.7134±0.0266	0.7158±0.0254	0.7104±0.0333	0.7206±0.0075	0.7225±0.0253
60%	**0.7317±0.0152**	0.7201±0.0404	0.7183±0.0504	0.7108±0.0250	0.7134±0.0171	0.7085±0.0025	0.7113±0.0302	0.7117±0.0109	07229±0.0221
	Missing features - 2, 6								
10%	**0.7763±0.0090**	0.7587±0.0201	0.7601±0.0246	0.7532±0.0197	0.7554±0.0172	0.7504±0.0205	0.7528±0.0307	0.7556±0.0212	0.7563±0.0211
20%	**0.7568±0.0225**	0.7508±0.0333	0.7427±0.0306	0.7345±0.0246	0.7433±0.0247	0.7317±0.0288	0.7346±0.0250	0.7409±0.0109	0.7416±0.0181
30%	0.7358±0.0139	**0.7512±0.0249**	0.7259±0.0249	0.7217±0.0215	0.7212±0.0259	0.7140±0.0156	0.7169±0.0146	0.7174±0.0152	0.7234±0.0228
40%	**0.7388±0.0132**	0.7254±0.0415	0.7239±0.0308	0.7201±0.0214	0.7199±0.0190	0.7039±0.0152	0.7056±0.0221	0.7003±0.0075	0.7013±0.0186
50%	**0.7272±0.0207**	0.7191±0.0567	0.7224±0.0526	0.7131±0.0238	0.7160±0.0216	0.6869±0.0200	0.6870±0.0198	0.6987±0.0163	0.6900±0.0391
60%	**0.7172±0.0100**	0.7166±0.0373	0.7169±0.0283	0.7100±0.0170	0.7122±0.0250	0.6797±0.0212	0.6732±0.0258	0.6849±0.0257	0.6857±0.0362
	Missing features -1, 2, 6								
10%	**0.7619±0.0189**	0.7562±0.0349	0.7553±0.0249	0.7448±0.0195	0.7459±0.0277	0.7480±0.0109	0.7403±0.0203	0.7509±0.0198	0.7515±0.0216
20%	**0.7518±0.0218**	0.7492±0.0254	0.7462±0.0269	0.7330±0.0200	0.7310±0.0332	0.7305±0.0307	0.7307±0.0277	0.7317±0.0090	0.7372±0.0194
30%	0.7359±0.0229	**0.7436±0.0293**	0.7243±0.0276	0.7165±0.0109	0.7188±0.0264	0.7042±0.0205	0.7147±0.0209	0.7159±0.0154	0.7117±0.0366
40%	**0.7201±0.0218**	0.7114±0.0326	0.7169±0.0463	0.6948±0.0109	0.7088±0.0208	0.7043±0.0198	0.7027±0.0322	0.7002±0.0149	0.6905±0.0184
50%	0.7148±0.0222	**0.7171±0.0469**	0.7141±0.0422	0.6984±0.0238	0.7030±0.0247	0.6812±0.0152	0.6827±0.0280	0.6827±0.0189	0.6861±0.0260
60%	**0.7128±0.0107**	0.7106±0.0359	0.7104±0.0258	0.7082±0.0259	0.7065±0.0343	0.6782±0.0090	0.6727±0.0272	0.6714±0.0090	0.6814±0.0243

Table 5.5: PERFORMANCE RESULTS FOR THE *Bupa* DATASET

Missing rates	Missing feature - 5								
	proposed method	case deletion		mean imputation		k-NN10			
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.7212±0.0333**	0.6876±0.0464	0.6903±0.0220	0.6810±0.0314	0.6888±0.0449	0.7040±0.0468	0.6837±0.0426	0.6865±0.0329	0.6865±0.0529
20%	**0.7147±0.0350**	0.6948±0.0403	0.6964±0.0275	0.6790±0.0114	0.6700±0.0351	0.6865±0.0389	0.6654±0.0506	0.6869±0.0427	0.6721±0.0312
30%	**0.6955±0.0347**	0.6923±0.0411	0.6932±0.0224	0.6771±0.0282	0.6772±0.0225	0.6881±0.0399	0.6779±0.0452	0.6831±0.0394	0.6856±0.0385
40%	**0.6945±0.0317**	0.6831±0.0383	0.6857±0.0305	0.6750±0.0361	0.6763±0.0284	0.6775±0.0369	0.6625±0.0426	0.6713±0.0434	0.6712±0.0518
50%	**0.6763±0.0242**	0.6581±0.0390	0.6654±0.0383	0.6575±0.0378	0.6538±0.0319	0.6665±0.0369	0.6683±0.0590	0.6735±0.0422	0.6692±0.0517
60%	0.6744±0.0296	0.6590±0.0476	0.6667±0.0376	0.6644±0.0402	0.6692±0.0344	0.6721±0.0403	**0.6788±0.0294**	0.6705±0.0167	0.6865±0.0382
	Missing features - 3, 5								
10%	**0.7051±0.0242**	0.6846±0.0430	0.6860±0.0360	0.6837±0.0426	0.6885±0.0260	0.6840±0.0374	0.6837±0.0309	0.6935±0.0356	0.6894±0.0531
20%	**0.7019±0.0192**	0.6928±0.0426	0.6912±0.0238	0.6856±0.0428	0.6692±0.0251	0.6598±0.0350	0.6596±0.0418	0.6810±0.0428	0.6673±0.0495
30%	**0.7008±0.0304**	0.6712±0.0467	0.6732±0.0209	0.6775±0.0411	0.6785±0.0161	0.6608±0.0386	0.6692±0.0467	0.6810±0.0421	0.6808±0.0477
40%	**0.6795±0.0294**	0.6720±0.0447	0.6762±0.0181	0.6588±0.0357	0.6558±0.0275	0.6308±0.0432	0.6471±0.0408	0.6515±0.0343	0.6269±0.0486
50%	**0.6787±0.0211**	0.6474±0.0111	0.6485±0.0359	0.6683±0.0357	0.6692±0.0439	0.6383±0.0399	0.6260±0.0383	0.6179±0.0409	0.6087±0.0484
60%	**0.6674±0.0344**	0.6429±0.0391	0.6419±0.0281	0.6606±0.0444	0.6619±0.0251	0.6337±0.0446	0.6375±0.0453	0.6275±0.0473	0.6154±0.0260
	Missing features -3, 5, 6								
10%	**0.6963±0.0284**	0.6874±0.0393	0.6886±0.0258	0.6895±0.0309	0.6923±0.0251	0.6917±0.0353	0.6856±0.0253	0.6927±0.0438	0.6837±0.0404
20%	**0.6951±0.0156**	0.6918±0.0170	0.6929±0.0252	0.6867±0.0462	0.6831±0.0331	0.6621±0.0323	0.6712±0.0384	0.6740±0.0311	0.6750±0.0506
30%	0.6763±0.0200	**0.6859±0.0419**	0.6713±0.0288	0.6740±0.0416	0.6731±0.0240	0.6565±0.0358	0.6653±0.0456	0.6742±0.0361	0.6654±0.0455
40%	**0.6699±0.0334**	0.6502±0.0330	0.6635±0.0379	0.6468±0.0452	0.6454±0.0225	0.6285±0.0358	0.6308±0.0398	0.6338±0.0358	0.6404±0.0228
50%	**0.6663±0.0338**	0.6478±0.0294	0.6400±0.0229	0.6611±0.0415	0.6446±0.0228	0.6163±0.0412	0.6173±0.0381	0.6169±0.0387	0.6154±0.0377
60%	**0.6699±0.0242**	0.6429±0.0238	0.6433±0.0219	0.6450±0.0332	0.6492±0.0233	0.6090±0.0056	0.6115±0.0645	0.5942±0.0367	0.6087±0.0301

Table 5.6: PERFORMANCE RESULTS FOR THE *Breast* DATASET

Missing rates	Missing feature - 2								
	proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.9756±0.0109**	0.9741±0.0117	0.9744±0.0104	0.9711±0.0077	0.9678±0.0056	0.9702±0.0103	0.9712±0.0087	0.9699±0.0101	0.9702±0.0084
20%	**0.9707±0.0049**	0.9678±0.0114	0.9637±0.0111	0.9699±0.0085	0.9620±0.0011	0.9696±0.0117	0.9634±0.0098	0.9674±0.0113	0.9654±0.0123
30%	**0.9703±0.0098**	0.9658±0.0106	0.9611±0.0167	0.9687±0.0104	0.9698±0.0094	0.9688±0.0102	0.9639±0.0126	0.9698±0.0101	0.9620±0.0085
40%	0.9707±0.0049	0.9740±0.0092	**0.9755±0.0093**	0.9701±0.0090	0.9727±0.0117	0.9706±0.0115	0.9605±00049	0.9700±0.0104	0.9707±0.0098
50%	**0.9711±0.0075**	0.9705±0.0139	0.9689±0.0081	0.9704±0.0114	0.9707±0.0114	0.9690±0.0094	0.9683±0.0098	0.9694±0.0118	0.9668±0.0136
60%	**0.9690±0.0076**	0.9563±0.0122	0.9634±0.0149	0.9673±0.0100	0.9668±0.0111	0.9683±0.0111	0.9678±0.0132	0.9687±0.0093	0.9680±0.0306
	Missing features - 2, 6								
10%	**0.9750±0.0056**	0.9732±0.0117	0.9719±0.0140	0.9687±0.0097	0.9680±0.0164	0.9686±0.0106	0.9688±0.0158	0.9686±0.0089	0.9659±0.0126
20%	**0.9724±0.0065**	0.9674±0.0104	0.9610±0.0111	0.9683±0.0093	0.9649±0.0106	0.9668±0.0108	0.9693±0.0069	0.9654±0.0085	0.9665pm0.0067
30%	**0.9701±0.0049**	0.9643±0.0117	0.9608±0.0103	0.9683±0.0100	0.9688±0.0089	0.9684±0.0092	0.9663±0.0101	0.9686±0.0088	0.9605±0.0074
40%	0.9691±0.0028	0.9650±0.0111	**0.9706±0.0136**	0.9671±0.0111	0.9629±0.0112	0.9650±0.0108	0.9644±0.0138	0.9644±0.0104	0.9571±0.0136
50%	**0.9707±0.0123**	0.9659±0.0106	0.9689±0.0221	0.9691±0.0100	0.9659±0.0069	0.9683±0.0106	0.9673±0.0110	0.9676±0.0104	0.9595±0.0103
60%	**0.9688±0.0146**	0.9556±0.0195	0.9585±0.0118	0.9681±0.0087	0.9639±0.0117	0.9682±0.0108	0.9639±0.0118	0.9673±0.0108	0.9683±0.0120
	Missing features -2, 6, 1								
10%	**0.9749±0.0098**	0.9728±0.0117	0.9708±0.0082	0.9676±0.0109	0.9688±0.0112	0.9694±0.0106	0.9663±0.0120	0.9647±0.0102	0.9693±0.0130
20%	**0.9695±0.0123**	0.9659±0.0126	0.9625±0.0033	0.9692±0.0097	0.9668±0.0080	0.9648±0.0096	0.9595±0.0103	0.9645±0.0129	0.9615±0.0099
30%	**0.9707±0.0176**	0.9653±0.0110	0.9610±0.0079	0.9681±0.0096	0.9608±0.0168	0.9674±0.0104	0.9654±0.0137	0.9650±0.0096	0.9639±0.0113
40%	0.9685±0.0102	0.9675±0.0129	**0.699±0.0068**	0.9675±0.0114	0.9629±0.0101	0.9625±0.0122	0.9649±0.0110	0.9637±0.0084	0.9634±0.0111
50%	**0.9691±0.0075**	0.9648±0.0139	0.9689±0.0081	0.9670±0.0088	0.9610±0.0150	0.9633±0.0107	0.9527±0.0100	0.9611±0.0118	0.9624±0.0153
60%	**0.9659±0.0049**	0.9366±0.0208	0.9585±0.0139	0.9652±0.0097	0.9651±0.0199	0.9639±0.0086	0.9615±0.0081	0.9615±0.0096	0.9610±0.0142

Table 5.7: PERFORMANCE RESULTS FOR THE *Titanic* DATASET

Missing rates	Missing feature - 2								
	proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
10%	**0.8115±0.0232**	0.8067±0.0224	0.8025±0.0339	0.8052±0.0112	0.7925±0.0187	0.7953±0.0206	0.8007±0.0287	0.7905±0.0216	0.7869±0.0217
20%	**0.7878±0.0204**	0.7809±0.0071	0.7822±0.0188	0.7728±0.0238	0.7760±0.0236	0.7578±0.0132	0.7610±0.0229	0.7609±0.0151	0.7674±0.0191
30%	**0.7851±0.0142**	0.7594±0.0108	0.7754±0.0151	0.7708±0.0226	0.7715±0.0221	0.7503±0.0238	0.7536±0.0326	0.7415±0.0195	0.7427±0.0277
40%	**0.7828±0.0198**	0.7533±0.0072	0.7666±0.0215	0.7640±0.0262	0.7561±0.0239	0.7466±0.0108	0.7534±0.0198	0.7428±0.0142	0.7482±0.0257
50%	**0.7740±0.0192**	0.7469±0.0189	0.7654±0.0505	0.7409±0.0078	0.7416±0.0384	0.7219±0.0213	0.7221±0.0208	0.7303±0.0132	0.7307±0.0226
60%	**0.7461±0.0244**	0.7432±0.0270	0.7402±0.0357	0.7419±0.0205	0.7446±0.0144	0.7141±0.0150	0.7161±0.0225	0.7116±0.0120	0.7139±0.0201
	Missing features - 2, 6								
10%	**0.7940±0.0112**	0.7853±0.0192	0.7867±0.0254	0.7815±0.0078	0.7790±0.0211	0.7809±0.0163	0.7820±0.0198	0.7802±0.0206	0.7818±0.0153
20%	**0.7881±0.0276**	0.7800±0.0194	0.7850±0.0160	0.7628±0.0284	0.7753±0.0191	0.7540±0.0185	0.7638±0.0188	0.7627±0.0233	0.7667±0.0185
30%	**0.7740±0.0213**	0.7533±0.0218	0.7642±0.0246	0.7615±0.0244	0.7703±0.0174	0.7590±0.0057	0.7596±0.0217	0.7403±0.0099	0.7488±0.0308
40%	**0.7747±0.0206**	0.7530±0.0122	0.7669±0.0229	0.7618±0.0120	0.7326±0.0248	0.7458±0.0214	0.7506±0.0249	0.7415±0.0173	0.7498±0.0245
50%	**0.7703±0.0213**	0.7513±0.0205	0.7636±0.0303	0.7465±0.0228	0.7408±0.0192	0.7216±0.0255	0.7494±0.0239	0.7301±0.0132	0.7364±0.0183
60%	**0.7701±0.0156**	0.7450±0.0207	0.7606±0.0299	0.7405±0.0150	0.7423±0.0302	0.7128±0.0135	0.7464±0.0262	0.7139±0.0216	0.7479±0.0150
	Missing features -2, 6, 1								
10%	**0.7842±0.0228**	0.7817±0.0199	0.7826±0.0205	0.7665±0.0213	0.7693±0.0252	0.7790±0.0209	0.7798±0.0266	0.7753±0.0281	0.7809±0.0209
20%	**0.7828±0.0169**	0.7802±0.0187	0.7808±0.0270	0.7618±0.0132	0.7588±0.0279	0.7527±0.0086	0.7629±0.0189	0.7440±0.0099	0.7464±0.0266
30%	**0.7640±0.0163**	0.7528±0.0196	0.7605±0.0121	0.7528±0.0172	0.7538±0.0154	0.7466±0.0249	0.7610±0.0247	0.7409±0.0065	0.7385±0.0186
40%	**0.7541±0.0189**	0.7483±0.0232	0.7469±0.0266	0.7505±0.0120	0.7348±0.0114	0.7441±0.0173	0.7367±0.0209	0.7378±0.0182	0.7315±0.0213
50%	**0.7516±0.0212**	0.7442±0.0130	0.7450±0.0202	0.7319±0.0250	0.7363±0.0115	0.7206±0.0264	0.7330±0.0175	0.7214±0.0142	0.7206±0.0236
60%	**0.7592±0.0057**	0.7449±0.0196	0.7504±0.0302	0.7079±0.0192	0.6944±0.0210	0.7116±0.0163	0.6918±0.0285	0.7089±0.0234	0.6940±0.0292

Table 5.8: PERFORMANCE RESULTS FOR THE *German* DATASET

Missing rates	Proposed method	case deletion		mean imputation		k-NN3		k-NN10	
		LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
	Missing feature(s) - 1								
10%	0.7667±0.0165	0.7556±0.0120	0.7600±0.0109	**0.7757±0.0171**	0.7670±0.0105	0.7567±0.0202	0.7547±0.0219	0.7573±0.0225	0.7640±0.0132
20%	**0.7644±0.0255**	0.7533±0.0231	0.7558±0.0172	0.7583±0.0300	0.7587±0.0166	0.7530±0.0229	0.7520±0.0204	0.7607±0.0168	0.7633±0.0233
30%	**0.7589±0.0158**	0.7393±0.0227	0.7361±0.0212	0.7473±0.0173	0.7480±0.0206	0.7400±0.0091	0.7420±0.0150	0.7420±0.0344	0.7427±0.0055
40%	**0.7533±0.0209**	0.7389±0.0226	0.7408±0.0120	0.7407±0.0159	0.7427±0.0210	0.7429±0.0209	0.7423±0.0077	0.7423±0.0088	0.7413±0.0250
50$	**0.7478±0.0287**	0.7367±0.0223	0.7385±0.0203	0.7296±0.0139	0.7350±0.0215	0.7411±0.0336	0.7393±0.0174	0.7267±0.0338	0.7307±0.0218
60%	**0.7322±0.0212**	0.7127±0.0201	0.7188±0.0212	0.7083±0.0171	0.7107±0.0112	0.7144±0.0393	0.7213±0.0186	0.7189±0.0550	0.7183±0.0256
	Missing features - 1, 2								
10%	0.7644±0.0115	0.7600±0.0190	0.7630±0.0128	**0.7670±0.0170**	0.7630±0.0175	0.7543±0.0221	0.7553±0.0279	0.7650±0.0203	0.7667±0.0122
20%	**0.7598±0.0126**	0.7550±0.0212	0.7583±0.0056	0.7530±0.0228	0.7533±0.0171	0.7460±0.0253	0.7477±0.0215	0.7530±0.0203	0.7573±0.0028
30%	**0.7588±0.0242**	0.7430±0.0150	0.7469±0.0122	0.7477±0.0236	0.7487±0.0201	0.7407±0.0251	0.7467±0.0213	0.7460±0.0225	0.7420±0.0084
40%	**0.7466±0.0084**	0.7322±0.0160	0.7409±0.0203	0.7313±0.0234	0.7320±0.0222	0.7344±0.0139	0.7347±0.0238	0.7448±0.0019	0.7390±0.0208
50%	**0.7344±0.0204**	0.7260±0.0335	0.7333±0.0206	0.7264±0.0184	0.7307±0.0106	0.7342±0.0096	0.7293±0.0202	0.7222±0.0168	0.7280±0.0281
60%	**0.7356±0.0184**	0.7067±0.0162	0.7100±0.0156	0.7022±0.0184	0.7140±0.0140	0.7133±0.0150	0.7137±0.0214	0.7149±0.0151	0.7138±0.187
	Missing features - 1, 2, 3								
10%	**0.7600±0.0209**	0.7497±0.0156	0.7570±0.0184	0.7587±0.0142	0.7580±0.0208	0.7497±0.0175	0.7560±0.0130	0.7557±0.0246	0.7547±0.0090
20%	**0.7589±0.0038**	0.7525±0.0197	0.7558±0.0088	0.7500±0.0174	0.7547±0.0207	0.7313±0.0159	0.7367±0.0103	0.7487±0.0224	0.7447±0.0107
30%	**0.7567±0.0233**	0.7366±0.0210	0.7389±0.0217	0.7443±0.0166	0.7410±0.0201	0.7353±0.0168	0.7253±0.0266	0.7428±0.0078	0.7430±0.0267
40%	**0.7411±0.0215**	0.7333±0.0120	0.7389±0.0116	0.7387±0.0164	0.7370±0.0222	0.7347±0.0058	0.7380±0.0201	0.7311±0.0334	0.7380±0.0180
50%	**0.7322±0.0139**	0.7207±0.0115	0.7293±0.0218	0.7267±0.0167	0.7273±0.0203	0.7267±0.0145	0.7240±0.0192	0.7219±0.0051	0.7260±0.0195
60%	**0.7244±0.0254**	0.7105±0.0205	0.7167±0.0129	0.7037±0.0122	0.7073±0.0205	0.7112±0.0069	0.7160±0.0158	0.7156±0.0051	0.7133±0.0211

Table 5.9: PERFORMANCE RESULTS FOR THE COMMUNITY HEALTH CARE DATASET

Proposed method	Case deletion		Mean imputation		k-NN3		k-NN10	
	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs	LS-SVMs	SVMs
0.7258±0.0289	0.6954±0.0479	0.7023±0.031	0.7147±0.0330	0.7164±0.0271	0.7110±0.0229	0.7187±0.0211	0.7190±0.0305	0.7229±0.023

Table 5.10: PERFORMANCE RESULTS AFTER DATA CLEANING FOR THE COMMUNITY HEALTH CARE DATASET

Threshold	0.60	0.65	0.80	1.00
Performance	0.7265±0.0212	0.7288±0.0210	**0.7327±0.0098**	0.7300±0.0331

Table 5.11: AVERAGE RANKINGS OF THE PROPOSED AND COMPARATIVE METHODS ON SEVEN PUBLIC DATASETS IN TERMS OF AVERAGE ACCURACY (p-VALUE=0.000704)

Methods	Ranking
Proposed method	1
case deletion + SVMs	3
mean imputation + SVMs	5
case deletion + LS-SVMs	5
mean imputation + LS-SVMs	5.4286
k-NN10 + SVMs	5.7143
k-NN10 + LS-SVMs	6.1429
k-NN3 + SVMs	6.7143
k-NN3 + LS-SVMs	7

5.5 Summary

Missing data is an inevitable problem in many real world health prediction applications. In this chapter, a transfer-based additive LS-SVMs model is proposed to handle missing data from a transfer learning perspective. The proposed model can learn the model weights from the source domain with the complete portion of the dataset, and transfer the learned knowledge to the target domain with missing data. The model can simultaneously determine the influence on the classification error caused by each incomplete sample using a fast leave-one-out cross validation strategy, which provides an alternative way to clean data to further improve the quality of data.

