University of Miami
Scholarly Repository

Open Access Dissertations | Electronic Theses and Dissertations

2012-05-04

Knowledge-Assisted Sequential Pattern Analysis: An Application in Labor Contraction Prediction

Zifang Huang
University of Miami, zifangh@gmail.com

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation
Huang, Zifang, "Knowledge-Assisted Sequential Pattern Analysis: An Application in Labor Contraction Prediction" (2012). *Open Access Dissertations*. 757.
https://scholarlyrepository.miami.edu/oa_dissertations/757

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.

UNIVERSITY OF MIAMI

KNOWLEDGE-ASSISTED SEQUENTIAL PATTERN ANALYSIS: AN APPLICATION IN LABOR CONTRACTION PREDICTION

By

Zifang Huang

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2012

©2012
Zifang Huang
All Rights Reserved

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

KNOWLEDGE-ASSISTED SEQUENTIAL PATTERN ANALYSIS: AN APPLICATION IN LABOR CONTRACTION PREDICTION

Zifang Huang

Approved:

Mei-Ling Shyu, Ph.D. Associate Professor of Electrical and Computer Engineering	Terri A. Scandura, Ph.D. Dean of the Graduate School
James M. Tien, Ph.D. Professor of Electrical and Computer Engineering	David J. Birnbach, M.D., M.P.H. Professor of Anesthesiology and Public Health
Mohamed Abdel-Mottaleb, Ph.D. Professor of Electrical and Computer Engineering	Xiaodong Cai, Ph.D. Associate Professor of Electrical and Computer Engineering

HUANG, ZIFANG (Ph.D., Electrical and Computer Engineering)

Knowledge-Assisted Sequential Pattern Analysis: An Application in Labor Contraction Prediction (May 2012)

Abstract of a dissertation at the University of Miami.

Dissertation co-supervised by Professor Mei-Ling Shyu,
Professor James M. Tien, and Professor David J. Birnbach.
No. of pages in text. (145)

Although neuraxial techniques, such as spinal and epidural, are still considered as the gold standard for labor analgesia, there are some parturients who cannot receive neuraxial analgesia because of pre-existing conditions, or who request analgesia other than epidural block. An alternative analgesia is remifentanil, which is a relatively new, very potent and short-acting opioid. It has been shown to be effective in the relief of labor pain, but reports to date have failed to find the optimal dosing regimen. A challenge to a systemic opioid is that it must match the unique time course of labor pain. A continuous infusion is not ideal, as the parturient experiences no pain between contractions. Moreover, a continuous infusion during times in which the patient does not experience pain, may increase the risks of respiratory depression, sedation and nausea. The continuous infusion also increases the amount of the drug to which the fetus is exposed.

Designing an optimal dosing regimen necessitates the prediction of the pace of contractions, so that the drug can be given shortly before the pain of the contraction begins. The prediction and thus drug administration should be made early enough to allow for the administration of intravenous analgesia that will have maximal efficacy during contractions, little effect between contractions, and minimal impact on the fetus. Towards such a need, we propose a knowledge-assisted sequential pattern

analysis framework to predict the changes in intrauterine pressure, which indicate the occurrence of labor contractions. The proposed framework predicts in real time and provides a prediction multiple seconds before a contraction occurs, so as to assist in designing optimal administration strategies of remifentanil in labor.

The proposed framework first selects a group of patients, from the stored record, who share similar demographic and obstetrical information with the current patient of interest. Second, it develops a sequential association rule mining approach to learn the patterns of the contractions from the historical patient tracings of the selected patients. Third, a sequential association rule-based collaborative filtering strategy is designed to dynamically select a training dataset from the historical patient tracings, as well as from the most recent training time series of the patient of interest. The training set is used for training a set of prediction models. A k-nearest neighbors (k-NN) based least squares support vector machine (LS-SVM) approach with heuristic parameter tuning is proposed to conduct the long-term time series prediction. A post-prediction process is also incorporated to further enhance the prediction results. Because to the best of our knowledge, there has been no previous study to predict future contractions, this work can be considered as a pioneer in the field.