Experiments were conducted to compare the proposed model with different

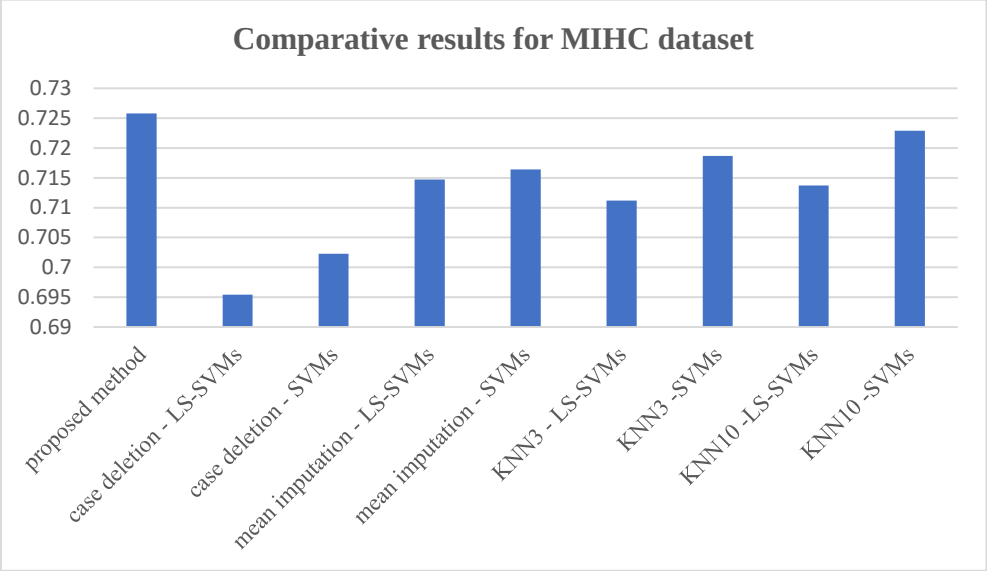


Figure 5.3: COMPARATIVE RESULTS FOR THE COMMUNITY HEALTH CARE DATASET

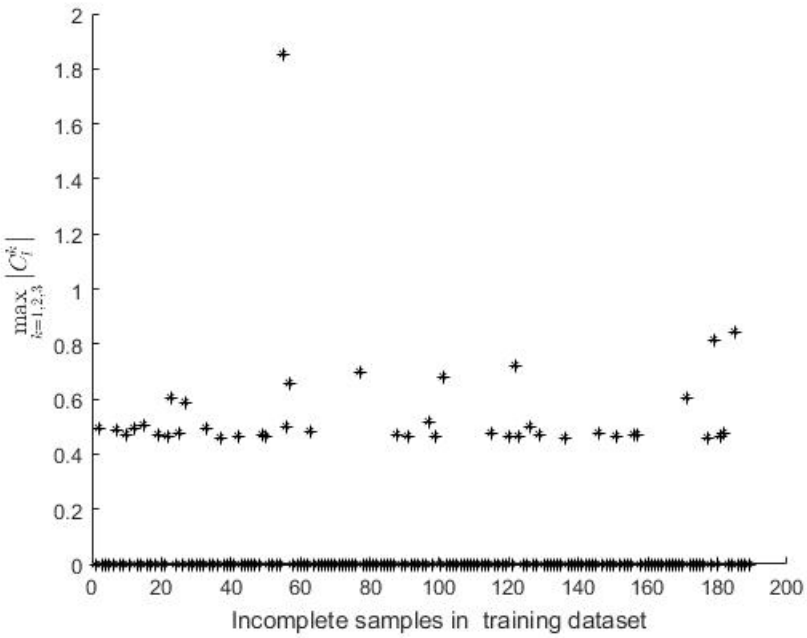


Figure 5.4: COMPARATIVE RESULTS AFTER DATA CLEANING FOR THE COMMUNITY HEALTH CARE DATASET

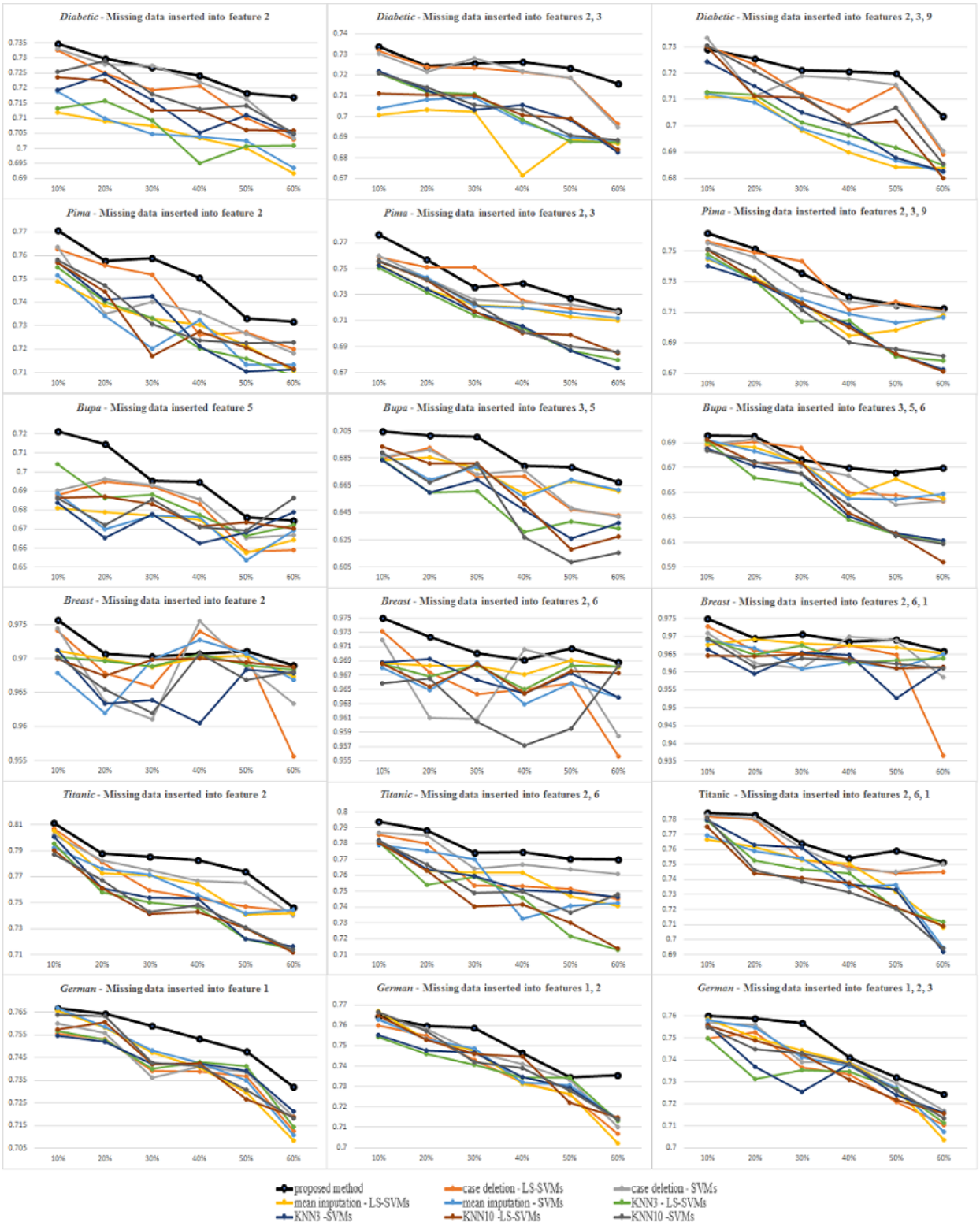


Figure 5.5: COMPARATIVE RESULTS OF PROPOSED AND COMPARATIVE METHODS ON SEVEN PUBLIC DATASETS

Table 5.12: HOLM POST-HOC COMPARISON RESULTS FOR THE PROPOSED AND COMPARATIVE METHODS IN TERMS OF AVERAGE ACCURACY WITH $\alpha = 0.05$

i	Methods	z-value	p-value	Holm=α/i
8	k-NN3 + LS-SVMs	4.09878	0.000042	0.00625
7	k-NN3 + SVMs	3.9036	0.000095	0.007143
6	k-NN10 + LS-SVMs	3.51324	0.000443	0.008333
5	k-NN10 + SVMs	3.22047	0.00128	0.01
4	mean imputation + LS-SVMs	3.02529	0.002484	0.0125
3	case deletion + LS-SVMs	2.73252	0.006285	0.016667
2	mean imputation + SVMs	2.73252	0.006285	0.025
1	case deletion + SVMs	1.36626	0.171857	0.05

traditional missing data treatments followed by LS-SVMs using the public UCI datasets. Experimental results confirm the effectiveness of the proposed model for classification with different combinations of missing data rates and missing features. Moreover, the proposed model is employed in a real world prediction of elderly QOL using a community health care dataset, which highlights the benefits and contributions of the proposed model to the health care application.

Chapter 6

A Deep Transfer Additive LS-SVMs Model for Predicting Elderly QOL with Imbalance Data

*The content of this Chapter was published in Wang et al. [2017b]:

Wang, G., Zhang, G., Choi, K.S. and Lu, J., "Deep additive Least Squares Support Vector Machines for classification with model transfer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017 (Accepted and on-line available)

6.1 Introduction

The additive kernel LS-SVMs (AK-LS-SVMs) have been widely used in many applications due to inherent advantages Duvenaud et al. [2011]; Maji et al. [2013]; Vedaldi and Zisserman [2012]. Additive kernels are commonly used specific for certain tasks including vision recognition, medical data analytics, and some specialized real-world scenarios Salgado et al. [2016]. Additionally, the analytical solution of AK-LS-SVMs can help formulate the fast leave-one-out cross-validation error estimate, which can drastically reduce the computational cost. However, AK-LS-SVMs still remain two problems. First, the classification performance of AK-LS-SVMs is comparatively low. Second, grid search for parameter tuning, such as generalization parameter C is time consuming. To overcome these problems, in this chapter, a novel model called deep transfer additive LS-SVMs (DTA-LS-SVMs) which stacks several AK-LS-SVMs based modules in a deep architecture and embeds model transfer learning is proposed. Considering that class imbalance problems are very common in the health data, we also gives an imbalanced version of the proposed model called iDTA-LS-SVMs to deal with imbalanced datasets. The generalization performances of the proposed model and its imbalanced version are expected to be improved by augmenting data input space via the hierarchical architecture, such that the manifold structure of the original data can be opened up to make it more separable. Besides, transfer learning is embedded to guarantee the consistency between adjacent modules to further enhance the higher module' classification capability. In the proposed model, the regularization parameter in each module can be randomly selected which also simplifies the learning process.

Extensive experiments are conducted on the public UCI datasets to compare the

performances of the proposed model and the traditional SVMs and LS-SVMs using additive kernels. The results show that DTA-LS-SVMs and iDTA-LS-SVMs can achieve better classification performance at a faster learning speed. A case study is conducted on the community health care dataset for predicting the elderly QOL, demonstrating the advantages of using the proposed model to handle class imbalance problems in the practical application in health care.

This chapter is organized as follows. Section 6.2 presents the proposed DTA-LS-SVMs and its imbalanced version iDTA-LS-SVMs. Section 6.3 extends the proposed model on class imbalance problems. Section 6.4 shows the experimental evaluations on the UCI public datasets. Section 6.5 shows a case study on a real world community health care dataset. Section 6.6 shows the statistical analysis of classification performances. Section 6.7 concludes the chapter.

6.2 Deep Transfer Additive LS-SVMs Model

6.2.1 Framework of the Proposed Model

Fig. 6.1 illustrates the framework of the proposed model. DTA-LS-SVMs consist of multiple AK-LS-SVMs based modules via a deep architecture. In the first layer, the original data input is used to construct the traditional AK-LS-SVMs. From the second layer, the original data input and the predicted output from the previous module are concatenated to be the new data input space to train the adjacent higher module. Moreover, model transfer learning is embedded to leverage the learned knowledge from the adjacent previous module to facilitate the learning process in the higher module.

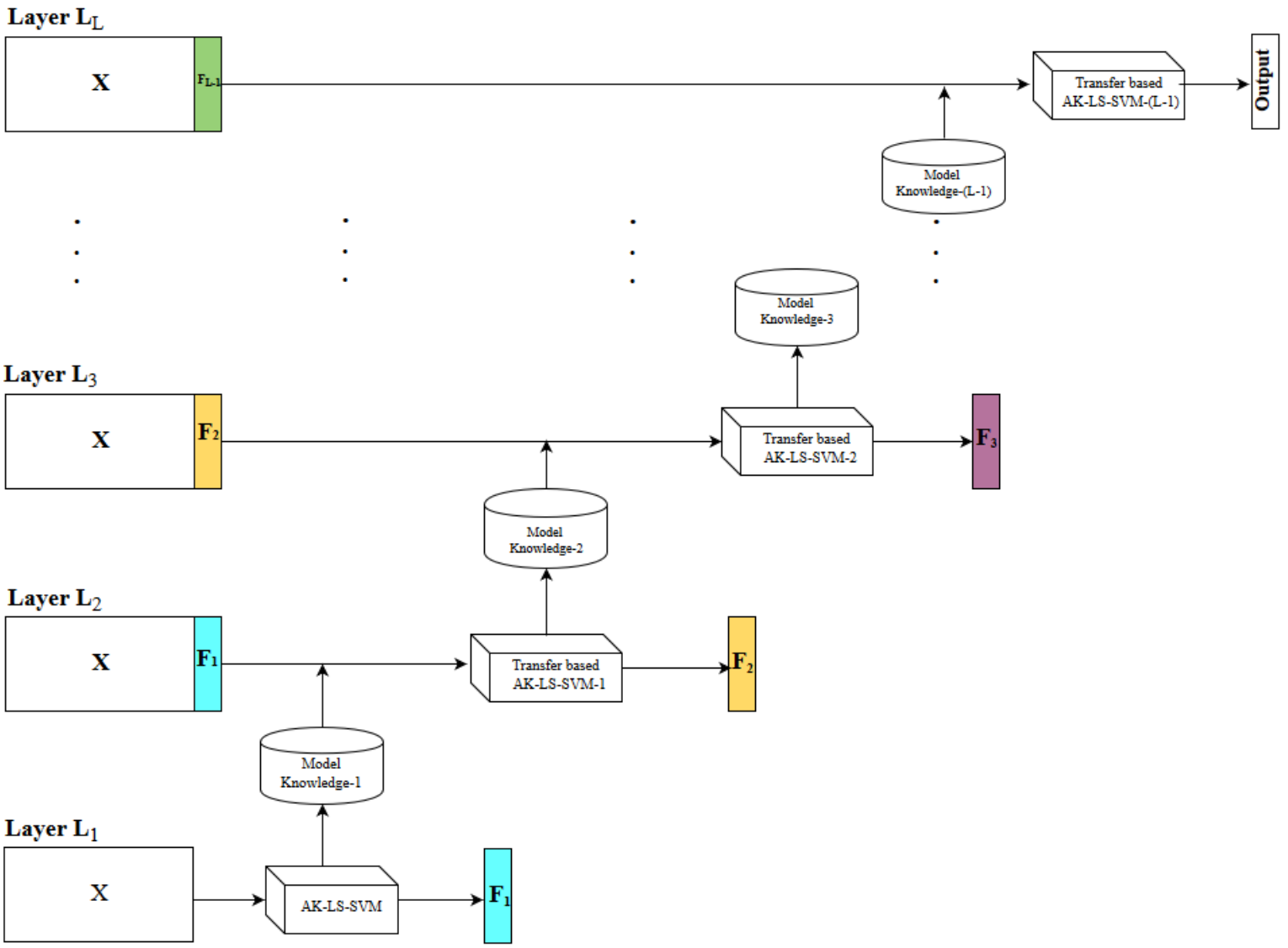


Figure 6.1: THE FRAMEWORK OF THE PROPOSED MODEL

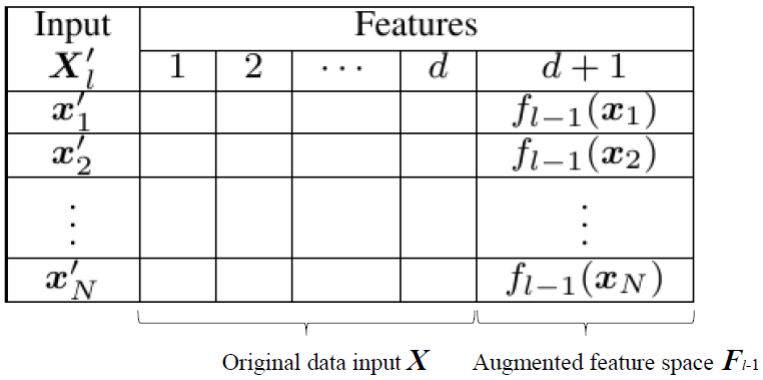


Figure 6.2: THE AUGMENTED SPACE $\vec{X}'_l$ OF $\vec{X}$

6.2.2 Deep Transfer Additive LS-SVMs Model

The main motivation of using deep hierarchy architecture in the proposed model is that it has a tendency to have a better performance at untangling the underlying factors of variation Bengio et al. [2013]. The recursive leverage of the predicted outcome from the adjacent previous module can help open the manifold of the original data space to make it more separable Vinyals et al. [2012].

Given a training dataset $\mathbf{S}$ of N samples $\mathbf{S} = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_N, y_N)\}$, the input dataset is denoted as $\mathbf{X}$ and the corresponding output dataset is denoted as $\mathbf{Y}$, where $\vec{x}_i = (x_1^i, x_2^i, \cdots, x_d^i) \in \mathbf{X} \subset \mathbf{R}^d$, $y_i \in \mathbf{Y} = \{+1, -1\}$. Each sample $\vec{x}_i$ contains d features. In the first layer L_1, the traditional AK-LS-SVMs is constructed and we can get the decision function $f_1(\vec{x}_i) = \vec{w}_1^T \varphi(\vec{x}_i) + b_1$ and consequently the predicted label vector $\vec{F}_1$ of $\mathbf{X}$, where $\vec{F}_1 = (f_1(\vec{x}_1), f_1(\vec{x}_2), \cdots, f_1(\vec{x}_N))$.

For the layer L_l $(l = 2, 3, ...L)$, the new data input $\mathbf{X}'_l$ is the concatenation of the original data input $\mathbf{X}$ and the predicted label vector $\vec{F}_{l-1}$ from the previous module, which can be denoted as $\mathbf{X} \oplus \vec{F}_{l-1}$ for simplicity. $\mathbf{X}'_l$ is illustrated in Fig. 6.2. AK-LS-SVMs with model transfer is used for the layer L_l to leverage the knowledge learned

from the source model in L_{l-1} to have $\lambda_l \vec{w}_{l-1}$, which can be obtained by reformulating the minimization problem of AK-LS-SVMs as:

$$\begin{aligned} \min_{\vec{w}_l, b_l} \quad & \frac{1}{2}(\vec{w}_l - \lambda_l \vec{w}_{l-1})^2 + \frac{C_l}{2}\sum_{i=1}^{N} \xi_{li}^2 \\ \text{s.t.} \quad & y_i = \vec{w}_l^T \varphi(\vec{x}_i^{'}) + b_l + \xi_{li} \\ & i = 1, 2, ..., N \end{aligned} \tag{6.1}$$

Lagrangian $\mathcal{L}_l$ of Eq. (6.1) is

$$\mathcal{L}_l(\vec{w}_l, b_l, \vec{\xi}_l; \vec{\alpha}_l) = J(\vec{w}_l, b_l) + \textstyle\sum_{i=1}^{N} \alpha_{li}(y_i - \vec{w}_l^T \varphi(\vec{x}_i^{'}) - b_l - \xi_{li}) \tag{6.2}$$

where $\vec{\alpha}_l \in \mathbf{R}^N$ is the vector of all Lagrangian multipliers. The system of linear equations can be obtained

$$\textstyle\sum_{i=1}^{N} \alpha_{li} \varphi(\vec{x}_i^{'})^T \varphi(\vec{x}_j^{'}) + b_l + \alpha_{li}/C_l = y_i - \lambda_l \vec{w}_{l-1}^T \varphi(\vec{x}_i^{'}) \tag{6.3}$$

The equation can be further written in matrix form as:

$$\begin{bmatrix} \widetilde{\mathbf{K}}_l + C_l^{-1}\mathbf{I} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha}_l \\ b_l \end{bmatrix} = \begin{bmatrix} \vec{Y} - \lambda_l \widetilde{\vec{Y}}_l \\ 0 \end{bmatrix} \tag{6.4}$$

where $\widetilde{\mathbf{K}}_l = [\mathbf{K}(\vec{x}_i^{'}, \vec{x}_j^{'})]_{N*N}$, $\mathbf{I}$ is a diagonal matrix with unity diagonal entries, $\vec{Y}$ is the actual labels of training samples, and $\widetilde{\vec{Y}}_l$ is the predicted labels of training samples obtained from the source model, i.e. $\vec{Y} = [y_1; \cdots; y_N]$, $\widetilde{\vec{Y}}_l = [y_{l1}^{'}; \cdots; y_{lN}^{'}] = \left[\vec{w}_{l-1}^T \varphi(\vec{x}_{l1}^{'}); \cdots; \vec{w}_{l-1}^T \varphi(\vec{x}_{lN}^{'})\right] = \left[\sum_{i=1}^{N} \alpha_{(l-1)i} \mathbf{K}(\vec{x}_{(l-1)i}, \vec{x}_{l1}^{'}), \cdots, \sum_{i=1}^{N} \alpha_{(l-1)i} \mathbf{K}(\vec{x}_{(l-1)i}, \vec{x}_{lN}^{'})\right]$. Here $\widetilde{\vec{Y}}_l$ can be obtained in a kernel form as long as $\vec{w}_{l-1}$ in the previous model and $\vec{w}_l$

in the current model are using the same kernel. Hence, for the safe use of the adopted Gaussian additive kernel, δ in each model must be the same.

Lastly, the model parameters can be calculated simply by using a matrix inversion:

$$\begin{bmatrix} \vec{\alpha}_l \\ b_l \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \vec{Y} - \lambda_l \widetilde{\vec{Y}}_l \\ 0 \end{bmatrix} \tag{6.5}$$

where $\mathbf{Q}_l = \mathbf{H}_l^{-1}$ and $\mathbf{H}_l$ is the first matrix from the left in Eq. (6.4). $\vec{w}$ can be determined by

$$\vec{w} = \lambda_l \vec{w}_{l-1} + \sum_{l=1}^{N} \alpha_{li} \varphi(\vec{x}_i^{'}) \tag{6.6}$$

Once λ_l is obtained, $\vec{\alpha}_l$ and b_l can be calculated. and then $\vec{w}_l$ and b_l can be calculated consequently. As a result, we can obtain the decision function $f_l(\vec{x}_{il}^{'}) = \vec{w}_l^T \varphi(\vec{x}_{il}^{'}) + b_l$ on L_l ($l \geq 2$), and the predicted label vector $\vec{F}_l = (f_l(\vec{x}_{1l}^{'}), f_l(\vec{x}_{2l}^{'}), \cdots, f_l(\vec{x}_{Nl}^{'}))$ for $\mathbf{X}_l'$. The layer continues to be added until the prediction accuracy performance has no improvement or the improvement is negligible, (i.e., $||\vec{F}_l - \vec{F}_{l-1}||_F^2 < \epsilon$). Here, the complete learning algorithm of the proposed model is given in Algorithm 6.1 that outputs the final decision function. Please note that in this study we select the parameter C_l from comparatively big intervals, i.e., $C_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$, to guarantee diversities between the modules from adjacent layers. However C_l can also be selected from different intervals depending upon the practical situations.

In DTA-LS-SVMs, transfer learning is embedded from the second layer. Referring to Eq. (6.6), we can observe that the classification on the l-th $(l >= 2)$ layer is in fact achieved in a way like a combination of multiple kernel functions from different layers. According to the excellent generalization performances of multi-kernel classifiers

Lanckriet et al. [2004]; Senechal et al. [2011]; Yeh et al. [2011], the module from the second layer in the proposed approach is supposed to obtain a better generalization performance than the previous one. In this sense, the learning process under such a stacked architecture tends to be greedy. Our experiments show that normally $L = 3, 4$ or 5 is an appropriate reference for small or medium datasets. If L is too big, it might lead to overfitting issues.

DTA-LS-SVMs can be used to solve multi-class classification problems by using the one-against-all strategy. Thus, the predicted output of the new sample $\vec{x}_i$ is determined by $\max_{k=1,\cdots,M} y_k(\vec{x}_i)$, where M denotes the number of the classes.

6.2.3 Fast Leave-one-out Cross Validation Strategy

The classification performance of the proposed model highly relies on the value of parameter λ_l. The fast leave-one-out cross validation strategy introduced in Chapter 3.2.5 is employed to find the optimal value of λ_l.

Similarly, by defining $\left[\vec{\alpha}_l^{'T}, b_l^{'}\right]^T = \mathbf{Q}_l \left[\vec{y}^T, 0\right]^T$, $\left[\vec{\alpha}_l^{''T}, b_l^{''}\right]^T = \mathbf{Q}_l \left[\vec{Y}_l^T, 0\right]^T$, and $\vec{\alpha}_l = \vec{\alpha}_l^{'} - \sum_{l=1}^{d} \lambda_l \vec{\alpha}_l^{''}$, the leave-one-out output $\tilde{y}_i$ of the i-th training sample can be represented as

$$\tilde{y}_{il} = y_i - \frac{\alpha_{il}^{'}}{Q_{iil}} + \frac{\lambda_l \alpha_{il}^{''}}{Q_{iil}} \tag{6.7}$$

The loss function below is adopted to avoid local minima issues:

$$l(\tilde{y}_{il}, y_i) = |1 - \tilde{y}_{il} y_i|_+ = \left| y_i \frac{\alpha_{il}^{'} - \lambda_l \alpha_{il}^{''}}{Q_{iil}} \right|_+ \tag{6.8}$$

where $|x|_+ = \max\{0, x\}$. The objective function becomes:

$$\begin{aligned} &\sum_{i=i}^{N} l(\tilde{y}_{il}, y_i) \\ \text{s.t} \quad &0 \leq \lambda_l \leq D \end{aligned} \tag{6.9}$$

where D is a constant. This optimization process can be implemented using a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 6.2.

6.2.4 Computational Complexity

One highlight of the proposed classifier DTA-LS-SVMs is its fast leave-one-out cross validation strategy for parameter tuning. The computational cost of DTA-LS-SVMs can be represented as $O(N^3 + (L-1)(N^3 + N)) = O(LN^3 + (L-1)N)$. $O(N^3)$ represents the computational cost for training the traditional AK-LS-SVMs module in the first layer. From the layer $(l \geq 2)$, the computational cost of each module consists of two parts. The first part $O(N^3)$ is the calculation of the matrix $\mathbf{Q}_l$ inverse related to the training set on the L_l. The second part $O(N)$ is the computational complexity to optimize Eq. (6.9) in each iteration in Algorithm 6.2.

Let us consider the traditional leave-one-out cross-validation strategy on SVMs. Theoretically, it takes $O(N^3)$ to train the SVMs. By using specific speed-up strategies Tsang et al. [2006], the training time can be accelerated to $O(N) - O(N^{2.3})$. Therefore, the computational cost of the leave-one-out cross validation becomes $O(N * N) - O(N * N^{2.3}) = O(N^2) - O(N^{3.3})$. For grid search for generalization parameter C (s_1 grid values) and kernel width σ (s_2 grid values), the computational cost is increased to $s_1 s_2 O(N^2) - s_1 s_2 O(N^{3.3})$. Normally s_1 and s_2 are greater than 3 in the experiment

Algorithm 6.1: Learning algorithm of DTA-LS-SVMs
Input: training set $\vec{X} = [\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N]$, $\vec{x}_i \in \vec{R}^d$, output set $\vec{Y} = [y_1, y_2, \cdots, y_N]$, $y_i \in \{+1, -1\}$ for binary classification, kernel width δ, number of layers L, $l = 1$
Output: The stacked structure of DTA-LS-SVMs with tuned parameter values
Procedure
Step 1: 1.1 Choose the regularization parameter C_1 randomly from an interval, i.e., $C_1 \in \{1, 10, 50, 100, 150, 200, 250, 500\}$. 1.2 Construct the 1st module using the traditional additive kernel LS-SVM shown and obtain $\vec{w}_1$, b_1 and the predicted labels $\vec{F}_1$ $(f_1(\vec{x}_{11}), f_1(\vec{x}_{21}), \cdots, f_1(\vec{x}_{N1}))$. Step 2: For $l = 2 : L$ do 2.1 $\vec{X}_l' = \vec{X} \oplus \vec{F}_{l-1}$ 2.2 Choose the regularization parameter C_l randomly from an interval, i.e., $C_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$. 3.3 Construct the l-th module by invoking Algorithm 6.2 on $\vec{X}_l'$ and obtain λ_l. 2.4 Calculate $\vec{w}_l$ and the predicted labels $\vec{F}_l$ $(f_l(\vec{x}_{1l}), f_l(\vec{x}_{2l}), \cdots, f_l(\vec{x}_{Nl}))$. Step 3: Calculate $\triangle_F = \|\vec{F}_l - \vec{F}_{l-1}\|_F^2$ Step 4: If $\triangle_F \leq \epsilon$ (a given threshold) End else Step 5: $l = l + 1$ Step 6: Output the stacked structure of the proposed classifier DTA-LS-SVM with tuned parameter values and the decision function in the L-th module as the final decision function.

Algorithm 6.2: Projected Sub-gradient Descent Algorithm
Input: $\vec{w}_{l-1}$, $\vec{X}_l^{'}$, $\vec{Y}$, C_l and kernel width σ
Output: λ_l
Procedure
Step 1: Calculate $\mathbf{Q}_l$, $\vec{\alpha}_l^{'}$, $\vec{\alpha}_l^{''}$ Step 2: $t = 1$ Step 3: Repeat $\tilde{y}_{il} = y_i - \frac{\alpha_{li}^{'}}{\vec{Q}_{iil}} + \frac{\lambda_l \alpha_{il}^{''}}{\vec{Q}_{iil}}, i = 1, 2, ..., N$ $d_i \leftarrow \vec{1}\{\tilde{y}_{il} y_i > 0\}, i = 1, 2, ..., N$ $\lambda_l \leftarrow \lambda_l - \frac{1}{\sqrt{t}} d_i y_i \frac{\alpha_{il}^{''}}{Q_{iil}}$ If $\lambda_l > D$ then $\lambda_l \leftarrow D$ End if $\lambda_l \leftarrow max(\lambda_l, 0)$ $t \leftarrow t + 1$ Step 4: Until convergence Step 5: Output λ_l

setting while the number of layers L in DTA-LS-SVMs is small ($3 \leq L \leq 5$). Therefore, although it seems that the SVMs are less computationally complex than DTA-LS-SVMs, experimentally, the actual running time of SVMs with grid search is much longer than that of DTA-LS-SVMs.

In terms of LS-SVMs, the computational complexity to train the LS-SVMs is $O(N^3)$ due to the calculation of the matrix $\mathbf{Q}$ by the inverse of $\mathbf{H}$. Therefore, its computational cost of leave-one-out cross validation with grid search becomes $s_1 s_2 O(N * N^3) = s_1 s_2 O(N^4)$. Referring to Eq. (6.7) and Eq. (6.8) with λ_l equal to 0, the computational complexity of LS-SVMs could be accelerated into $s_1 s_2 O(N^3 + N)$. In summary, the proposed classifier DTA-LS-SVMs has a superior advantage in the running speed compared with SVMs and LS-SVMs.

6.3 Extension on Class Imbalance Problems

Class imbalance problems frequently occur in many real world scenarios. We extend the proposed DTA-LS-SVMs model to its cost-sensitive or imbalanced version to specifically deal with it.

For classification with imbalanced data, the decision boundary tends to get too close to the minority class, which needs to be pushed away. One solution is to give different error costs to the positive and negative classes Batuwita and Palade [2013]. Thus, the optimization problem of AK-LS-SVMs is reformulated accordingly as below

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\|\vec{w}\|^2 + \frac{C}{2}\left[\frac{N^-}{N}\sum_{i=1}^{N^+}\xi_i^2 + \frac{N^+}{N}\sum_{i=N^++1}^{N}\xi_i^2\right] \\ \text{s.t.} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i \end{aligned} \tag{6.10}$$

where N^+ and N^- represent the number of positive and negative classes respectively. Consequently, according to Eq. (6.10), DTA-LS-SVMs can be extended to its imbalanced version called iDTA-LS-SVMs.

Similar to Eqs. (6.2)-(6.5), we observe that only $\mathbf{H}_l$ in the first matrix in Eq. (6.4) needs to be modified:

$$\mathbf{H}_l = \begin{bmatrix} \widetilde{\vec{K}}_l + C_l^{-1}\mathbf{E} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \tag{6.11}$$

where

$$\mathbf{E} = \begin{pmatrix} \frac{N^-}{N} & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{N^-}{N} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{N^+}{N} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{N^+}{N} \end{pmatrix} \tag{6.12}$$

The remaining derivations still remain the same. By comparing $\mathbf{H}_l$ in Eq. (6.11) with the first matrix in Eq. (6.4), the only difference is between $\mathbf{E}$ in Eq. (6.11) and $\mathbf{I}$ in Eq. (6.4). Only under the condition that N^- equals N^+, $\mathbf{E}$ degenerates into $\mathbf{I}$. Moreover, the fast leave-one-out cross validation strategy proposed in DTA-LS-SVMs still can be used in iDTA-LS-SVMs to reduce the high computational complexity.

6.4 Experiments

The proposed model DTA-LS-SVMs and its imbalanced version iDTA-LS-SVMs were evaluated on the balanced and imbalanced UCI datasets and compare the performances with those using the additive kernel LS-SVMs and additive kernel SVMs. In the data pre-processing stage, the input data were normalized. For any UCI datasets with missing data, case deletion treatment method was applied. For performance measurements, we use both accuracy and F1-score for balanced datasets and only F1-score for imbalanced datasets. F1-score Lewis and Gale [1994] given in Eq. (6.13) is a harmonic mean between precision and recall. A higher F1-Score implies that both recall and precision are comparatively higher.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.13}$$

where $\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$ and $\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$. All the experiments were implemented using 64-bit MATLAB on a computer with an Intel Core i5-6300 2.40 GHz CPU and 8.00GB RAM.

Table 6.1: UCI DATASETS DESCRIPTION

Type	Dataset	Sample size	Feature	Class(%)
Imbalanced	*breast cancer*	683	9	65.52 34.48
	Pima Indians	768	8	65.02 34.98
	Indians Liver	579	10	71.50 28.50
Balanced	*Australian*	690	14	44.50 55.50
	diabetic	1151	19	53.08 46.92
	credit approval	653	15	45.33 54.67
	mammographic	830	5	48.55 51.45

6.4.1 UCI datasets

The performances of the proposed model and the comparative methods are evaluated on seven public UCI datasets. The dataset information are summarized in Table 6.1.

6.4.2 Parameter Setup

In the experiments, different additive kernels were tested in DTA-LS-SVMs, iDTA-LS-SVMs and the comparative methods on each dataset. Here only the experimental results using the Gaussian additive kernel are displayed. For DTA-LS-SVMs and iDTA-LS-SVMs, the number of modules is usually set to 3 or 4 due to the small or medium sample size of the adopted datasets. ϵ is set to 0.1. δ is set to be the average value of the standard deviations for all respective features. For the comparative methods, the grid search algorithm with 10-fold cross-validation was used to find the optimal values for parameters C and δ. The intervals of

$\{1, 10, 50, 100, 150, 200, 250, 500\}$ and $\{0.1, 1, 5, 10, 20, 50, 100, 150, 200\}$ were searched for C and δ, respectively. Here only the performances using the optimal parameters are displayed.