We evaluate the performance of the proposed framework using actual data from anonymous patients with varied contraction patterns. The data include patient demographic and obstetrical information, and measured intrauterine pressure time series. Overall, the proposed framework outperforms several well-known prediction methods, and it accomplishes that in real time. Meanwhile, experiments that compare each component with some other famous algorithms are conducted. The promising experimental results show that all proposed components improve the prediction preci-

sion, and the proposed framework achieves the effectiveness, robustness and efficiency that are needed for designing the optimal dosing regimen of remifentanil.

To my beloved parents

Acknowledgment

I would like to acknowledge and extend my heartfelt gratitude to my dissertation co-advisors, Dr. Mei-Ling Shyu, Dr. James M. Tien, and Dr. David J. Birnbach, for their guidance and support throughout this work. I am especially thankful to Dr. Mei-Ling Shyu, a devoted researcher and mentor, for providing a stimulating research environment, for giving me constructive advice during the academic program, and for always encouraging me to reach higher grounds in research. I am very grateful to Dr. James M. Tien and Dr. David J. Birnbach, who invested a significant amount of time to provide me with valuable guidance and detailed feedback, which have helped me achieve additional insights into my dissertation topic and have given me new perspectives on research.

I would also like to thank my dissertation committee members, Dr. Mohamed Abdel-Mottaleb and Dr. Xiaodong Cai, for valuable input and support. Their feedback was important for improving the quality of this work. I am very grateful to Dr. Mei-Ling Shyu, the Department of Electrical and Computer Engineering, and Dr. James M. Tien for the financial support during the Ph.D. program. Special thanks go to the staff in the Department of Electrical and Computer Engineering, Michelle R. Perez, Rosamund Coutts, and Angie Del-Llano, for their kindness and willingness to help with both administrative and personal issues.

My gratitude goes to the lovely friends I met at the University of Miami for their priceless friendship, which makes this journey easier and more meaningful. In particular, I would like to thank all the members in the Data Mining, Database and Multimedia Research Group at the University of Miami for their help, consideration, and encouragement. A very special recognition needs to be given to Dr. Michael M. Vigoda for his extensive help and support in communicating with the technicians in GE Company and Jackson Memorial Hospital, which made the experiments on real patient data possible. I would like to extend my sincere gratitude to Mike Jordan, Dr. Jayanthie S. Ranasinghe, Juan A. Mesa, and Jean Heichman for their help and assistance in extracting and retrieving labor monitoring data, without which the experimental validation of the proposed framework in this dissertation would not be possible.

Finally, I dedicate this dissertation to my beloved parents for their unconditional love and support in every way possible throughout the Ph.D. program, this dissertation and beyond.

ZIFANG HUANG

University of Miami

May 2012

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In the United States today, pregnant women predominantly choose neuraxial blockade (epidural and combined spinal epidural) for the management of labor pain. Although neuraxial techniques are considered the gold standard for labor analgesia, some women cannot receive the neuraxial analgesia because of pre-existing conditions or preference for analgesias other than an epidural block. In some parts of the world, anesthesiologists are not trained to administer neuraxial blocks; in addition, the lack of analgesic efficacy and side effects of traditional opioids (morphine, meperidine, fentanyl) to treat labor pain has necessitated the search for an alternative approach.

One alternative is the accurately timed delivery of remifentanil. Remifentanil is a relatively new, very potent, short-acting μ-opioid agonist, which is chemically related to fentanyl (Evron, Glezerman, Sadan, Boaz, and Ezri 2005). Its major advantages over other opioids include rapid onset of action and rapid clearance rate by red blood cells and tissue esterase to an inactive metabolite (Evron, Glezerman, Sadan, Boaz, and Ezri 2005). Consequently, prolonged administration does not cause accumulation of the drug and has minimal effects on the neonate. Remifentanil can be used either

as a continuous infusion or as boluses, and has been shown to be effective in the relief of labor pain (Volmanen, Akural, Raudaskoski, and Alahuhta 2002).

A challenge to the administration of any systemic opioid is that it should optimally match the individual time course of the labor pain. A continuous infusion is suboptimal, as the parturient experiences no pain between contractions, and it may increase the risk of respiratory depression, sedation, and other side effects. The onset of the opioid's effect is approximately 30 seconds (Hill 2008), so the prediction should be made approximately 30 seconds (adjustable according to the requirement) ahead of the next contraction to accurately match the effect of analgesia. However, anticipating contractions over a protracted time interval is challenging because of the inherent uncertainty of the labor experience.

Data mining techniques could be utilized to analyze the uterine contraction-related data, and thus to improve prediction accuracy and efficiency. Data mining, also known as knowledge discovery in databases, is the process of discovering new patterns from data, and is a method of extracting knowledge that cannot be observed from the surface (Witten and Frank 2005). Data mining provides interesting patterns in terms of rules, groups, and characterizations by using association rule mining (Agrawal, Imieliński, and Swami 1993), classification (Jain, Duin, and Mao 2000), clustering (Xu and Wunsch 2005), and sequential pattern analysis techniques (Shyu, Huang, and Luo 2009). Before developing any data mining techniques, it is necessary to understand the fundamental reason for uterine contractions from the physiological perspective.

Uterine contractility is a direct consequence of the underlying electrical activities in the myometrial cells. Spontaneous electrical activities in the muscles in the uterus are composed of intermittent bursts of spike action-potentials (Marshall 1962). Sin-

gle spikes can initiate contractions, but multiple, higher-frequency, and coordinated spikes are needed for forceful and maintained contractions (Marshall 1962).

Isolated myometrial tissue studies, using microelectrodes or extracellular electrodes, have demonstrated the connection between electrical events and contractions (Marshall 1959; Wolfs and Rottinghuis 1970). The frequency, amplitude, and duration of contractions are determined mainly by the frequency of the occurrence of uterine electrical bursts, the total number of cells that are simultaneously active during the bursts, and the duration of the uterine electrical bursts, respectively (Marshall 1962). Each burst stops before the uterus has completely relaxed (Marshall 1962). The agents that directly stimulate or inhibit uterine contractions do so by altering the electrical properties and the excitability or conductivity of myometrial cells.

Studies have been performed in an attempt to find specific types of cells that may act as the pacemakers for the human uterus (Duquette, Shmygol, Vaillant, Mobasheri, Pope, Burdyga, and Wray 2005). However, no specific pacemaker has been found thus far. It has therefore been suggested that the spontaneous electrical behavior exhibited by the myometrium is an inherent property of the smooth-muscle cells within the myometrium.

To conduct predictions, we need to monitor uterine contractions and extract the patterns. Since electrical activities of the uterus are correlated to uterine contractions (Marshall 1962), it is intuitive to record the electrical activities directly from the uterus. Numerous studies (Garfield and Maner 2007; Terrien, Marque, and Germain 2008; Rabotti, Mischi, Oei, and Bergmans 2010) have provided convincing evidence that uterine electromyography (EMG) activity can be appraised from non-invasive trans-abdominal surface measurements, as shown in Fig. 1.1, and can be a power-

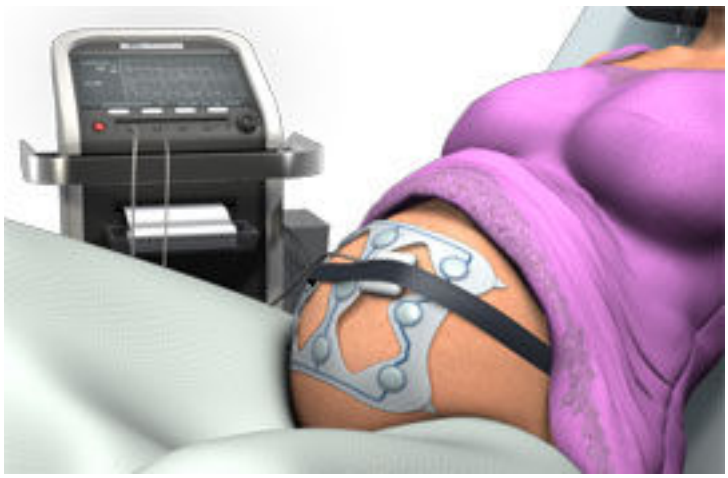

Figure 1.1: Typical Uterine EMG Setup (OB-Tools 2011)

ful tool in characterizing parturition. Typical uterine EMG setup includes abdominal surface electrodes, electrical filters/amplifiers, and acquisition and analysis hardware/software. The EMG bursts correspond to uterine contractions. Both EMG and electrocardiography (ECG) interpret electrical activities, where ECG is a transthoracic interpretation of electrical activities of the heart over time captured and externally recorded by skin electrodes. The presence of ECG signals often corrupt the EMG signals recorded from the trunk area (Hu, Mak, and Luk 2009), unreliable for conducting prediction. The uterine magnetomyogram (MMG) is a noninvasive technique that measures the magnetic fields associated with the action potentials, while the detection of uterine contractions from the MMG signals is still under study (La Rosa, Nehorai, Eswaran, Lowery, and Preissl 2008).