6.4.3 Experimental Results Analysis

Table 6.2: PERFORMANCE RESULTS ON BALANCED UCI DATASETS

Datasets	Metrics	Performances					
		DTA-LS-SVM		LS-SVM		SVM	
		training	testing	training	testing	training	testing
Australian	Accuracy	0.8936±0.0099	**0.8678±0.0212**	0.8583±0.0098	0.8500±0.0266	0.8589±0.0075	0.8572±0.0174
	F1-score	0.9050±0.0091	**0.8766±0.0215**	0.8665±0.0060	0.8630±0.0193	0.8620±0.0082	0.8619±0.0190
diabetic	Accuracy	0.7720±0.0096	**0.7395±0.0197**	0.7391±0.0221	0.7220±0.0242	0.7275±0.0068	0.7188±0.0194
	F1-score	0.7668±0.0163	**0.7368±0.0202**	0.7434±0.0380	0.7262±0.0281	0.7262±0.0112	0.7153±0.0194
credit approval	Accuracy	0.8985±0.0075	**0.8786±0.0161**	0.8611±0.0138	0.8536±0.0203	0.8678±0.0087	0.8648±0.0230
	F1-score	0.9065±0.0095	**0.8895±0.0154**	0.8683±0.0097	0.8632±0.0205	0.8684±0.0079	0.8682±0.0192
mammographic	Accuracy	0.8231±0.0078	**0.8273±0.0233**	0.8155±0.0103	0.8121±0.0313	0.8189±0.0116	0.8104±0.0349
	F1-score	0.8227±0.0102	**0.8321±0.0266**	0.8172±0.0173	0.8167±0.0357	0.8181±0.0120	0.8137±0.0350

Table 6.3: PERFORMANCE RESULTS ON IMBALANCED UCI DATASETS

Datasets	Performances					
	iDTA-LS-SVM		LS-SVM		SVM	
	training	testing	training	testing	training	testing
breast cancer	0.9861±0.0064	**0.9801±0.0026**	0.9760±0.0074	0.9738±0.0069	0.9806±0.0041	0.9728±0.0123
Pima Indians	0.8748±0.0167	**0.8359±0.0097**	0.8401±0.0131	0.8284±0.0317	0.8336±0.0084	0.8285±0.0248
Indians liver	0.8438±0.0829	**0.8403±0.0083**	0.8390±0.0166	0.7986±0.0580	0.8343±0.0080	0.8323±0.0188

Table 6.4: PERFORMANCE RESULTS ON THE *Australian* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

Australia	Metrics	Performances							
		6:4		5:5		4:6		3:7	
		training	testing	training	testing	training	testing	training	testing
DTA-LS-SVM	Accuracy	0.9086±0.0087	**0.8572±0.0111**	0.9009±0.0148	**0.8577±0.0110**	0.9047±0.0215	**0.8589±0.0141**	0.9043±0.0175	**0.8551±0.0149**
	F1-score	0.9081±0.0083	**0.8716±0.0125**	0.9123±0.0121	**0.8727±0.0098**	0.9124±0.0199	**0.8742±0.0131**	0.9132±0.0171	**0.8703±0.0132**
LS-SVM	Accuracy	0.8425±0.0384	0.8417±0.0431	0.8417±0.0598	0.8304±0.0692	0.8413±0.0544	0.8283±0.0559	0.8449±0.0695	0.8265±0.0762
	F1-score	0.8586±0.0200	0.8595±0.0281	0.8631±0.0307	0.8546±0.0396	0.8600±0.0272	0.8519±0.0325	0.8651±0.0337	0.8496±0.0418
SVM	Accuracy	0.8621±0.0148	0.8504±0.0221	0.8649±0.0143	0.8487±0.0139	0.8717±0.0202	0.8498±0.0176	0.8797±0.0251	0.8478±0.0161
	F1-score	0.8663±0.0139	0.8564±0.0194	0.8716±0.0150	0.8529±0.0126	0.8788±0.0176	0.8563±0.0164	0.8871±0.0242	0.8556±0.0155

Table 6.5: PERFORMANCE RESULTS ON THE *Diabetic* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

Diabetic	Metrics	Performances							
		6:4		5:5		4:6		3:7	
		training	testing	training	testing	training	testing	training	testing
DTA-LS-SVM	Accuracy	0.7990±0.0082	**0.7386±0.0139**	0.8028±0.0125	**0.7241±0.0164**	0.8052±0.0121	**0.7159±0.0132**	0.8183±0.0150	**0.7076±0.0129**
	F1-score	0.7974±0.0090	**0.7331±0.0117**	0.7986±0.0154	**0.7127±0.0197**	0.8084±0.0186	**0.7071±0.0132**	0.8115±0.0240	**0.6965±0.0162**
LS-SVM	Accuracy	0.7445±0.0216	0.7180±0.0208	0.7431±0.0166	0.7163±0.0245	0.7315±0.0286	0.6991±0.0237	0.7209±0.0143	0.6815±0.0237
	F1-score	0.7451±0.0333	0.7270±0.0250	0.7377±0.0409	0.7096±0.0338	0.7355±0.0402	0.7021±0.0333	0.7268±0.0482	0.6770±0.0412
SVM	Accuracy	0.7330±0.0134	0.7015±0.0199	0.7297±0.0136	0.7038±0.0145	0.7193±0.0100	0.6987±0.0151	0.7075±0.0216	0.6875±0.0210
	F1-score	0.7350±0.0219	0.7038±0.0147	0.7262±0.0226	0.6936±0.0276	0.7339±0.0154	0.6965±0.0095	0.7004±0.0355	0.6733±0.0432

Table 6.6: PERFORMANCE RESULTS ON THE *credit approval* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

credit approval	Metrics	Performances							
		6:4		5:5		4:6		3:7	
		training	testing	training	testing	training	testing	training	testing
DTA-LS-SVM	Accuracy	0.8985±0.0071	**0.8664±0.0115**	0.9110±0.0100	**0.8621±0.0116**	0.9031±0.0119	**0.8615±0.0112**	0.9190±0.0149	**0.8614±0.0126**
	F1-score	0.9074±0.0075	**0.8773±0.0122**	0.9206±0.0107	**0.8708±0.0124**	0.9120±0.0107	**0.8720±0.0097**	0.9270±0.0134	**0.8687±0.0168**
LS-SVM	Accuracy	0.8509±0.0305	0.8481±0.0556	0.8426±0.0763	0.8202±0.0890	0.8295±0.0798	0.8120±0.0997	0.8508±0.0649	0.8328±0.0630
	F1-score	0.8623±0.0182	0.8608±0.0406	0.8614±0.0382	0.8411±0.0506	0.8609±0.0396	0.8405±0.0556	0.8673±0.0363	0.8560±0.0423
SVM	Accuracy	0.8681±0.0050	0.8622±0.0127	0.8696±0.0164	0.8612±0.0169	0.8778±0.0149	0.8564±0.0108	0.8677±0.0164	0.8537±0.0193
	F1-score	0.8695±0.0059	0.8628±0.0149	0.8716±0.0159	0.8625±0.0167	0.8802±0.0167	0.8604±0.0117	0.8754±0.0172	0.8600±0.0132

Table 6.7: PERFORMANCE RESULTS ON THE *mammographic* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

mammographic	Metrics	Performances							
		6:4		5:5		4:6		3:7	
		training	testing	training	testing	training	testing	training	testing
DTA-LS-SVM	Accuracy	0.8325±0.0045	**0.8187±0.0171**	0.8234±0.0120	**0.8224±0.0118**	0.8241±0.0173	**0.8106±0.0138**	0.8261±0.0240	**0.8095±0.0165**
	F1-score	0.8317±0.0078	**0.8093±0.0149**	0.8214±0.0130	**0.8234±0.0132**	0.8216±0.0264	**0.8091±0.0213**	0.8275±0.0267	**0.8050±0.0171**
LS-SVM	Accuracy	0.8213±0.0137	0.8051±0.0173	0.8212±0.0086	0.8039±0.0181	0.8250±.0180	0.8036±0.0186	0.8309±0.0172	0.8024±0.0149
	F1-score	0.8202±0.0166	0.8017±0.0201	0.8151±0.0081	0.7988±0.0183	0.8240±0.0202	0.8006±0.0178	0.8326±0.0209	0.7975±0.0153
SVM	Accuracy	0.8139±0.0125	0.8060±0.0284	0.8089±0.0124	0.8000±0.0153	0.8108±0.0197	0.7972±0.0215	0.8092±0.0277	0.7914±0.0153
	F1-score	0.8165±0.0119	0.8079±0.0264	0.8135±0.0138	0.7978±0.0159	0.8110±0.0187	0.8008±0.0189	0.8052±0.0296	0.7867±0.0234

Table 6.8: PERFORMANCE RESULTS ON THE *breast cancer* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

breast cancer	F1-score							
	6:4		5:5		4:6		3:7	
	training	testing	training	testing	training	testing	training	testing
iDTA-LS-SVM	0.9818±0.0038	**0.9798±0.0053**	0.9821±0.0040	**0.9787±0.0044**	0.9859±0.0065	**0.9796±0.0031**	0.9874±0.0043	**0.9775±0.0053**
LS-SVM	0.9735±0.0051	0.9742±0.0115	0.9684±0.0183	0.9649±0.0236	0.9750±0.0044	0.9675±0.0111	0.9745±0.0093	0.9711±0.0057
SVM	0.9811±0.0042	0.9711±0.0063	0.9797±0.0081	0.9749±0.0064	0.9821±0.0042	0.9726±0.0083	0.9854±0.0088	0.9718±0.0043

Table 6.9: PERFORMANCE RESULTS ON THE *Pima Indians* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

Pima Indians	F1-score							
	6:4		5:5		4:6		3:7	
	training	testing	training	testing	training	testing	training	testing
iDTA-LS-SVM	0.8665±0.0157	**0.8345±0.0171**	0.8697±0.0088	**0.8345±0.0066**	0.8700±0.0129	**0.8336±0.0121**	0.8918±0.0188	**0.8254±0.0088**
LS-SVM	0.8399±0.0141	0.8265±0.0239	0.8409±0.0179	0.8295±0.0205	0.8360±0.0252	0.8263±0.0189	0.8385±0.0280	0.8222±0.0178
SVM	0.8379±0.0104	0.8185±0.0138	0.8417±0.0175	0.8198±0.0161	0.8426±0.0150	0.8174±0.0075	0.8478±0.0134	0.8141±0.0135

Table 6.10: PERFORMANCE RESULTS ON THE *Indians liver* DATASET WITH DIFFERENT RATIOS OF TRAINING AND TESTING DATA

Indians liver	F1-score							
	6:4		5:5		4:6		3:7	
	training	testing	training	testing	training	testing	training	testing
iDTA-LS-SVM	0.8387±0.0104	**0.8375±0.0135**	0.8422±0.0085	**0.8373±0.0078**	0.8382±0.0113	**0.8336±0.0096**	0.8353±0.0177	**0.8294±0.0177**
LS-SVM	0.9197±0.0583	0.8285±0.0153	0.8714±0.0537	0.8117±0.0212	0.8850±0.0779	0.7995±0.0542	0.9203±0.0760	0.7813±0.0542
SVM	0.8369±0.0106	0.8294±0.0160	0.8420±0.0133	0.8253±0.0133	0.8410±0.0175	0.8288±0.0117	0.8363±0.0203	0.8226±0.0088

Table 6.2 lists the results of DTA-LS-SVMs and the comparative methods on imbalanced datasets. Table 6.3 lists the results of iDTA-LS-SVMs and the comparative methods on balanced datasets. For DTA-LS-SVMs and iDTA-LS-SVMs, if the training accuracy of the adjacent higher layer was improved compared with the previous layer, one more processing layer wa added, otherwise we terminated the learning process. We take a complete training process on the *Australian* dataset as an example. The first module obtained a training accuracy of 0.8589. The second module obtained an accuracy of 0.8734. The third module obtained an accuracy of 0.8880. Obviously the classification performance was improving. Therefore, we continued training the fourth module which obtained an accuracy of 0.8734. Since there was a decrease in the classification performance, we stopped the learning process. In table 6.2, DTA-LS-SVMs obtained considerably higher accuracies than those using the comparative methods on the balanced UCI datasets. In Table 6.3, iDTA-LS-SVMs maintained the advantage on the imbalanced datasets in terms of F1-score. We also observed that although in some cases LS-SVMs performed worse than SVMs, DTA-LS-SVMs and iDTA-LS-SVMs which are based on LS-SVMs can always achieve better performance results. Overall, the proposed DT-AK-LS-SVMs and iDTA-LS-SVMs exhibited good generalization performances on both balanced and imbalanced datasets.

To further evaluate the performance of the proposed model, we divided the training and testing datasets at different ratios. The ratios were set to 6:4, 5:5, 4:6 and 3:7 on each UCI dataset. Table 6.4-6.7 display the experimental results of DTA-LS-SVMs and the comparative methods on the balanced datasets in terms of accuracy and F1-score. Tables 6.8-6.10 display the experimental results of iDTA-LS-SVMs and the comparative methods for the imbalanced datasets in terms of F1-score. it can be seen that after changing the ratios of training and testing sets, the proposed classifiers

DTA-LS-SVMs and iDTA-LS-SVMs still outperformed the other methods on all the datasets. For example, in Table 6.5 for the *mammographic* dataset at the 5:5 ratio; DTA-LS-SVMs not only achieved the highest accuracy but also the highest F1-score on the testing sets. Another example is for the imbalanced *Pima Indians* dataset in Table 6.9 in which iDTA-LS-SVMs maintained an advantage over the comparative methods in terms of the F1-score. We also noticed that DTA-LS-SVMs and iDTA-LS-SVMs did not exhibit a rising trend in accuracy and F1-score when enlarging the size of the training dataset. This could be because in essential DTA-LS-SVMs and iDTA-LS-SVMs are multiple kernel combination methods that usually have a better representation capability on training small data.

Overall, from the experimental results, DTA-LS-SVMs and iDTA-LS-SVMs are tolerant to the changes of the training dataset sample size and is a favorable choice for classification on balanced and imbalanced datasets.

In terms of the running time, Table. 6.11 shows that DTA-LS-SVMs and iDTA-LS-SVMs run much faster at training compared to the other methods. As explained in Section 6.2.4, the running time of the proposed model is the summation of the time taken in each module and the time taken for parameter tuning of λ_l $(l = 1, 2, \cdots, L)$ using a fast leave-one-out cross validation strategy. The regularization parameter C_l $(l = 1, 2, \cdots, L)$ in each module can be randomly selected, which also significantly simplifies the learning process. In comparison, LS-SVMs and SVMs need to find the optimal values for C and δ in each module by grid search which is much more computationally expensive.

Table 6.11: TRAINING AND TESTING TIME (SECONDS) ON UCI DATASETS

Type	**Datasets**	Running time					
		(i)DTA-LS-SVMs		LS-SVMs		SVMs	
		training	testing	training	testing	training	testing
Imbalanced	*breast cancer*	2.278	1.979	8776.1	1.399	19553	2.498
	Pima Indians	2.731	2.376	10512	1.447	28950	3.018
	Indians liver	1.886	1.679	6488.1	1.249	13304	2.087
Balanced	*australian*	3.472	2.935	12103	1.835	22059	3.209
	diabetic	12.698	10.600	73863	6.568	129790	12.943
	credit approval	3.638	2.782	11660	2.203	18138	3.190
	mammographic	2.279	1.933	8340.3	1.034	29387	3.212

6.5 A Case Study on a Real World Community Health Care Dataset

6.5.1 Data Collection

The same community health care dataset introduced in Chapter 4.3.1 was employed to investigate the classification performances of the proposed model and its imbalanced version for predicting the elderly QOL. We re-categorized the QOL outcome based on the scale of 1 to 5 into two classes 'poor' and 'good' with the ratio of 1:2.64. The missing data were pre-processed using the k-NN imputation method.

6.5.2 Experimental Design

We applied iDTA-LS-SVMs on this imbalanced dataset and compared its classification performance with those using DTA-LS-SVMs, LS-SVMs, and SVMs. The number of processing layers was set to three. To make a detailed comparison on the performances of the proposed models, experiments were divided into two parts:

Exp (1): In the first layer of DTA-LS-SVMs and iDTA-LS-SVMs, the complete portion of the dataset was used to train an AK-LS-SVMs model. From the second layer, the whole dataset after k-NN imputation treatment and the outputs predicted from the previous layer were concatenated to be the new data input.

Exp (2): In the first layer of DTA-LS-SVMs and iDTA-LS-SVMs, the whole pre-processed dataset after imputation is the data input.

6.5.3 Results Analysis

Tables 6.12, 6.13, and 6.14 show the classification accuracies and running time of DTA-LS-SVMs, iDTA-LS-SVMs, LS-SVMs and SVMs. We can see that in Exp (2) iDTA-LS-SVMs achieved the highest accuracy (0.7425) and F1-score (0.8553) on the testing dataset among all the methods. In Exp (1), iDTA-LS-SVMs achieved an accuracy of 0.7331 and a F1-score of 0.8492, which still outperformed the other methods. In addition, DTA-LS-SVMs and iDTA-LS-SVMs have the superior advantage in terms of the running time. The experimental results demonstrated that iDTA-LS-SVMs have the strong capability for classification of imbalanced datasets in the real world sceneries. Moreover, although the traditional LS-SVMs gained the lowest accuracy and F1-score, iDTA-LS-SVMs consisting of several AK-LS-SVMs based modules can achieve the best accuracy. We believe such performance improvement is attributed to the deep stacked architecture and the embedded transfer learning.

We also found that DTA-LS-SVMs and iDTA-LS-SVMs had higher accuracy and F1-score on the testing dataset than the training dataset. There are two possible reasons to explain this. First, the deep stacked architecture resulted in the enhanced

performance result. Second, in the data pre-processing stage, missing data in the dataset were imputed using the k-NN imputation treatment. Therefore, the distributions in the training and testing datasets might be mismatched. Referring to the conclusions in González and Abu-Mostafa [2015], we postulate that mismatched distributions occurred in this experiment, which led to a higher testing accuracy and F1-score in the proposed model and its imbalanced version.

Table 6.12: PERFORMANCE RESULTS ON THE COMMUNITY HEALTH CARE DATASET USING iDTA-LS-SVMS

Performances	Exp (1)		Exp (2)	
	training	testing	training	testing
Accuracy	0.7261±0.0094	0.7331±0.0218	0.7210±0.0102	**0.7425±0.0266**
F1-score	0.8413±0.0063	0.8492±0.0147	0.8358±0.0100	**0.8553±0.0233**

Table 6.13: PERFORMANCE RESULTS ON THE COMMUNITY HEALTH CARE DATASET USING DTA-LS-SVMS AND THE OTHER COMPARATIVE METHODS

Performances	DTA-LS-SVMs				LS-SVMs		SVMs	
	Exp (1)		Exp (2)					
	training	testing	training	testing	training	testing	training	testing
Accuracy	0.7203±0.0133	0.7266±0.0309	0.7294±0.0146	0.7224±0.0391	0.7510±0.0452	0.7050±0.0398	0.7274±0.0166	0.7201±0.0384
F1-score	0.8374±0.0090	0.8380±0.0205	0.8430±0.0101	0.8379±0.0262	0.8452±0.0103	0.8268±0.0439	0.8435±0.0057	0.8341±0.0133

Table 6.14: TRAINING AND TESTING TIME (SECONDS) ON THE COMMUNITY HEALTH CARE DATASET

iDTA-LS-SVMs				DTA-LS-SVMs				LS-SVMs		SVMs	
Exp (1)		Exp (2)		Exp (1)		Exp (2)					
training	testing	traning	testing	training	testing	training	testing	training	testing	training	testing
5.530	5.201	5.837	5.437	4.343	4.019	4.673	4.217	4950.6	2.398	1157.8	4.183

6.6 Statistical Analysis

To test for statistical differences among the experimental results of the proposed model and the comparative methods, we conducted the Friedman test followed by Holm post-hoc test Demšar [2006]; Garcia and Herrera [2008] for multiple comparisons on four balanced and four imbalanced datasets. The Friedman ranking test evaluates whether there is a statistically significant difference among all the methods. If the p-value is smaller than 0.5, the null hypothesis that there is no significant difference will be rejected. The Holm post-hoc test further verifies whether there is a statistical difference between the outstanding Friedman ranking method and the other remaining method. The level of confidence is set as $\alpha = 0.05$. First, we conducted two Friedman ranking tests to evaluate differences between (1) DTA-LS-SVMs and the comparative methods on four balanced UCI datasets in terms of accuracy and F1-score; (2) iDTA-LS-SVMs and the comparative methods on three imbalanced UCI datasets and one real-world *TRUS* dataset in terms of F1-score. Experimental results of Exp (1) and Exp (2) on the *TRUS* dataset were included.

The results of Friedman test (1) in terms of accuracy and F1-score are shown in Tables 6.15 and 6.17, respectively. We can see that there are significant differences between DTA-LS-SVMs and other comparative methods using different performance measurements. Following that, we conducted the Holm post-hoc tests to compare the best ranking method DTA-LS-SVMs with LS-SVMs and SVMs in terms of accuracy and F1-score. The results are listed in Tables 6.16 and 6.18 where the methods are ranked based on the obtained z-values. Holm post-hoc test rejects the hypothesis of equivalence for the methods with $p < \alpha/i$. It is clearly seen that DTA-LS-SVMs is at least comparable to LS-SVMs and SVMs on balanced datasets in terms of accuracy;

and is comparable to LS-SVMs and statistically better than SVMs in terms of F1-score.

The results of Friedman test (2) in terms of F1-score are shown in Table 6.19. The results show that there are significant differences between iDTA-LS-SVMs and the other comparative methods. Holm post-hoc test results are listed in Table 6.20. iDTA-LS-SVMs statistically outperforms the other methods on the imbalanced datasets.

In summary, the proposed model and its imbalanced version are at least comparable to LS-SVMs and SVMs or even outperform them in terms of accuracy and/or F1-score.

Table 6.15: AVERAGE RANKINGS OF DTA-LS-SVMS AND THE COMPARATIVE METHODS ON BALANCED DATASETS IN TERMS OF ACCURACY (p-VALUE= 0.049787)

Methods	Ranking
DTA-LS-SVMs	1
LS-SVMs	2.5
SVMs	2.5

Table 6.16: HOLM POST-HOC COMPARISON RESULTS FOR DTA-LS-SVMS AND THE OTHER METHODS IN TERMS OF ACCURACY WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	LS-SVMs	2.12132	0.033895	0.025
1	SVMs	2.12132	0.033895	0.05

Table 6.17: AVERAGE RANKINGS OF DTA-LS-SVMS AND THE COMPARATIVE METHODS ON BALANCED DATASETS IN TERMS OF F1-SCORE (p-VALUE= 0.038774)

Methods	Ranking
DTA-LS-SVMs	1
LS-SVMs	2.25
SVMs	2.75

Table 6.18: HOLM POST-HOC COMPARISON RESULTS FOR DTA-LS-SVMS AND THE OTHER METHODS IN TERMS OF F1-SCORE WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	SVMs	2.474874	0.013328	0.025
1	LS-SVMs	1.767767	0.0771	0.05

Table 6.19: AVERAGE RANKINGS OF iDTA-LS-SVMS AND THE COMPARATIVE METHODS ON IMBALANCED DATASETS IN TERMS OF F1-SCORE (p-VALUE= 0.022371)

Methods	Ranking
iDTA-LS-SVMs	1
LS-SVMs	2.6
SVMs	2.4

Table 6.20: HOLM POST-HOC COMPARISON RESULTS FOR iDTA-LS-SVMS AND THE OTHER METHODS WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	LS-SVMs	2.529822	0.011412	0.025
1	SVMs	2.213594	0.026857	0.05

6.7 Summary

This chapter proposes a novel deep transfer additive LS-SVMs model DTA-LS-SVMs and its imbalanced version iDTA-LS-SVMs to enhance the generalization performance of AK-LS-SVMs. Inspired by the deep stacked architecture and transfer learning, the proposed model stacks multiple AK-LS-SVMs in a chain, where the predicted outputs from the previous module are concatenated with the original data to become the new data input in the higher module. The novelty embodies in model transfer between the adjacent modules to guarantee their consistency. In addition, a proposed fast leave-one-out cross validation strategy for parameter tuning guarantees the advantageous classification performance of the proposed model in circumstances that the regularization parameter in each module can be randomly selected.

The proposed model and its imbalanced version are evaluated on seven public UCI datasets and one real world community health care dataset. Experimental results show that the proposed model can achieve comparatively better classification performances on both balanced and imbalanced datasets at a faster speed, particularly exhibiting potential to be used in the real world health data with class imbalance problems.

Chapter 7

A Deep Cross-output Transfer LS-SVMs Model for Diagnosing Prostate Cancer with Imbalance Data

7.1 Introduction

This chapter proposes a novel deep cross-output knowledge transfer model based on LS-SVMs called DCOT-LS-SVMs to improve the classification capability of LS-SVMs while avoiding the complicated parameter tuning process that occurs in many kernel machines. The proposed model has two significant characteristics: (1) the DCOT-LS-SVMs is inspired by a stacked hierarchical architecture that combines several layer-by-layer LS-SVMs modules. The module in the higher layer has appended features of the predictions from all previous modules; and (2) cross-output knowledge transfer is used to leverage the learned knowledge from the predictions of the adjacent lower module to improve the learning process in the higher module. Model

parameters, such as a trade-off parameter C and a kernel width δ, can be randomly assigned to each module which greatly simplifies the learning process. Moreover, DCOT-LS-SVMs is able to autonomously and quickly decide the extent of the cross-output knowledge transfer between adjacent modules through a fast leave-one-out cross-validation strategy. Additionally, we present an imbalanced version of DCOT-LS-SVMs, called IDCOT-LS-SVMs, given that imbalanced datasets are common in real-world health care scenarios.

The effectiveness of the proposed model is demonstrated through a comparison with traditional SVMs and LS-SVMs on public UCI datasets and a real-world health care application for the diagnosis of prostate cancer.

We must notice that the proposed model in this chapter and the deep transfer additive LS-SVMs model presented in Chapter 6 have the commonality that they both utilize the predicted outcome containing discriminative information from the previous layer(s) to open the manifold structure of the original data to make it more separable. Therefore, essentially both models are trying to extract high-level or deep features so as to enhance classification performance via a stacking design. The novelty of these two proposed models is embodied in the incorporation of it with transfer learning, and the use of the fast leave one-out cross validation strategy for tuning the parameter reflecting the degree of knowledge transfer. The differences between two models can be summarized into three points. First, The former model focuses on output knowledge transfer across domains in adjacent layers while the latter one focuses on model knowledge transfer. Second, the ways of two models to expand the feature space in the higher layers of the deep stacked architecture are different. Third, the former model can randomly assign the model's parameters in every module which greatly simplifies the learning process, while the latter one requires the same kernel

width in each module for the safe use of the additive kernel.

This chapter is organized as follows. Section 7.2 presents the proposed deep cross-output transfer model. Section 7.3 discusses how to extend the proposed model on class imbalance problems. Section 7.4 gives the evaluation on UCI public datasets. Section 7.5 gives a case study on a real-world health care dataset for diagnosis of prostate cancer. Section 7.6 shows the statistical analysis of classification performances. Section 7.7 concludes the chapter.

7.2 Deep Cross-output Transfer LS-SVMs Model

7.2.1 Framework of the Proposed Model

The encompassing framework of the proposed model conforms to a deep stacked architecture. It combines several layers of LS-SVMs modules. The original dataset provides the input data for the first layer to construct a traditional LS-SVMs classifier. Each module from the second layer upwards is a cross-output knowledge transfer LS-SVMs classifier. The input data for these layers comprises the original features and the appended features from all previous layers' prediction outputs. Cross-output knowledge transfer is used to leverage the learned output knowledge from the module in the previous layer to improve the learning process in the current layer. The framework of the cross-output knowledge transfer approach using stacked-structure LS-SVMs is illustrated in Fig. 7.1.

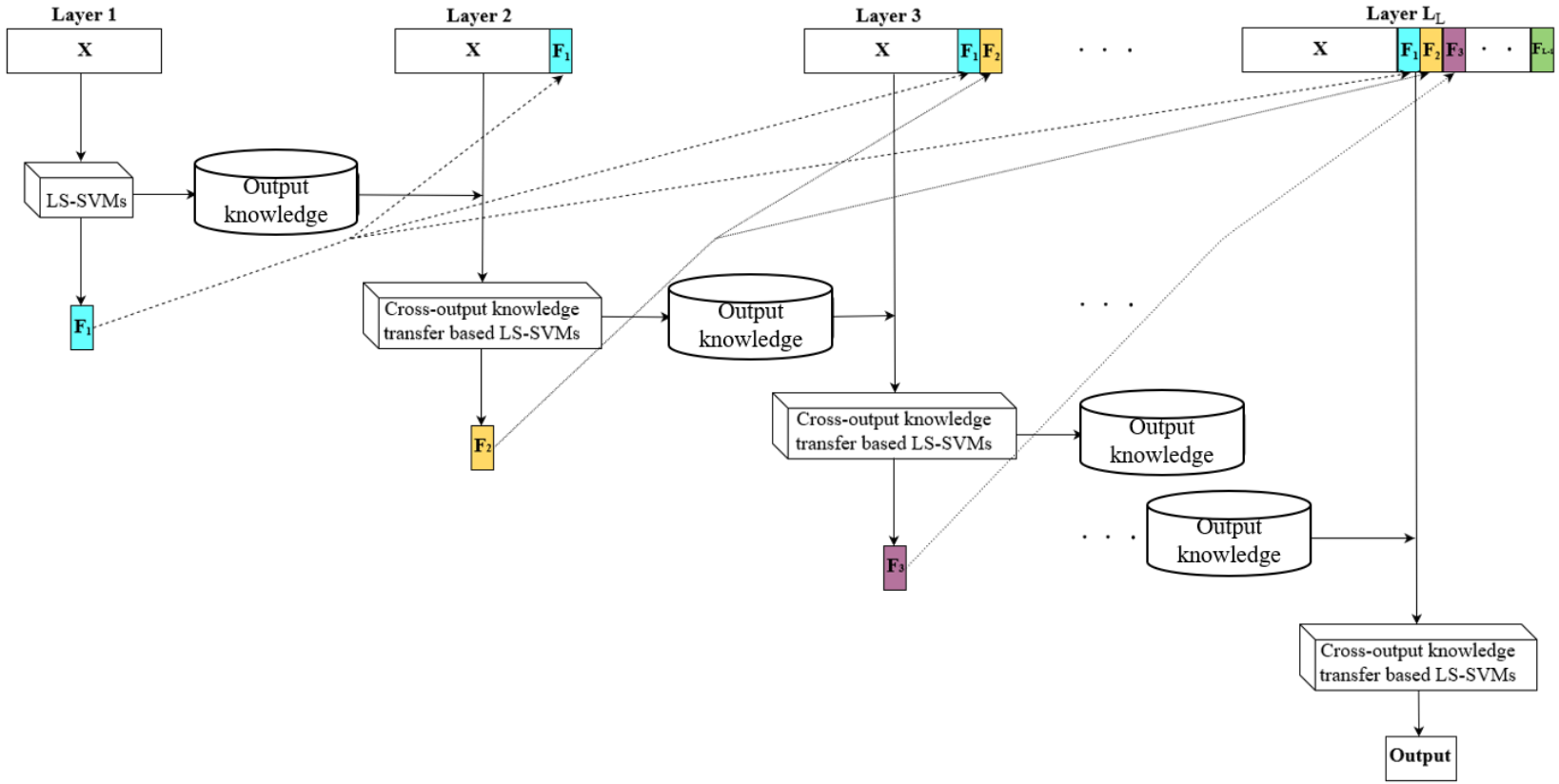


Figure 7.1: THE STACKED ARCHITECTURE AND LEARNING PROCESS IN DCOT-LS-SVMS

7.2.2 Cross-output Knowledge Transfer Under a Stacked Architecture

This subsection explains in detail how the proposed model DCOT-LS-SVMs and its imbalanced version IDCOT-LS-SVMs work.

Given a dataset $\mathbf{D} = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_i, y_i), \cdots, (\vec{x}_N, y_N)\}$, where $\vec{x}_i = (x_1^i, x_2^i, \cdots, x_d^i) \in \mathbf{X} \subset \mathbf{R}^d$ and $y_i \in \mathbf{Y} = \{-1, 1\}$. $\mathbf{X}$ and $\mathbf{Y}$ are the input dataset and output dataset, respectively. Each input $\vec{x}_i$ contains d features, i.e., $f_1, f_2, \cdots, f_d$. In the first layer L_1 of DCOT-LS-SVMs, a traditional LS-SVMs is trained to find an optimal hyperplane $f(\vec{x}) = \vec{w}^T\varphi(\vec{x}) + b$. From the second layer $L_l(l = 2, 3, \cdots, L)$, the data input $\mathbf{X}_l$ is an augmentation of the data input set $\mathbf{X}_{l-1}$ and the predicted output vector from the previous layer. $\mathbf{X}_l$ can be denoted as $\mathbf{X}_{l-1} \oplus \vec{F}_{l-1}$. The output knowledge is embedded across adjacent modules to improve the learning process in the current module. In this scenario, the dataset in the previous module is regarded as a source domain $D_{S(l-1)}$, and the dataset in the current module is regarded as a target domain $D_{(Tl)}$. We postulate that the outputs from adjacent modules retain some similarity. $y_{i(l-1)}, (i = 1, 2, \cdots, N)$ gives the predicted outputs from the $(l-1)$-th module in $D_{S(l-1)}$. $(y_i - \xi_{il}), (i = 1, 2, \cdots, N)$ gives the predicted outputs from the l-th module in D_{Tl}. The goal is to construct a model in D_{Tl} where $(y_i - \xi_{il})$ can remain as similar to the known $y_{i(l-1)}$ as possible. In other words, $\sum_{i=1}^{N}(y_i - \xi_i)y_{i(l-1)}$ should be maximized. A weighting parameter μ_l is used to reflect the influence level of the learned output knowledge from $D_{S(l-1)}$ to that of D_{Tl}. Therefore, the optimization

problem of LS-SVMs is reformulated as

$$\begin{aligned} \min_{\vec{w}_l, b_l} \quad & \frac{1}{2}\vec{w}_l^2 + \frac{C_l}{2}\sum_{i=1}^{N}\xi_{il}^2 - \mu_l \sum_{i=1}^{N}(y_i - \xi_{il})y_{i(l-1)} \\ \text{s.t} \quad & y_i = \vec{w}_l^T\varphi(\vec{x}_{il}) + b_l + \xi_{il}, i = 1, 2, ..., N \end{aligned} \tag{7.1}$$

After derivations, we have the equivalent formulation:

$$\begin{aligned} \min_{\vec{w}_l, b_l} \quad & \frac{1}{2}\vec{w}_l^2 + \frac{C_l}{2}\sum_{i=1}^{N}(\xi_{il} + \frac{\mu_l}{2C_l}y_{(l-1)i})^2 \\ \text{s.t} \quad & y_i = \vec{w}_l^T\varphi(\vec{x}_{il}) + b_l + \xi_{il}, i = 1, 2, ..., N \end{aligned} \tag{7.2}$$

where $\frac{\mu_l}{2C_l}$ represents the influence level of the probabilistic outputs from $D_{S(l-1)}$ to D_{Tl}. We can observe that if μ_l is set to 0 in Eq. (7.2), it becomes the objective function of traditional LS-SVMs. The Lagrangian J_l of Eq. (7.2) is

$$J_l = \frac{1}{2}\vec{w}_l^2 + \frac{C_l}{2}\sum_{i=1}^{N}(\xi_{il} + \frac{\mu_l}{2C_l}y'_{i(l-1)})^2 + \sum_{i=1}^{N}\alpha_{il}(y_i - \vec{w}_l^T\varphi(\vec{x}_{il}) - b_l - \xi_{il}) \tag{7.3}$$

where $\vec{\alpha}_l = (\alpha_{1l}, \alpha_{2l}, ..., \alpha_{Nl}) \in \mathbf{R}^N$ is the vector of all the Lagrangian multipliers. The system of linear equations can be obtained

$$\sum_{j=1}^{N}\alpha_{il}\varphi_l(\vec{x}_{jl})^T\varphi_l(\vec{x}_{il}) + b_l + \frac{\alpha_{il}}{C_l} = y_i + \frac{\mu_l}{2C_l}y'_{i(l-1)} \tag{7.4}$$

Replacing $\varphi_l(\vec{x}_{jl})\varphi_l(\vec{x}_{il})$ using $\mathbf{K}(\vec{x}_{jl}, \vec{x}_{il})$, the above equation can be further written in matrix form as

$$\begin{bmatrix} \mathbf{K}_l + \frac{1}{C_l}\mathbf{N} & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \begin{bmatrix} \vec{\alpha}_l \\ b_l \end{bmatrix} = \begin{bmatrix} \vec{Y} + \frac{\mu_l}{2C_l}\vec{M}_l \\ 0 \end{bmatrix} \tag{7.5}$$

where $\mathbf{N}$ is an identity matrix, $\vec{Y}$ is the output vector of all the samples in the training dataset, and $\vec{M}_l$ is the predicted output vector of these training samples that are obtained from the previous module, i.e., $\vec{M}_l = \left(y_{1(l-1)}, y_{2(l-1)}, \cdots, y_{N(l-1)}\right)^T = \left(\sum_{i=1}^{N} \alpha_{i(l-1)} K_{l-1}(\vec{x}_{i(l-1)}, \vec{x}_{1(l-1)}), \cdots, \sum_{i=1}^{N} \alpha_{i(l-1)} K_{l-1}(\vec{x}_{i(l-1)}, \vec{x}_{N(l-1)})\right)^T$. Obviously, we can see that the kernel functions $\mathbf{K}_{l-1}$ and $\mathbf{K}_l$ do not need to be the same. The kernel function in each module is therefore independent and can be selected randomly without generality. In this study, we use Gaussian kernels with different kernel widths in different modules.

Lastly, the module parameters can be calculated simply by using a matrix inversion:

$$\begin{bmatrix} \vec{\alpha}_l \\ b_l \end{bmatrix} = \mathbf{P}_l \begin{bmatrix} \vec{Y} + \frac{\mu_l}{2C_l} \vec{M}_l \\ 0 \end{bmatrix} \tag{7.6}$$

where $\mathbf{P}_l = \mathbf{H}_l^{-1}$ and $\mathbf{H}_l$ is the first matrix on the left in Eq. (7.5). Once we obtain μ_l, $\vec{\alpha}_l$, $\vec{w}_l$ and b_l can be calculated. We can easily obtain the decision function for the new sample $\vec{x}_t$ (i.e., $\vec{x}_{tl}$ at the l-th layer) as below:

$$\begin{aligned} f_l(\vec{x}_t) &= \vec{w}_l^T \varphi_l(\vec{x}_{tl}) + b_l \\ &= \sum_{i=1}^{N} \alpha_i K_l(\vec{x}_{il}, \vec{x}_{tl}) + b_l \end{aligned} \tag{7.7}$$

and the predicted output vector $\vec{F}_l = (f_l(\vec{x}'_{1l}), f_l(\vec{x}'_{2l}), \cdots, f_l(\vec{x}'_{Nl}))$ in the l-th layer ($l \geq 2$) can be obtained. L layers are added until the accuracy shows no further improvement (i.e., $|\vec{F}_{l+1} - \vec{F}_l| < \epsilon$). Although classification performance increases with the number of modules, an appropriate value of L may lead to over-fitting that heavily distorts the original feature space due to the successive expansion of

the feature space. From extensive experiments, we determined that $L = 3, 4$ or 5 is appropriate for small- and medium-sized datasets. Pseudo-code for the entire learning algorithm for the proposed DCOT-LS-SVMs is presented in Algorithm 7.1. Note that we select the parameters C_l and δ_l from wide interval ranges, i.e., $C_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$ and $\delta_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$ to guarantee diversity between modules in adjacent layers. However, the range of these intervals could be adjusted to suit the situation.

7.2.3 Fast Leave-one-out Cross Validation Strategy

The classification performance of the proposed model depends on the value of the parameter μ_l. The fast leave-one-out cross validation strategy introduced in Chapter 3.2.5 is employed to determine the optimal value of μ_l.

Similarly, by defining $\left[\vec{\alpha}_l^{'T}, b_l^{'}\right]^T = \mathbf{P}_l \left[\vec{y}^T, 0\right]^T$, $\left[\vec{\alpha}_l^{''T}, b_l^{''}\right]^T = \mathbf{P}_l \left[\vec{M}_l^T, 0\right]^T$, and $\vec{\alpha}_l = \vec{\alpha}_l^{'} + \frac{\mu_l}{2C_l}\vec{\alpha}_l^{''}$, the leave-one-out output $\tilde{y}_i$ of the i-th training sample can be represented as

$$\tilde{y}_{il} = y_i - \frac{\alpha_{il}^{'}}{P_{iil}} - \frac{\frac{\mu_l}{2C_l}\alpha_{il}^{''}}{P_{iil}} \tag{7.8}$$

The loss function below is adopted to avoid local minima issues:

$$l(\tilde{y}_{il}, y_i) = |1 - \tilde{y}_{il}y_i|_+ = \left| y_i \frac{\alpha_{il}^{'} - \frac{\mu_l}{2C_l}\alpha_{il}^{''}}{P_{iil}} \right|_+ \tag{7.9}$$

where $|x|_+ = \max\{0, x\}$. Finally, the objective function becomes

$$\begin{aligned} &\sum_{i=i}^{N} l(\tilde{y}_{il}, y_i) \\ &\text{s.t} \quad 0 \leq \mu_l \leq D \end{aligned} \tag{7.10}$$

where D is a constant. This optimization process can be implemented by a projected sub-gradient descent algorithm. The pseudo-code is given in Algorithm 7.2.