The tocodynamometer is a pressure-sensitive contraction transducer, as shown in Fig. 1.2, which externally measures the tension of the maternal abdominal wall. However, this measurement is easily disturbed by patient movement or other interfer-

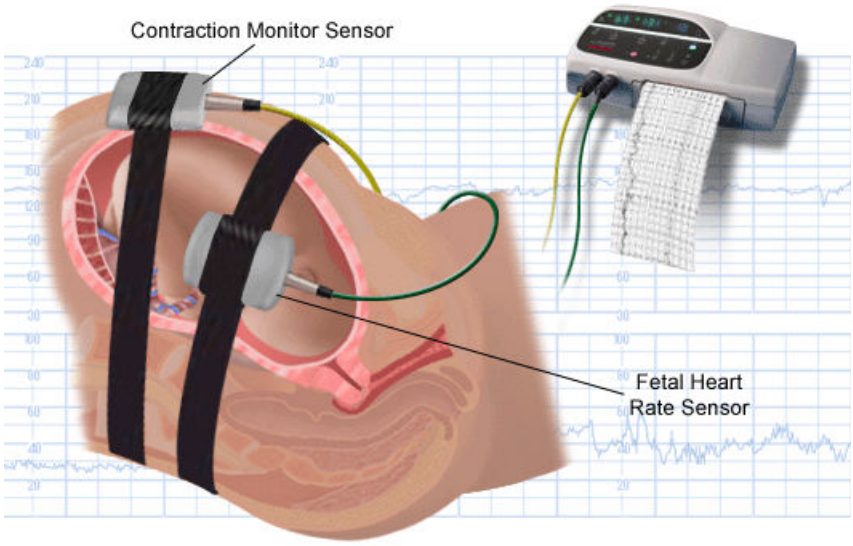

Figure 1.2: Tocodynamometer Setup (Palomar Pomerado Health 2011)

ence. An alternative to the external measurement is the intrauterine pressure catheter (IUPC) which measures the exact force of the contractions during labor. It becomes much easier to detect the contractions, since the pressure increases when the uterus starts to contract and decreases when the uterus starts to relax. In our study, we analyze the signal collected by IUPC because it is far more accurate and specific than the tocodynamometer.

1.1 Motivation and Challenges

The time to peak analgesic effect of a remifentanil bolus usually varies between 60-90 seconds (Saunders and Glass 2002; Glass, Hardman, Kamiyama, Quill, Marton, Donn, Grosse, and Hermann 1993) and a contraction generally lasts approximately 60 seconds. Thus a remifentanil bolus given at the beginning of a uterine contraction

will not give the peak analgesic effect for the contraction for which it is administered. Therefore, if a remifentanil bolus is to be optimally administered, it is necessary to anticipate a forthcoming contraction. This, unfortunately, has been difficult. When the peak analgesic effect of a remifentanil bolus does not coincide with the peak of the contraction, the risk for sedation, respiratory depression, and fetus depression increases. After a review of the physiology of labor contractions, we appreciate that a pacemaker for contractions has thus far not been found (Duquette, Shmygol, Vaillant, Mobasheri, Pope, Burdyga, and Wray 2005), and there is no such factor from the physiological point of view that is able to predict the uterine activities ahead of time. Therefore, an appropriate way for us to approach this problem is to analyze the intrauterine pressure time series, and undertake the prediction of contractions based on the learned patterns. We aim to develop data mining techniques, and discover the pattern/knowledge that is hidden within the data. Meanwhile, in order to predict the next contraction multiple seconds before it occurs, a long-term prediction is required. The ever changing labor contraction pattern and diversity across different patients have posed many challenges to building a predication model with regard to denoising, adaptive long-term time series prediction, knowledge discovery from the patients' tracings, and personalization of the prediction.