7.2.4 Computational Complexity

DCOT-LS-SVMs and IDCOT-LS-SVMs feature fast computation, attribute to the leave-one-out cross validation for parameter tuning under the above stacked architecture. The computational complexity can be represented as $O(N^3 + (L-1)(N^3 + N))$, which contains two parts. The first part $O(N^3)$ represents the computational complexity of the traditional LS-SVM model construction in the first layer. The second part $O((L-1)(N^3+N))$ represents the computational complexity from the second to the L layer. Since the complexity of inverse computation of matrix $\mathbf{P}_l$ for the training set at the l-th layer ($l \geq 2$) is $O(N^3)$ and the complexity of each iteration in Algorithm 7.2 to optimize Eq. (7.10) is $O(N)$, the total computational cost of IDCOT-LS-SVMs becomes $O(N^3 + (L-1)(N^3 + N)) = O(L * N^3 + (L-1) * N)$.

Let us consider the traditional leave-one-out cross-validation strategy for SVMs. Theoretically, the computational complexity to train a SVMs is $O(N^3)$. By using specific speed-up strategies Tsang et al. [2006], the computational complexity can be reduced to $O(N) - O(N^{2.3})$ such that the complexity of the leave-one-out cross validation for SVMs becomes $O(N*N) - O(N*N^{2.3}) = O(N^2) - O(N^{3.3})$. Also, grid search is needed for tuning the generalization parameter $C_l (l = 1, 2, \cdots, L)$ (assuming s_1 grid values) and the kernel width $\sigma_l (l = 1, 2, \cdots, L)$ (assuming s_2 grid values). Therefore the complexity of SVMs becomes $s_1 s_2 O(N^2) - s_1 s_2 O(N^{3.3})$. In general, s_1 and s_2 are normally greater than 3 but the number of layers L in DCOT-LS-SVM is small ($3 \leq L \leq 5$). Therefore, although it seems that the computational complexity of

Algorithm 7.1: Learning algorithm of DCOT-LS-SVMs
Input: training set $\vec{X} = [\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N]$, $\vec{x}_i \in \vec{R}^d$, output set $\vec{Y} = [y_1, y_2, \cdots, y_N]$, $y_i \in \{+1, -1\}$ for binary classification, number of layers L, $l = 1$
Output: The stacked structure of DCOT-LS-SVMs with tuned parameter values
Procedure
Step 1: 1.1 Randomly choose the regularization parameter C_1 and kernel width δ_1 from intervals, i.e., $C_1 \in \{1, 10, 50, 100, 150, 200, 250, 500\}$, $\delta_1 \in \{1, 10, 50, 100, 150, 200, 250, 500\}$. 1.2 Construct the 1st module using the traditional LS-SVMs and obtain $\vec{w}_1$, b_1 and the predicted output vector $\vec{F}_1 = (f_1(\vec{x}_{11}), f_1(\vec{x}_{21}), \cdots, f_1(\vec{x}_{N1}))$. Step 2: For $l = 2 : L$ do 2.1 $\vec{X}_l = \vec{X}_{l-1} \oplus \vec{F}_{l-1}$ 2.2 Randomly choose the regularization parameter C_l and kernel width δ_l from intervals, i.e., $C_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$, $\delta_l \in \{1, 10, 50, 100, 150, 200, 250, 500\}$. 2.3 Construct the lth module by applying 7.1 on $\vec{X}_l$ and obtain μ_l. 2.4 Calculate $\vec{w}_l$ and the predicted output vector $\vec{F}_l = (f_l(\vec{x}_{1l}), f_l(\vec{x}_{2l}), \cdots, f_l(\vec{x}_{Nl}))$ accordingly. Step 3: Calculate $\triangle_F = \|\vec{F}_l - \vec{F}_{l-1}\|_F^2$ Step 4: If $\triangle_F \leq \epsilon$ (a given threshold) End else Step 5: $l = l + 1$ Step 6: Output the stacked structure of the proposed classifier DTO-LS-SVMs with tuned parameter values and the decision function in the L-th module as the final decision function.

Algorithm 7.2: Projected Sub-gradient Descent Algorithm
Input: $\vec{w}_{l-1}$, $\vec{X}_l$, $\vec{Y}$, C_l and kernel width σ_l
Output: μ_l
Procedure
Step 1: Calculate $\mathbf{P}_l$, $\vec{\alpha}_l^{'}$, $\vec{\alpha}_l^{''}$ Step 2: $t = 1$ Step 3: Repeat $\tilde{y}_{il} = y_i - \frac{\alpha_{li}^{'}}{P_{iil}} - \frac{\frac{\mu_l}{2C_l}\alpha_{il}^{''}}{P_{iil}}, i = 1, 2, ..., N$ $d_i \leftarrow \vec{1}\{\tilde{y}_{il} y_i > 0\}, i = 1, 2, ..., N$ $\mu_l \leftarrow \mu_l - \frac{1}{\sqrt{t}} d_i y_i \frac{\alpha_{il}^{''}}{P_{iil}}$ If $\mu_l > D$ then $\mu_l \leftarrow D$ End if $\mu_l \leftarrow \max(\mu_l, 0)$ $t \leftarrow t + 1$ Step 4: Until convergence Step 5: Output μ_l

DCOT-LS-SVMs and IDCOT-LS-SVMs is higher than that of SVMs, our experiments reveal that the actual running time of SVM with grid search is much longer than that of DCOT-LS-SVMs and IDCOT-LS-SVMs.

On the other hand, the computational complexity to train a LS-SVMs is $O(N^3)$. Therefore, when the leave-one-out cross validation with grid search is applied, the complexity is $s_1 s_2 O(N * N^3) = s_1 s_2 O(N^4)$. Also, if we set μ_l in Eq. (7.9) and Eq. (7.10) to 0, it is reduced to the fast leave-one-out cross validation for the traditional LS-SVMs and can be expressed as $s_1 s_2 O(N^3 + N)$. In summary, the proposed DCOT-LS-SVMs and IDCOT-LS-SVMs are advantageous in running speed when compared with the traditional SVMs and LS-SVMs.

7.3 Extension on Class Imbalance Problems

Considering class imbalance problems are very common in heath care prediction, DCOT-LS-SVMs can be extended to an imbalanced version IDCOT-LS-SVMs such that classification with class imbalances can also be solved using the proposed deep cross-output knowledge transfer.

When handling imbalanced datasets, the decision boundary of LS-SVMs tend to get too close to the minority class which needs to be pushed away. One solution is to apply different error costs to the positive and negative classes. Here, a simple cost matrix is introduced to the learning process. The objective function of LS-SVMs is reformulated to handle imbalanced datasets

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}\|\vec{w}\|^2 + \frac{C_1}{N^+}\sum_{i=1}^{N^+}\xi_i^2 + \frac{C_2}{N^-}\sum_{i=N^++1}^{N}\xi_i^2 \\ \text{s.t.} \quad & y_i = \vec{w}^T\varphi(\vec{x}_i) + b + \xi_i, i = 1, 2, \cdots, N^+, N^+ + 1, \cdots, N \end{aligned} \tag{7.11}$$

where N^+ and N^- represent the numbers of positive and negative classes respectively. C_1 and C_2 are two different given constants. When $N^+ > N^-$, C_2 must be bigger than C_1.

Similar to Eq. (7.3) to Eq. (7.6), we can easily find that only $\mathbf{H}_l$ in the first matrix in Eq. (7.5) needs to be modified into

$$\mathbf{H}_l = \begin{bmatrix} \mathbf{K}_l + \mathbf{E}_l & \vec{1} \\ \vec{1}^T & 0 \end{bmatrix} \tag{7.12}$$

where

$$\mathbf{E}_l = \begin{pmatrix} \frac{C_{l1}}{N^+} & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{C_{l1}}{N^+} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{l2}}{N^-} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{C_{l2}}{N^-} \end{pmatrix} \tag{7.13}$$

C_{l1} and C_{l2} have the same roles as C_1 and C_2 in Eq. (7.11) and the remaining derivations remain the same.

7.4 Experiments

In the experiments, the proposed model DCOT-LS-SVMs and its imbalanced version IDCOT-LS-SVMs were evaluated on the public UCI datasets. Their performances are compared with those using the traditional LS-SVMs and SVMs. Non-parametric statistical tests are used to check if the differences observed are significant or not. The experiments are implemented in 64-bit MATLAB R2014a on a computer with an Intel Core i5-6300 2.4 GHz CPU and 8.00GB RAM.

7.4.1 UCI Datasets

To evaluate the classification performances of the proposed methods, we adopted seven standard UCI datasets, including four balanced and three imbalanced which are summarized in Table 7.1.

Table 7.1: UCI DATASETS DESCRIPTION

Types	Datasets	Sample sizes	Features	Class(%)
Imbalanced	*BREAST*	683	9	65.52 34.48
	PIMA	768	8	65.02 34.98
	ILPD	579	10	71.50 28.50
Balanced	*AUS*	690	14	44.50 55.50
	DIABETIC	1151	19	53.08 46.92
	CREDIT	653	15	45.33 54.67
	MAMMOGRAPHIC	830	5	48.55 51.45

7.4.2 Parameter Setup

For DCOT-LS-SVMs and IDCOT-LS-SVMs, Gaussian kernel parameters were randomly selected according to Algorithm 7.1. Different kernels were used for LS-SVMs and SVMs for each adopted dataset. Here we only display the best experimental results using the Gaussian kernel. Kernel width δ was selected from $\{0.1, 1, 5, 10, 20, 50, 100, 150, 200\}$. Regularization parameter C was selected from $\{1, 10, 50, 100, 150, 200, 250, 500\}$. For DCOT-LS-SVMs and IDCOT-LS-SVMs, the number of modules L was set to 3, 4 or 5 since the size of the adopted datasets was small or medium. The value of D in Algorithm 7.2 was set to 1, and the value of ϵ in Algorithm 7.1 was set to 0.1.

7.4.3 Experimental Results Analysis

DCOT-LS-SVMs were evaluated on four balanced UCI datasets and the results, in terms of accuracy and AUC, are shown in Table 7.2. IDCOT-LS-SVMs were evaluated on three imbalanced UCI datasets and the results, in terms of AUC, are shown in

Table 7.3. We can observe that the proposed DCOT-LS-SVMs achieved the best classification performances on both balanced and imbalanced datasets compared with the traditional LS-SVMs and SVMs.

In addition, the running time of all the methods on each dataset are shown in Table 7.11. We can see that DCOT-LS-SVMs and IDCOT-LS-SVMs, under the stacked architecture, spent the minimum running time to train the models when compared with the other methods that follows a shallow architecture. Here, the running time of DCOT-LS-SVMs and IDCOT-LS-SVMs are defined as the accumulation of the running time of each module and the time used to optimize parameter $\mu_l (l = 2, 3, \cdots, L)$ at the l-th layer. The experimental results show that the proposed approaches have superior advantages in speed. This is each module's parameters can be randomly assigned and the optimal value of μ_l can be found autonomously and efficiently using the fast leave-one-out cross validation strategy. LS-SVMs and SVMs however took much longer time for model selection (C and δ) by grid search.

To further verify the effectiveness and robustness of the proposed model, we performed one more experiment as follows. We added white Gaussian noise of different levels (5%, 8%, 12%) on the adopted UCI datasets. Tables 7.4, 7.5, 7.6, 7.8, 7.9, 7.10 and 7.7 show the corresponding experimental results. The noises indeed influenced the performance of all the methods, which resulted in steady decrease in performance metrics with increasing noise levels. Experimental results show that DCOT-LS-SVMs and IDCOT-LS-SVMs still remain advantageous over traditional LS-SVMs and SVMs with the noisy data, demonstrating the robustness of the proposed model.

Table 7.2: PERFORMANCE RESULTS ON BALANCED DATASETS

Datasets	Metrics	Performances					
		DCOT-LS-SVMs		LS-SVMs		SVMs	
		training	testing	training	testing	training	testing
AUS	Accuracy	0.8782±0.0075	**0.8750±0.0143**	0.8328±0.0053	0.8332±0.0677	0.8589±0.0075	0.8409±0.0174
	AUC	0.8803±0.0112	**0.8847±0.0175**	0.8452±0.0854	0.8428±0.0814	0.8531±0.0110	0.8433±0.0260
DIABETIC	Accuracy	0.8102±0.0113	**0.7494±0.0208**	0.7391±0.0221	0.7127±0.0242	0.7275±0.0068	0.7136±0.0194
	AUC	0.7670±0.0119	**0.7526±0.0200**	0.7292±0.0279	0.7163±0.0237	0.7348±0.0124	0.7059±0.0155
CREDIT	Accuracy	0.8821±0.0106	**0.8786±0.0161**	0.8611±0.0138	0.8357±0.0203	0.8678±0.0087	0.8577±0.0230
	AUC	0.8886±0.0087	**0.8849±0.0141**	0.8595±0.0456	0.8433±0.0455	0.8638±0.0091	0.8680±0.0193
MAMMOGRAPHIC	Accuracy	0.8384±0.0048	**0.8333±0.0095**	0.8155±0.0103	0.8088±0.0313	0.8189±0.0116	0.8016±0.0349
	AUC	0.8383±0.0048	**0.8344±0.0094**	0.8253±0.0085	0.8113±0.0235	0.8137±0.0164	0.8017±0.0339

Table 7.3: PERFORMANCE RESULTS ON IMBALANCED UCI DATASETS

Datasets	Performances					
	IDCOT-LS-SVMs		LS-SVMs		SVMs	
	training	testing	training	testing	training	testing
BREAST	0.9705±0.0041	**0.9677±0.0153**	0.9655±0.0073	0.9523±0.0173	0.9631±0.0056	0.9512±0.0121
PIMA	0.7328±0.0129	**0.7248±0.0247**	0.7065±0.0380	0.6878±0.0186	0.7872±0.0114	0.7128±0.0254
ILPD	0.6759±0.1313	**0.5777±0.0443**	0.6053±0.1120	0.5520±0.0569	0.6575±0.0187	0.5358±0.0323

Table 7.4: PERFORMANCE RESULTS ON THE *Aus* DATASET

Aus		Performances					
Noise level	Metrics	DCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	Accuracy	0.8724±0.0373	**0.8724±0.0191**	0.8496±0.0029	0.8279±0.0582	0.8614±0.0102	0.8399±0.0101
	AUC	0.8695±0.0088	**0.8680±0.0158**	0.8411±0.0222	0.8465±0.0349	0.8676±0.0106	0.8606±0.0212
8%	Accuracy	0.8769±0.0073	**0.8644±0.0237**	0.8488±0.0117	0.8260±0.0169	0.8618±0.0220	0.8319±0.0306
	AUC	0.8762±0.0127	**0.8642±0.0165**	0.8393±0.0450	0.8642±0.0165	0.8675±0.0108	0.8601±0.0262
12%	Accuracy	0.8769±0.0117	**0.8572±0.0237**	0.8475±0.0630	0.8168±0.0373	0.8612±0.0191	0.8251±0.0169
	AUC	0.8801±0.0080	**0.8607±0.0285**	0.8375±0.0511	0.8168±0.0503	0.8612±0.0139	0.8502±0.0232

Table 7.5: PERFORMANCE RESULTS ON THE *Diabetic* DATASET

Diabetic		Performances					
Noise level	Metrics	DCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	Accuracy	0.7486±0.0109	**0.7382±0.0449**	0.7042±0.0202	0.6908±0.0286	0.7289±0.0175	0.7107±0.0061
	AUC	0.7507±0.0089	**0.7401±0.0292**	0.7065±0.0192	0.6936±0.0190	0.7322±0.0112	0.7141±0.0216
8%	Accuracy	0.7111±0.0193	**0.7312±0.0102**	0.7046±0.0087	0.6815±0.0143	0.7252±0.0061	0.7064±0.0081
	AUC	0.6997±0.0490	**0.7329±0.0214**	0.7059±0.0182	0.6836±0.0432	0.7272±0.0118	0.7087±0.0154
12%	Accuracy	0.7304±0.0121	**0.7197±0.0347**	0.6887±0.0368	0.6656±0.0143	0.7174±0.0017	0.7012±0.0224
	AUC	0.7325±0.0152	**0.7230±0.0279**	0.6889±0.0296	0.6672±0.0317	0.7188±0.0119	0.7042±0.0159

Table 7.6: PERFORMANCE RESULTS ON THE *Credit* DATASET

Credit		Performances					
Noise level	Data sets	DCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	Accuracy	0.8799±0.0139	**0.8633±0.0108**	0.8623±0.0092	0.8467±0.018	0.8685±0.0101	0.8612±0.0011
	AUC	0.8825±0.0079	**0.8691±0.0201**	0.8504±0.0357	0.8431±0.0583	0.8748±0.0086	0.8687±0.0218
8%	Accuracy	0.8836±0.0061	**0.8612±0.0108**	0.8608±0.0618	0.8362±0.0613	0.8662±0.0030	0.8602±0.0072
	AUC	0.8856±0.0071	**0.8665±0.0275**	0.8581±0.0366	0.8358±0.0354	0.8720±0.0072	0.8674±0.0167
12%	Accuracy	0.8832±0.0139	**0.8602±0.0396**	0.8420±0.0061	0.8337±0.0072	0.8656±0.0015	0.8591±0.0036
	AUC	0.8846±0.0122	**0.8654±0.0259**	0.8355±0.0385	0.8252±0.0343	0.8713±0.0105	0.8657±0.0242

Table 7.7: PERFORMANCE RESULTS ON THE *Mammographic* DATASET

Mammographic		Performances					
Noise level	Metrics	DCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	Accuracy	0.8306±0.0110	**0.8281±0.0298**	0.8294±0.0126	0.8200±0.0222	0.7958±0.0122	0.8020±0.032
	AUC	0.8309±0.0107	**0.8279±0.0303**	0.8299±0.0123	0.8215±0.0215	0.7907±0.0218	0.7967±0.0064
8%	Accuracy	0.8260±0.0232	**0.8257±0.0231**	0.8337±0.0086	0.8164±0.0228	0.7953±0.0093	0.7947±0.0168
	AUC	0.8420±0.0183	**0.8420±0.0182**	0.8346±0.0087	0.8173±0.0221	0.7964±0.0091	0.7963±0.0179
12%	Accuracy	0.8306±0.0110	**0.8281±0.0298**	0.8230±0.0105	0.8152±0.0304	0.7984±0.0062	0.7899±0.0224
	AUC	0.8355±0.0099	**0.8221±0.0258**	0.8239±0.0101	0.8161±0.0309	0.7973±0.0091	0.7967±0.0064

Table 7.8: PERFORMANCE RESULTS ON THE *Breast* DATASET

Breast		Performances					
Noise level	Metric	IDCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	AUC	0.9772±0.0043	**0.9671±0.0130**	0.9559±0.010	0.9571±0.0342	0.9559±0.0100	0.9570±0.0342
8%		0.9755±0.0051	**0.9628±0.0167**	0.9550±0.0137	0.9550±0.0163	0.9545±0.0137	0.9549±0.0163
12%		0.9723±0.0032	**0.9608±0.0144**	0.9407±0.0434	0.9310±0.0342	0.9407±0.0434	0.9309±0.0342

Table 7.9: PERFORMANCE RESULTS ON THE *Pima* DATASET

Pima		Performances					
Noise level	Metric	IDCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	AUC	0.7266±0.0233	**0.7134±0.0370**	0.7015±0.0562	0.6903±0.0376	0.7802±0.0158	0.7082±0.0266
8%		0.7222±0.0146	**0.7107±0.0302**	0.6882±0.0485	0.6900±0.0316	0.7756±0.0204	0.7020±0.0101
12%		0.7207±0.0124	**0.7015±0.0214**	0.6882±0.0562	0.6877±0.0376	0.7720±0.164	0.6899±0.0222

Table 7.10: PERFORMANCE RESULTS ON THE *ILPD* DATASET

ILPD		Performances					
Noise level	Metric	IDCOT-LS-SVMs		LS-SVMs		SVMs	
		Training	Testing	Training	Testing	Training	Testing
5%	AUC	0.6906±0.0487	**0.5692±0.0454**	0.6661±0.1496	0.5468±0.0189	0.6337±0.0533	0.5572±0.0321
8%		0.6740±0.0889	**0.5658±0.0433**	0.5871±0.1102	0.5312±0.0370	0.6481±0.0466	0.5550±0.0355
12%		0.6696±0.0963	**0.5606±0.0284**	0.5750±0.0797	0.5180±0.0189	0.6601±0.0267	0.5540±0.0263

Table 7.11: TRAINING AND TESTING TIME (SECONDS) ON SEVEN UCI DATASETS

Datasets		Running time		
		DCOT-LS-SVMs	LS-SVMs	SVMs
Imbalanced	*BREAST*	3.387	7768.2	21597
	PIMA	4.446	10439.3	30664
	ILPD	3.058	6190.7	15496
balanced	*AUS*	5.376	11306	25908
	DIABETIC	14.439	71870	130696
	CREDIT	4.018	10871	19902
	MAMMOGRAPHIC	3.326	8784.7	26390

7.5 A Case Study on a Real World Prostate Cancer Dataset

This section describes a real-world case study with the application of the proposed model and its imbalanced version in detail. In this case study, DCOT-LS-SVMs and IDOCT-LS-SVMs were used to detect prostate cancer based upon the findings from transrectal ultrasound (TRUS)-guided biopsy, digital rectal examination (DRE), prostate-specific antigen (PSA) level and risk factors in Chinese population.