- Denoising challenge, *i.e.*, the challenge of removing and decreasing the noise in the collected signal.

Even though we are analyzing the intrauterine pressure time series collected by the intrauterine pressure catheter (which measures the uterine activity more accurately than the tocodynamometer), the signals are also contaminated by noise, and suffer

from loss of data because of patient movements, system imperfections, and other unforeseen interference. Data quality directly influences the model's prediction ability. Due to various sources of the noise, the noise presents different patterns, such as low frequency interference, high frequency interference, bursts, and missing values. Therefore, a low pass filter or a high pass filter or a band pass filter cannot filter out all types of noise. In addition, the filters usually cause delays and result in misshaping of the original signal.

- Adaptive long-term time series prediction challenge, *i.e.*, the challenge of time series prediction, which is adaptive to the changing patterns of the time series, over a large prediction horizon.

Remifentanil is a short-acting opioid anesthetic agent, but it does not take effect immediately. In order to make sure that the analgesic effect of remifentanil coincides with uterine contractions, the prediction model must anticipate the next contraction early enough to allow the administration of the remifentanil. Therefore, a long-term prediction of labor contractions appears to be necessary. Long-term time series prediction (Bontempi 2008; Huang and Shyu 2010; Zhou, Xu, and Wu 2010) is a very challenging task due to the growing uncertainties arising from various sources, for instance, the accumulation of errors and the lack of information about the future (Weigend and Gershenfeld 1994). In addition, the pattern of uterine contractions changes as labor progresses. Also, different patients present diverse contraction patterns in terms of intensity, duration, and period. Thus this renders a huge challenge in obtaining a single model that is suitable for all patients at all stages of labor.

- Knowledge discovery from the patient tracing records challenge, *i.e.*, the challenge of extracting knowledge from a set of historical labor tracings, and using it to assist predicting contractions of the current patient of interest.

We have access to a large set of historical patient labor tracings. How to utilize the historical patient tracings to help predict a new patient's upcoming contraction becomes an interesting and difficult challenge. Even though patient intrauterine pressure tracings present different patterns, we can observe repeating sequential patterns that they have in common, especially for those patients who share similar demographic and obstetrical information. For example, a patient has a contraction that lasts 60 seconds with the highest intrauterine pressure at 85 mmHg, and the following contraction that lasts 65 seconds with the highest intrauterine pressure at 90 mmHg. The same sequential pattern, *i.e.*, a contraction that lasts 60 seconds with the highest intrauterine pressure at 85 mmHg followed by a contraction that lasts 65 seconds with the highest intrauterine pressure at 90 mmHg, is also observed in other patients. Therefore, if the patient of interest experiences a contraction that lasts 60 seconds with the highest intrauterine pressure at 85 mmHg, the chance of her having the next contraction that lasts 65 seconds with the highest intrauterine pressure at 90 mmHg is high. This observation motivated us to discover interesting sequential patterns among the historical patient tracings, and attempt to utilize the sequential patterns to predict future contractions for a new patient.

- Personalizing the prediction challenge, *i.e.*, the challenge of making the prediction model customized for each patient.

The contraction pattern differs from woman to woman, from pregnancy to pregnancy, and also changes in different stages of a woman's labor, which makes it very challenging to accurately perform a prediction. It would be preferred if the prediction model is personalized, which is adaptive to each individual patient's age, weight, gestational age, labor anesthesia and oxytocin usage, *etc.* In order to achieve this goal, a primary condition is that the patients' demographic and obstetrical information should be available as one of the inputs to the model training process. The demographic and obstetrical features are mostly nominal, and how to integrate the nominal features in predicting numerical results becomes another challenge.

1.2 Contributions and Limitations

To address the aforementioned challenges, we propose a novel knowledge-assisted sequential pattern prediction framework for predicting the intrauterine pressure multiple seconds ahead in real time. The flowchart of the framework is shown in Fig. 1.3. In the proposed framework, we first select a group of historical patient tracings (HT) from the stored records of patients information (HI) based on the current patient's demographic and obstetrical information. The selected patients are those who share similar demographic and obstetrical information with the current patient of interest. We design a sequential association rule-based collaborative training dataset selection method to dynamically select a training dataset from the HT and the current patient's own most recent training time series for training the prediction models. A k-nearest neighbors (k-NN) based least squares support vector machine (LS-SVM) approach with heuristic parameter tuning is proposed to conduct long-term time series predic-

tion. A post-prediction process is proposed to further enhance the prediction results. The proposed framework conducts the prediction in real time. To the best of our knowledge, there has been no previous study to predict contractions.

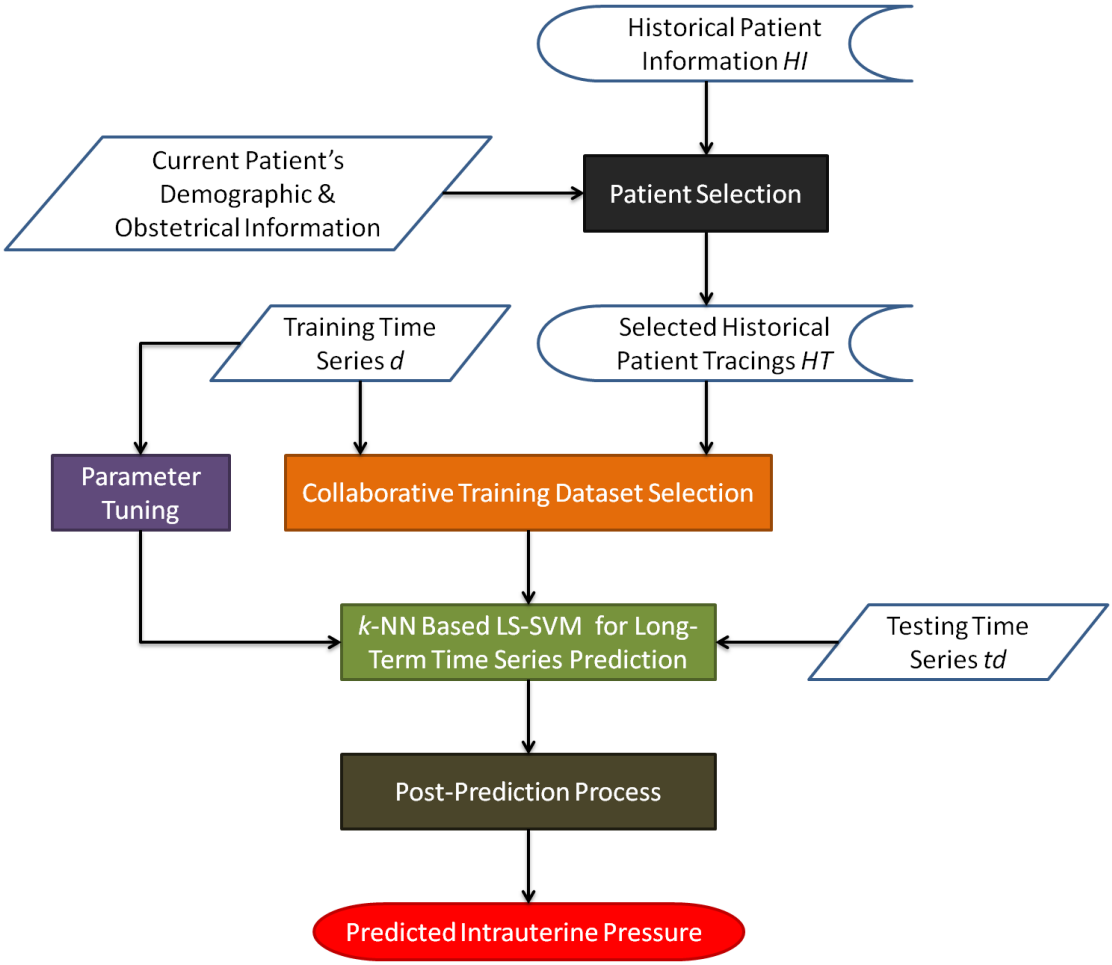

Figure 1.3: Flowchart of the Proposed Framework

Generally, there are four main contributions of this dissertation, which are summarized and listed as follows.