7.5.1 Data Collection

The adopted dataset includes 1230 records of Chinese men who had undergone TRUS-guided prostate biopsy, which is retrieved from a TRUS-guided prostate biopsy database in a hospital in Hong Kong. The clinicopathological data include age, PSA level, DRE finding, TRUS prostate volume and TRUS finding. They are indeed the risk factors in several existing statistical diagnostic models that are used to predict the outcome of biopsy, such as the European Randomized Study of Screening for Prostate Cancer (ERSPC) risk calculator Kranse et al. [2008] and the Prostate Cancer Prevention Trial (PCPT) risk calculator Ankerst et al. [2014].

In our dataset, continuous variables such as age and PSA level were represented as mean values with standard deviations. Categorical variables were represented as percentages. For example, the percentage difference in men with abnormal DRE versus men with normal DRE was calculated. Table 7.12 list the baseline characteristics of the cohort.

The purpose of the case study is to diagnose prostate cancer based on the findings from TRUS-guided biopsy, DRE finding, PSA level and other diagnostic factors in Chinese population. The proposed approaches are used to predict the diagnostic outcome, i.e., whether a patient has prostate cancer or not.

It can be observed from the last row in Table 7.12 that there is a class imbalance issue in the dataset. IDCOT-LS-SVMs were therefore applied for model construction and the generalization performances were compared with those using DCOT-LS-SVMs, LS-SVMs and SVMs. The number of layers of the IDCOT-LS-SVMs and DCOT-LS-SVMs were set to 3.

7.5.2 Results Analysis

We can see from the experimental results given in Table 7.13 that IDCOT-LS-SVMs had the best classification performance among all the methods (accuracy = 0.9290 and AUC = 0.8138), proving that it is effective for handling imbalanced datasets and demonstrates better generalization performance. Also, it is shown that although DCOT-LS-SVMs is primarily developed for handling balanced datasets, it still outperformed the traditional LS-SVMs and SVMs in both tasks in terms of accuracy and AUC. We believe that such advances on the generalization performance are caused by the stacked structure and the embedded knowledge transfer learning.

7.5.3 Contribution

Class imbalance is a common issue in health datasets as uneven distribution of classes is not unusual, e.g. normal cases outweighing diseased or rare cases. This can lead to a very poor prediction outcome on the minority class when traditional classification methods are adopted. On the other hand, to collect large volumes of data from patients is often very expensive and time-consuming due to the low proportion of cancer cases in screening, privacy concerns and more. Therefore it is impractical to achieve more balanced class distributions by adding patient samples. Using the proposed approach IDCOT-LS-SVMs, generalization performance on imbalanced datasets can be enhanced such that more reliable prediction can be obtained to assist doctors in making prostate cancer diagnosis. IDCOT-LS-SVMs also greatly improve the effectiveness of imbalanced health care datasets like the prostate cancer dataset so as to avoid the waste of data.

7.6 Statistical Analysis

To evaluate the statistical significance of the difference in performance observed from the above experiments, we carried out the Friedman ranking test followed by the Holm Post-Hoc test on the classification results of the UCI datasets with no noise and the prostate cancer dataset.

We conducted two Friedman ranking tests. The first test was to evaluate whether there were differences in classification performance between DCOT-LS-SVMs and the comparative methods on four balanced UCI datasets and the prostate cancer dataset in terms of accuracy and AUC. The second test was to evaluate if there were differences in classification performance between IDCOT-LS-SVMs and the comparative methods

Table 7.12: BASELINE CHARACTERISTICS OF THE COHORT

	Value	Percentage
Total number of patients	1230	
Number and percentage of patients with respect to		
PSA level (ng ml^{-1})		
<4	144	11.71
4-10	662	53.82
10.1-20	231	18.78
20.1-50	91	7.40
>50	102	8.29
Age(year, mean±s.d.)	67±8	
Estimated prostate volume on TRUS (ml, mean±s.d.)	52.30±26.43	
PSA level (ng ml^{-1})	42.45±274.26	
DRE (number of patients)		
Normal	993	80.73
Abnormal	237	19.27
TRUS finding (number of patients)		
Normal	1125	91.46
Abnormal	105	8.54
Overall prostate cancer detection rate		24.57

Table 7.13: PERFORMANCE RESULTS ON THE PROSTATE CANCER DATASET

Methods	Data sets	Performances	
		Accuracy	AUC
IDCOT-LS-SVMs	training	0.9549±0.0052	0.8524±0.0160
	testing	**0.9290±0.0181**	**0.8138±0.0259**
DCOT-LS-SVMs	training	0.9361±0.0028	0.8515±0.0118
	testing	0.9133±0.0027	0.7913±0.0323
LS-SVMs	training	0.8977±0.0055	0.8110±0.0354
	testing	0.8957±0.0129	0.7539±0.0460
SVMs	training	0.9055±0.0088	0.8064±0.0279
	testing	0.8962±0.0227	0.7747±0.0226

Table 7.14: TRAINING AND TESTING TIME (SECONDS) ON THE PROSTATE CANCER DATASET

Methods	IDCOT-LS-SVMs		DCOT-LS-SVMs		LS-SVMs		SVMs	
Data set	training	testing	training	testing	training	testing	training	testing
Running time	5.837	5.437	4.673	4.217	4950.6	2.398	1157.8	4.183

Table 7.15: AVERAGE RANKINGS OF DCOT-LS-SVMS AND THE COMPARATIVE METHODS IN TERMS OF ACCURACY (p=0.022371)

Methods	Ranking
DCOT-LS-SVMs	1
LS-SVMs	2.6
SVMs	2.4

on three imbalanced UCI datasets and the prostate cancer dataset in terms of AUC.

The ranking results of the first Friedman test in terms of accuracy and AUC are shown in Tables 7.15 and 7.17, respectively. Statistical results reveal that there are significant differences in classification performance between DCOT-LS-SVMs and the other comparative methods. Then we conducted the Holm Post-Hoc tests to compare the top-ranked DCOT-LS-SVMs with LS-SVMs and SVMs in terms of accuracy and AUC. The results are presented in Tables 7.16 and 7.18 where the methods are ranked according to the obtained z-values obtained. The results in these tables show that DCOT-LS-SVMs statistically outperforms SVMs and LS-SVMs on the five datasets in terms of accuracy and AUC respectively.

The ranking results of the second Friedman test are shown in Table 7.19. The results reveal that there are significant differences between IDCOT-LS-SVMs and the other comparative methods. Then we conducted the Holm Post-Hoc test to compare the top-ranked IDCOT-LS-SVMs with LS-SVMs and SVMs. The results in Table 7.20 show that IDCOT-LS-SVMs is at least comparable to LS-SVMs; and statistically outperforms SVMs on the four datasets in our experiments.

In summary, the proposed methods DCOT-LS-SVMs and IDCOT-LS-SVMs are at least comparable to or even better than LS-SVMs and SVMs in terms of accuracy and/or AUC with the much faster learning speed.

Table 7.16: HOLM POST-HOC COMPARISON RESULTS FOR DCOT-LS-SVM AND THE OTHER METHODS IN TERMS OF ACCURACY WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	LS-SVMs	2.529822	0.011412	0.025
1	SVMs	2.213594	0.026857	0.05

Table 7.17: AVERAGE RANKINGS OF DCOT-LS-SVMS AND THE COMPARATIVE METHODS IN TERMS OF AUC ($p = 0.022371$)

Methods	Ranking
DCOT-LS-SVMs	1
LS-SVMs	2.4
SVMs	2.6

Table 7.18: HOLM POST-HOC COMPARISON RESULTS FOR DCOT-LS-SVMS AND THE OTHER METHODS IN TERMS OF AUC WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	SVMs	2.529822	0.011412	0.025
1	LS-SVMs	2.213594	0.026857	0.05

Table 7.19: AVERAGE RANKINGS OF IDCOT-LS-SVMS AND THE COMPARATIVE METHODS IN TERMS OF AUC ($p = 0.038774$)

Methods	Ranking
IDCOT-LS-SVMs	1
LS-SVMs	2.25
SVMs	2.75

Table 7.20: HOLM POST HOC COMPARISON RESULTS FOR IDCOT-LS-SVMS AND THE OTHER METHODS IN TERMS OF AUC WITH $\alpha = 0.05$

i	Methods	z-value	p-value	$Holm = \alpha/i$
2	SVMs	2.474874	0.013328	0.025
1	LS-SVMs	1.767767	0.0771	0.05

7.7 Summary

This chapter proposes a deep cross-output knowledge transfer LS-SVMs model DCOT-LS-SVMs and its imbalanced version IDCOT-LS-SVMs to improve the classification performance of LS-SVMs on both balanced and imbalanced datasets. Moreover, the proposed model can effectively avoid complicated process of parameter tuning of C and δ, which significantly simplifies the learning process. In addition, grounded on LS-SVMs, DCOT-LS-SVMs and IDCOT-LS-SVMs can rapidly determine how much output knowledge to transfer between modules using a fast leave-one-out cross validation strategy. Compared with the traditional SVMs and LS-SVMs, DCOT-LS-SVMs and IDCOT-LS-SVMs exhibit comparable or even better classification performances with much faster learning speed and are more robust against different levels of noise. The case study illustrates that the proposed model is beneficial for solving real world health care prediction problems with imbalance data.

Chapter 8

Conclusion and Future Work

This chapter concludes the thesis and provides the future research directions for this topic.

8.1 Conclusions

In this thesis, advanced AI techniques are leveraged as a lens to explore the health data for prediction. A serial of prediction models are customized to improve the accuracy of prediction with the consideration of characteristic problems within the health data, including small sample size, missing data and class imbalances. The findings of this study are summarized as follows:

The output-based transfer LS-SVMs model transfers the probabilistic output knowledge from the existing prediction model or on-line tool to the current interest of domain for classification with insufficient data. The experimental evaluation demonstrate the effectiveness of the proposed model in handling small sample size problem in medical prognosis by achieving better accuracy than other methods. The output-based transfer LS-SVMs model require no prior knowledge of the modeling details and the training data of the existing model or on-line tool, which pose no

challenges to the real world health care applications.

The novel additive LS-SVMs model is to handle classification with missing data. The key feature of the model is evaluating the influences on the classification error caused by missing features using a fast leave-one-out cross validation strategy when performing classification. That is, the proposed model work readily with missing data instead of preprocessing them. Moreover, knowledge of the influence can provide the guidance for the health professionals to further improve the data collection process.

The transfer-based additive LS-SVMs model follows the core idea of the additive LS-SVMs model for tackling missing data. The main difference is that the proposed model is trained from a transfer learning perspective to minimize the disagreement between the complete portion of the dataset and the whole dataset with missing data. In addition,the proposed model concentrates on the influence levels of incomplete instances and thus can be used as an alternative way for data cleaning to guarantee data quality.

The deep transfer additive LS-SVMs model DTA-LS-SVMs and its imbalanced version iDTA-LS-SVMs are to improve the prediction performance on balanced and imbalanced datasets. It uses a hierarchical architecture to unfold the manifold of the original data space in a stacked way to enhance the learning process and model transfer to guarantee the consistency between adjacent modules. iDTA-LS-SVMs especially show good accuracy of prediction to handle class imbalance problems in the real world application.

Similarly, the deep cross-output transfer LS-SVMs model DCOT-LS-SVMs and its imbalanced version IDCOT-LS-SVMs uses the deep architecture to enhance the prediction performance on balanced and imbalanced datasets. The novelty of the proposed model lies in that it is the output knowledge to be leveraged across adjacent

modules to guarantee the consistency, and the kernels in each modules are therefore independent which greatly reduces the time for model selection. The experimental results demonstrate that the proposed model exhibit good prediction performances with much faster learning speed and are robust against different levels of noise compared to other methods. The case study exhibits the feasibility of using the IDCOT-LS-SVMs for dealing with the class imbalance problem in the cancer diagnosis.

Each of the proposed prediction model has been applied to a real world health care application and achieved an overall improvement in the predictive performance. Moreover, the characteristic problems within health care data, such as insufficient data, missing data and class imbalance, are particularly taken into account. They can be well handled simultaneously when performing classification instead of being tackled separately in the data pre-processing stage, which provide convenience to use. These real world case studies are exemplars to demonstrate the benefits of using advanced machine learning models in health care, hereby promoting practical applications in clinical practice in future.

8.2 Future Study

This thesis raises a number of opportunities for future research in terms of feasibility validation and theory development.

(a) Our main aim in this study is to transform AI techniques to perform advanced health care predictive analytics. Accordingly, we have proposed five prediction models with special attention to the characteristic health data problem in practice. In future, we shall apply these models to various health applications for clinical practice to assist practitioners in diagnosis and treatment planning with more confidence.

(b) The proposed models are based on the LS-SVMs framework. In future the core ideas of these models can be extended to different frameworks in the family of kernel ridge regression, such as ELMs for health data analytics.

(c) The proposed transfer learning based models only leverage knowledge from a single source domain. This motivates us to extend them into multiple source domain transfer in future. For example, for the output-based transfer LS-SVMs model presented in Chapter 3, the output knowledge can be learned and integrated from several existing prediction models or on-line tools to facilitate the learning process on the current interest of domain.

(d) The proposed models are supervised learning algorithms that assume the labels of the dataset are always available. In future, this study can also be extended in refining the proposed models to learn from unlabeled data or noisy labels, which has great practical significance in health care.

(e) More challenging situations may occur in the health data analytics, such as dynamic and data-intensive environments. How to upgrade our proposed models to specifically deal with the associated issues is worthy to be further investigated.

Bibliography

Zubair Afzal, Marjolein Engelkes, Katia Verhamme, Hettie M Janssens, Miriam CJM Sturkenboom, Jan A Kors, and Martijn J Schuemie. Automatic generation of case-detection algorithms to identify children with asthma from large electronic health record databases. *Pharmacoepidemiology and Drug Safety*, 22(8):826–833, 2013. 1

Ruben Amarasingham, Billy J Moore, Ying P Tabak, Mark H Drazner, Christopher A Clark, Song Zhang, W Gary Reed, Timothy S Swanson, Ying Ma, and Ethan A Halm. An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. *Medical Care*, 48(11): 981–988, 2010. 2

Fabrizio Angiulli. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 25–32. ACM, 2005. 29

Donna P Ankerst, Josef Hoefler, Sebastian Bock, Phyllis J Goodman, Andrew Vickers, Javier Hernandez, Lori J Sokoll, Martin G Sanda, John T Wei, Robin J Leach, et al. Prostate cancer prevention trial risk calculator 2.0 for the prediction of low-vs high-grade prostate cancer. *Urology*, 83(6):1362–1368, 2014. 151

PierFrancesco Bassi, Emilio Sacco, Vincenzo De Marco, Maurizio Aragona, and

Andrea Volpe. Prognostic accuracy of an artificial neural network in patients undergoing radical cystectomy for bladder cancer: a comparison with logistic regression analysis. *BJU International*, 99(5):1007–1012, 2007. 12, 13

Gustavo EAPA Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6): 519–533, 2003. 26

Rukshan Batuwita and Vasile Palade. Class imbalance learning methods for Support Vector Machines. 2013. 115

Vahid Behbood, Jie Lu, and Guangquan Zhang. Fuzzy refinement domain adaptation for long term prediction in banking ecosystem. *IEEE Transactions on Industrial Informatics*, 10(2):1637–1646, 2014. 24

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 552–560, 2013. 108

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007. 23

Rich Caruana. Multitask learning. In *Learning to Learn*, pages 95–133. Springer, 1998. 25

Gerardo Casañola-Martin, Teresa Garrigues, Marival Bermejo, Isabel González-Álvarez, Nam Nguyen-Hai, Miguel Ángel Cabrera-Pérez, Huong Le-Thi-Thu, et al.

Exploring different strategies for imbalanced adme data problem: case study on caco-2 permeability modeling. *Molecular Diversity*, 20(1):93–109, 2016. 30

James WF Catto, Derek A Linkens, Maysam F Abbod, Minyou Chen, Julian L Burton, Kenneth M Feeley, and Freddie C Hamdy. Artificial intelligence in predicting bladder cancer outcome. *Clinical Cancer Research*, 9(11):4172–4177, 2003. 10

Gavin C Cawley. Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In *International Joint Conference on Neural Networks*, pages 1661–1668. IEEE, 2006. 16

ESY Chan, SKH Yip, SM Hou, HY Cheung, WM Lee, and CF Ng. Age, tumour stage, and preoperative serum albumin level are independent predictors of mortality after radical cystectomy for treatment of bladder cancer in Hong Kong Chinese. *Hong Kong Medical Journal*, 19(5):400–406, 2013. 49

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 29

Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1): 1–6, 2004. 28

Gal Chechik, Geremy Heitz, Gal Elidan, Pieter Abbeel, and Daphne Koller. Max-margin classification of incomplete data. In *Advances in Neural Information Processing Systems*, pages 233–240, 2006. 27

Lynda Chin, Jannik N Andersen, and P Andrew Futreal. Cancer genomics: from

discovery science to personalized medicine. *Nature Medicine*, 17(3):297–303, 2011a. 2

Lynda Chin, William C Hahn, Gad Getz, and Matthew Meyerson. Making sense of cancer genomic data. *Genes & Development*, 25(6):534–555, 2011b. 3

Kup-Sze Choi, Rebecca KP Wai, and Esther YT Kwok. Healthcare information system: a facilitator of primary care for underprivileged elderly via mobile clinic. In *Proc. 1st Int. Conf. Smart Health*, pages 107–112, Beijing, China, 2013. 66

Murat Çınar, Mehmet Engin, Erkan Zeki Engin, and Y Ziya Ateşçi. Early prostate cancer diagnosis by using artificial neural networks and support vector machines. *Expert Systems with Applications*, 36(3):6357–6361, 2009. 15

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 14

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on Information Theory*, 13(1):21–27, 1967. 18

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977. 26

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006. 88, 129

Li Deng and Dong Yu. Deep convex net: A scalable architecture for speech pattern classification. In *Proc. 12th Ann. Int. Conf. Speech Comm. Assoc.*, pages 2285–2288, Florence, Italy, 2011. 30, 31

Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tur. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *Proc. 4th IEEE Workshop Spoken Language Tech.*, pages 210–215, Miami, FL, USA, 2012. 31

Zhaohong Deng, Yizhang Jiang, Kup-Sze Choi, Fu-Lai Chung, and Shitong Wang. Knowledge-leverage-based tsk fuzzy system modeling. *IEEE Transactions on Neural Networks and Learning Systems*, 24(8):1200–1212, 2013. 24

Zhaohong Deng, Yizhang Jiang, Hisao Ishibuchi, Kup-Sze Choi, and Shitong Wang. Enhanced knowledge-leverage-based tsk fuzzy system modeling for inductive transfer learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):11, 2016. 24

A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091, 2006a. 26

A Rogier T Donders, Geert JMG van der Heijden, Theo Stijnen, and Karel GM Moons. Review: a gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091, 2006b. 26

Jacques Donzé, Drahomir Aujesky, Deborah Williams, and Jeffrey L Schnipper. Potentially avoidable 30-day hospital readmissions in medical patients: derivation and validation of a prediction model. *JAMA Internal Medicine*, 173(8):632–638, 2013. 2

Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural

network classification models: a methodology review. *Journal of Biomedical Informatics*, 35(5):352–359, 2002. 10, 13

Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012. 23

David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. Additive Gaussian processes. In *Advances in Neural inf. Process. Syst.*, pages 226–234, 2011. 81, 105

Thorsten H Ecke, Peter Bartel, Steffen Hallmann, Stefan Koch, Jürgen Ruttloff, Henning Cammann, Michael Lein, Mark Schrader, Kurt Miller, and Carsten Stephan. Outcome prediction for prostate cancer detection rate with artificial neural network (ann) in daily routine. In *Urologic Oncology: Seminars and Original Investigations*, volume 30, pages 139–144. Elsevier, 2012. 13

James R Egner. Ajcc cancer staging manual. *JAMA*, 304(15):1726–1727, 2010. 12

Charles Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001. 29

Orhan Er, Nejat Yumusak, and Feyzullah Temurtas. Chest diseases diagnosis using artificial neural networks. *Expert Systems with Applications*, 37(12):7648–7655, 2010. 13

Fern FitzHenry, Harvey J Murff, Michael E Matheny, Nancy Gentry, Elliot M Fielstein, Steven H Brown, Ruth M Reeves, Dominik Aronsky, Peter L Elkin, Vincent P

Messina, et al. Exploring the frontier of electronic health record surveillance: the case of post-operative complications. *Medical Care*, 51(6):509, 2013. 1

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *Proceedings of International Conference on Machine Learning*, volume 96, pages 148–156, 1996. 17

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 283–291. ACM, 2008. 24

Salvador Garcia and Francisco Herrera. An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *Journal of Machine Learning Research*, 9(Dec):2677–2694, 2008. 88, 129

Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010. 25, 27

Mathew J Garnett, Elena J Edelman, Sonja J Heidorn, Chris D Greenman, Anahita Dastur, King Wai Lau, Patricia Greninger, I Richard Thompson, Xi Luo, Jorge Soares, et al. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575, 2012. 3

R Gildersleeve and P Cooper. Development of an automated, real time surveillance tool for predicting readmissions at a community hospital. *Applied Clinical Informatics*, 4(2):153, 2013. 2

Geoffrey S Ginsburg and Jeanette J McCarthy. Personalized medicine: revolutionizing drug discovery and patient care. *Trends in Biotechnology*, 19(12):491–496, 2001. 2

Carlos R González and Yaser S Abu-Mostafa. Mismatched training and test distributions can outperform matched ones. *Neural Computation*, 27(2):365–387, 2015. 127

Assaf Gottlieb, Gideon Y Stein, Eytan Ruppin, Russ B Altman, and Roded Sharan. A method for inferring medical diagnoses from patient similarities. *BMC Medicine*, 11(1):194, 2013. 1

Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017. 30

Margaret A Hamburg and Francis S Collins. The path to personalized medicine. *New England Journal of Medicine*, 2010(363):301–304, 2010. 2

Taizo Hanai, Yasushi Yatabe, Yusuke Nakayama, Takashi Takahashi, Hiroyuki Honda, Tetsuya Mitsudomi, and Takeshi Kobayashi. Prognostic models in patients with non-small-cell lung cancer using artificial neural networks in comparison with logistic regression. *Cancer Science*, 94(5):473–477, 2003. 11

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Linear methods for regression. *The Elements of Statistical Learning*, pages 1–57, 2009. 9

Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009. 30

Nicholas J Horton and Ken P Kleinman. Much ado about nothing. *The American Statistician*, 2012. 26

Cheng-Lung Huang, Hung-Chang Liao, and Mu-Chen Chen. Prediction model building and feature selection with support vector machines in breast cancer diagnosis. *Expert Systems with Applications*, 34(1):578–587, 2008. 15

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006a. 17

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 601–608, 2006b. 22

Brian Hutchinson, Li Deng, and Dong Yu. Tensor deep stacking networks. *IEEE Trans. Pattern Anal. Mach. Intel.*, 35(8):1944–1957, 2013. 31

Joseph G Ibrahim, Ming-Hui Chen, and Stuart R Lipsitz. Monte carlo em for missing covariates in parametric regression models. *Biometrics*, 55(2):591–596, 1999. 27

Vasudevan Jagannathan, Charles J Mullett, James G Arbogast, Kevin A Halbritter, Deepthi Yellapragada, Sushmitha Regulapati, and Pavani Bandaru. Assessment of commercial NLP engines for medication information extraction from dictated clinical notes. *International Journal of Medical Informatics*, 78(4):284–291, 2009. 3

Miles F Jefferson, Neil Pendleton, Sam B Lucas, and Michael A Horan. Comparison of a genetic algorithm neural network with logistic regression for predicting outcome after surgery for patients with nonsmall cell lung carcinoma. *Cancer*, 79(7):1338–1342, 1997. 11

Dokyoon Kim, Hyunjung Shin, Young Soo Song, and Ju Han Kim. Synergistic effect

of different levels of genomic data for cancer clinical outcome prediction. *Journal of Biomedical Informatics*, 45(6):1191–1198, 2012. 3

John P Klein and Mei-Jie Zhang. Survival analysis, software. *Encyclopedia of Biostatistics*, 2005. 11

William A Knaus, Douglas P Wagner, and Joanne Lynn. Short-term mortality predictions for critically ill hospitalized adults: science and ethics. *Science*, 254 (5030):389–394, 1991. 34

Ries Kranse, Monique Roobol, and Fritz H Schröder. A graphical device to represent the outcomes of a logistic regression analysis. *The Prostate*, 68(15):1674–1680, 2008. 151

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inform. Process. Syst.*, pages 1097–1105, 2012. 30

Gert RG Lanckriet, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004. 111

Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, pages 63–66, 2001. 29

KF Leung, M Tay, SSW Cheng, and F Lin. Hong Kong Chinese version World Health Organization quality of life measure-abbreviated version. *WHOQOL-BREF (HK)*, 1997. 90

KF Leung, WW Wong, MSM Tay, MML Chu, and SSW Ng. Development and validation of the interview version of the hong kong chinese whoqol-bref. *Quality of Life Research*, 14(5):1413–1419, 2005. 90

David Lewis and William Gale. Training text classifiers by uncertainty sampling. 1994. 116

Jiaqian Li, Kuo-Kun Tseng, Haiting Dong, Yifan Li, Ming Zhao, and Mingyue Ding. Human sperm health diagnosis with principal component analysis and k-nearest neighbor algorithm. In *2014 International Conference on Medical Biometrics*, pages 108–113. IEEE, 2014. 18

Yan Li, Peng Wen, et al. Classification of eeg signals using sampling techniques and least square support vector machines. In *International Conference on Rough Sets and Knowledge Technology*, pages 375–382. Springer, 2009. 16

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014. 26

Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *19th International Conference on Machine Learning*, volume 2, pages 387–394, July 2002. 22

Wei Liu, Huaxiang Zhang, and Jianbo Li. Inductive transfer through neural network error and dataset regrouping. In *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, volume 1, pages 777–781, Shanghai, China, 2009. 24

Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang.

Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems*, 80:14–23, 2015. 24

Giovanni Lughezzani, Maxine Sun, Shahrokh F Shariat, Lars Budäus, Rodolphe Thuret, Claudio Jeldres, Daniel Liberman, Francesco Montorsi, Paul Perrotte, and Pierre I Karakiewicz. A population-based competing-risks analysis of the survival of patients treated with radical cystectomy for bladder cancer. *Cancer*, 117(1):103–109, 2011. 50

Roger Luis, L Enrique Sucar, and Eduardo F Morales. Inductive transfer for learning bayesian networks. *Machine Learning*, 79(1-2):227–255, 2010. 24

Subhransu Maji, Alexander C Berg, and Jitendra Malik. Efficient classification for additive kernel SVMs. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(1):66–77, 2013. 105

Anil N Makam, Oanh K Nguyen, Billy Moore, Ying Ma, and Ruben Amarasingham. Identifying patients with diabetes and the earliest date of diagnosis in real time: an electronic health record case-finding algorithm. *BMC Medical Informatics and Decision Making*, 13(1):81, 2013. 1

Sebastián Maldonado and Julio López. Imbalanced data classification using second-order cone programming support vector machines. *Pattern Recognition*, 47(5): 2070–2079, 2014. 30

James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011. 1

Alberto M Marchevsky, Sachin Patel, Karen J Wiley, Mark A Stephenson, Margaret Gondo, Richard W Brown, Eunhee S Yi, William F Benedict, Rose C Anton, and Philip T Cagle. Artificial neural networks and logistic regression as tools for prediction of survival in patients with stages i and ii non-small cell lung cancer. *Modern Pathology*, 11(7):618–625, 1998. 11

Alberto M Marchevsky, Swati Shah, and Sachin Patel. Reasoning with uncertainty in pathology: artificial neural networks and logistic regression as tools for prediction of lymph node status in breast cancer patients. *Modern Pathology*, 12(5):505–513, 1999. 11

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. 13

Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. Mapping and revising markov logic networks for transfer learning. In *22nd Conference on Artificial Intelligence*, volume 7, pages 608–614, July 2007. 23

F Millan-Rodriguez, G Chechile-Toniolo, J Salvador-Bayarri, J Palou, and J Vicente-Rodriguez. Multivariate analysis of the prognostic factors of primary superficial bladder cancer. *The Journal of Urology*, 163(1):73–78, 2000. 12

Rupert G Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011. 11

Nomograms. Nomogram predicting the probability of mortality due to bladder cancer versus other causes. URL `http://labs.fccc.edu/nomograms/nomogram.php?id=48audience=1`. 50

Lucila Ohno-Machado. Modeling medical prognosis: survival analysis techniques. *Journal of Biomedical Informatics*, 34(6):428–439, 2001. 11, 34

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 20, 21, 22

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2): 199–210, 2011. 23

G Parthiban, A Rajesh, and SK Srivatsa. Diagnosis of heart disease for diabetic patients using naive bayes method. *International Journal of Computer Applications*, 24(3): 7–11, 2011. 17

Kristiaan Pelckmans, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5):684–692, 2005. 27

Piyaphol Phoungphol, Yanqing Zhang, and Yichuan Zhao. Robust multiclass classification for learning from imbalanced biomedical data. *Tsinghua Science and Technology*, 17(6):619–628, 2012. 30

Kemal Polat and Salih Güneş. Breast cancer diagnosis using least square support vector machine. *Digital Signal Processing*, 17(4):694–701, 2007. 16

Kemal Polat, Bayram Akdemir, and Salih Güneş. Computer aided diagnosis of ecg data on the least square support vector machine. *Digital Signal Processing*, 18(1): 25–32, 2008. 16

Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, pages 705–724, 1981. 10

Asha Rajkumar and G Sophia Reena. Diagnosis of heart disease using datamining algorithm. *Global Journal of Computer Science and Technology*, 10(10):38–43, 2010. 17

Gunnar Rätsch, Takashi Onoda, and Klaus Robert Müller. An improvement of adaboost to avoid overfitting. In *Proc. of the Int. Conf. on Neural Information Processing*. Citeseer, 1998. 17

Irina Rish. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM, 2001. 17

Enrique Romero and Rene Alquezar. A new incremental method for function approximation using feed-forward neural networks. In *Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 2, pages 1968–1973. IEEE, 2002. 17

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958. 13

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898, 2005. 25

Emanuel Rubin and Howard M Reisner. *Essentials of Rubin's pathology*. Lippincott Williams & Wilkins, 2009. 12

Stuart Russell and Peter Norvig. *Artificial Intelligence: A modern approach*. Upper Saddle River, New Jersey: Prentice Hall, 2003. 12

Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *Proc. 12th Int. Conf. Artificial Intell. and Statistics*, volume 1, page 3, Clearwater, Florida, USA, 2009. 30

Catia Matos Salgado, Joaquim Laurens Viegas, Carlos Santos Azevedo, Marta Costa Ferreira, Susana M Vieira, and Joao Miguel da Costa Sousa. Takagi-Sugeno fuzzy modeling using mixed fuzzy clustering. *IEEE Trans. Fuzzy Syst.*, 2016. 105

Yardena Samuels, Zhenghe Wang, Alberto Bardelli, Natalie Silliman, Janine Ptak, Steve Szabo, Hai Yan, Adi Gazdar, Steven M Powell, Gregory J Riggins, et al. High frequency of mutations of the PIK3CA gene in human cancers. *Science*, 304(5670): 554–554, 2004. 3

Marta Sanchez-Carbayo, Nicholas D Socci, Juanjo Lozano, Fabien Saint, and Carlos Cordon-Cardo. Defining molecular profiles of poor outcome in patients with invasive bladder cancer using oligonucleotide microarrays. *Journal of Clinical Oncology*, 24(5):778–789, 2006. 15

Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2004. 24

Henry Selvaraj, S Thamarai Selvi, D Selvathi, and L Gewali. Brain mri slices classification using least squares support vector machine. *International Journal of Intelligent Computing in Medical Sciences & Image Processing*, 1(1):21–33, 2007. 16

Thibaud Senechal, Vincent Rapp, Hanan Salam, Renaud Seguier, Kevin Bailly, and Lionel Prevost. Combining aam coefficients with lgbp histograms in the multi-kernel svm framework to detect facial action units. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 860–865. IEEE, 2011. 111

Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, (19):221–248, 2017. 2

Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7(Jul):1283–1314, 2006. 27

Mai Shouman, Tim Turner, and Rob Stocker. Applying k-nearest neighbour in diagnosing heart disease patients. *International Journal of Information and Education Technology*, 2(3):220, 2012. 18

Daniel L Silver and Robert E Mercer. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 90–101. Springer, 2002. 25

Daniel L Silver and Robert E Mercer. Sequential inductive transfer for coronary artery disease diagnosis. In *Proceedings of International Joint Conference on Neural Networks*, pages 2635–2641. IEEE, 2007. 25

RP Singson, Randa Alsabeh, Stephen A Geller, and Alberto Marchevsky. Estimation of tumor stage and lymph node status in patients with colorectal adenocarcinoma using probabilistic neural networks and logistic regression. *Modern Pathology*, 12 (5):479–484, 1999. 11

Alexander J Smola, SVN Vishwanathan, and Thomas Hofmann. Kernel methods for missing variables. In *Proceedings of 10th International Conference on Artifical Intelligence and Statistics (AISTATS)*, page 325, Barbados, 2005. 27

Kamna Solanki, Parul Berwal, and Sandeep Dalal. Analysis of application of data mining techniques in healthcare. *International Journal of Computer Applications*, 148(2), 2016. 13

SIV Sousa, FG Martins, MCM Alvim-Ferraz, and Maria C Pereira. Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. *Environmental Modelling & Software*, 22(1):97–103, 2007. 10

J.A.K Suykens, T Van Gestel, J De Brabanter, B De Moor, and J Vandewalle. *Least Squares Support Vector Machine Classifiers*. World Scientific, Singapore, 2002. 15, 38

Richard J Sylvester, Adrian PM van der Meijden, Willem Oosterlinck, J Alfred Witjes, Christian Bouffioux, Louis Denis, Donald WW Newling, and Karlheinz Kurth. Predicting recurrence and progression in individual patients with stage ta t1 bladder cancer using eortc risk tables: a combined analysis of 2596 patients from seven EORTC trials. *European Urology*, 49(3):466–477, 2006. 1

Bayu Adhi Tama. Detection of type 2 diabetes mellitus disease with data mining approach using support vector machine. In *Proceeding of The 2010 International Conference on Informatics, Cybernetics, and Computer Applications (ICICCA2010)*. Gopalan College of Engineering and Management, Bangalore, India, 2010. 18

Kai Ming Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002. 29

Ivan Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (6):448–452, 1976. 29

IW-H Tsang, JT-Y Kwok, and Jacek M Zurada. Generalized core vector machines. *IEEE Trans. on Neural Netw.*, 17(5):1126–1140, 2006. 112, 141

Marc J Van De Vijver, Yudong D He, Laura J Van't Veer, Hongyue Dai, Augustinus AM Hart, Dorien W Voskuil, George J Schreiber, Johannes L Peterse, Chris Roberts, Matthew J Marton, et al. A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, 347(25):1999–2009, 2002. 1

Tony Van Gestel, Johan AK Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, and Joos Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1):5–32, 2004. 16

Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Trans. Pattern Anal. Mach. Intell*, 34(3):480–492, 2012. 105

Paul M Vespa, Val Nenov, and Marc R Nuwer. Continuous eeg monitoring in the intensive care unit: early findings and clinical efficacy. *Journal of Clinical Neurophysiology*, 16(1):1–13, 1999. 2

Veena Vijayan and Aswathy Ravikumar. Study of data mining algorithms for prediction and diagnosis of diabetes mellitus. *International Journal of Computer Applications*, 95(17), 2014. 18

Oriol Vinyals, Yangqing Jia, Li Deng, and Trevor Darrell. Learning with recursive perceptual representations. In *Advances in Neural Inf. Process. Syst.*, pages 2825–2833, 2012. 31, 108

Guanjin Wang, Kin-Man Lam, Zhaohong Deng, and Kup-Sze Choi. Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques. *Computers in Biology and Medicine*, 63:124–132, 2015. 53

Guanjin Wang, Zhaohong Deng, and Kup-Sze Choi. Tackling missing data in community health studies using additive LS-SVM classifier. *IEEE Journal of Biomedical and Health Informatics*, pages 579–587, 2016a. 58

Guanjin Wang, Guangquan Zhang, Kup-Sze Choi, Kin-Man Lam, and Jie Lu. An output-based knowledge transfer approach and its application in bladder cancer prediction. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 356–363. IEEE, 2017a. 33

Guanjin Wang, Guangquan Zhang, Kup-Sze Choi, and Jie Lu. Deep additive least squares support vector machines for classification with model transfer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017b. (Accepted and on-line available). 104

Mingliang Wang, Han-Xiong Li, Xin Chen, and Yun Chen. Deep learning-based model reduction for distributed parameter systems. *IEEE Trans. Syst., Man, Cybern., Syst.*, 46(12):1664–1674, 2016b. 30

John B Welsh, Lisa M Sapinoso, Andrew I Su, Suzanne G Kern, Jessica Wang-Rodriguez, Christopher A Moskaluk, Henry F Frierson, and Garret M Hampton.

Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Research*, 61(16):5974–5978, 2001. 3

D Randall Wilson and Tony R Martinez. Heterogeneous radial basis function networks. In *IEEE International Conference on Neural Networks, 1996*, volume 2, pages 1263–1267. IEEE, 1996. 17

Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972. 29

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. 31

Donghui Wu, Zhelong Wang, Ye Chen, and Hongyu Zhao. Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. *Neurocomputing*, 190:35–49, 2016. 30

Chi-Yuan Yeh, Chi-Wei Huang, and Shie-Jue Lee. A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Syst. with Appl.*, 38 (3):2177–2186, 2011. 111

Jinbo Bi Tong Zhang. Support vector classification with input data uncertainty. *Advances in Neural Information Processing Systems*, 17:161–169, 2005. 27

Peng Zhang and Jing Peng. Svm vs regularized least squares classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 176–179. IEEE, 2004. 16

Chengyi Zheng, Nazia Rashid, Yi-Lin Wu, River Koblick, Antony T Lin, Gerald D Levy, and T Craig Cheetham. Using natural language processing and machine

learning to identify gout flares from electronic clinical notes. *Arthritis care & Research*, 66(11):1740–1748, 2014. 3

Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 814–822. ACM, 2011. 25

Jiayu Zhou, Jun Liu, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013. 25

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006. 29

Hua Zuo, Guangquan Zhang, Vahid Behbood, and Jie Lu. Feature spaces-based transfer learning. In *16th World Congress of the International Fuzzy Systems Association, and 9th Conference of the European Society for Fuzzy Logic and Technology*, 2015a. 23

Hua Zuo, Guangquan Zhang, Vahid Behbood, Jie Lu, and Xianli Meng. Transfer learning in hierarchical feature spaces. In *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 183–188, 2015b. 24