1. Design a new sequential association rule mining approach to analyze the sequential uterine contraction pattern, and group the patients based on some

demographic and obstetrical features that have an impact on the sequential uterine contraction pattern.

A new peak point detection and interpolation-based filtering strategy is proposed to remove the noise in the intrauterine time series. The denoised intrauterine time series are then used for pattern analysis. A contraction can be described by the corresponding peak height and period. The contraction pattern is defined by the combination of the height and the period of a contraction. Instead of calculating the mean or some other statistic measures of a sequence of contractions, we propose a sequential association rule mining approach to analyze the sequential uterine contraction pattern. The height and the period are discretized respectively for the rule mining purpose.

We further analyze whether the demographic and obstetrical features have some impacts on the sequential uterine contraction pattern. The demographic and obstetrical features we considered include maternal age, body mass index, gestational age, number of pregnancies, living children pregnancy history, labor anesthesia, and indication of oxytocin. We divide the patients into groups with different values of one feature, and then analyze the sequential uterine contraction pattern of each group. If the sequential uterine contraction patterns of these groups are different, it means that this feature has a strong impact or determines the sequential uterine contraction pattern. Thus the feature is kept for grouping the patients; otherwise, the feature is discarded. This process is repeated for each of the considered features. Based on the analysis of 611 patients' information, we observe that maternal age, gestational age, labor anesthesia,

and indication of oxytocin impact the sequential uterine contraction pattern more significantly than the rest of the features. Therefore, these four features are used to group the patients for later analysis. This is a new perspective on analyzing the uterine contraction pattern. It is the first attempt in the field to consider the sequential uterine contraction patterns, and to use demographic and obstetrical information to differentiate patients.

2. Design a new sequential association rule-based collaborative filtering strategy to dynamically select the training dataset to train the prediction models.

   A new sequential association rule-based collaborative filtering strategy is designed to dynamically select the training dataset from the patient training database, as well as from the current patient's own training dataset, and introduce a concept of using the 'future' to predict the future. The collaborative training dataset selection component utilizes a proposed sequential association rule mining (ARM) technique to discover frequent sequential contraction patterns from the historical patient tracings. Also, this component employs the collaborative filtering technique to utilize the discovered sequential contraction patterns of the communities of similar patients to assist in predicting the contraction pattern of the current patient. To the best of our knowledge, this is the first attempt that tries to incorporate collaborative filtering and association rule mining techniques in time series prediction.

   Collaborative filtering is the process of filtering information using techniques involving collaboration among multiple data sources (Sarwar, Karypis, Konstan, and Riedl 2001; Linden, Smith, and York 2003). The rationale behind

collaborative filtering is that a user's preferences, interests, or behaviors can be predicted by leveraging the preferences, interests, or behaviors of the communities of similar users. In our proposed framework, collaborative filtering is done by formulating the contraction prediction problem in the manner that if the current patient's contraction pattern matches the contraction patterns of some previous patients, it is very likely that the current patient's next contraction will be similar to, if not the same as, the next contractions of other patients in the patient database. Based on such an assumption, our proposed framework builds a database of contraction patterns/profiles (in the forms of sequential association rules) from the historical patient labor tracings, and searches for the matching sequential association rules of the current patient's contraction pattern to predict the pattern of the current patient's next contraction. The idea is that if the current patient's contraction pattern matches with the condition part of a rule (*i.e.*, the contraction pattern of the past patients), the patient's next contraction pattern is predicted to be the same as the consequent part of the rule (*i.e.*, the next contraction pattern of previous patients). Through the matching process, we can selectively utilize the available data to train the prediction models.

3. Design a novel k-NN based LS-SVM approach with heuristic parameter tuning strategies for long-term time series prediction.

   As a classical kernel-based method for regression, LS-SVM has been widely applied to time series prediction, especially LS-SVMs with nonlinear kernels such as radial basis function (RBF) kernels (Suykens, Gestel, Brabanter, Moor,

and Vandewalle 2002). However, selecting optimal hyper-parameters has always been a challenging task for researchers. We propose a novel heuristic parameter tuning approach to decide the appropriate value ranges and search strategies for both the regularization factor and the Gaussian kernel parameter of LS-SVM with the RBF kernel. The parameter tuning method is based on information extracted from the training dataset, and it is adaptive to the characteristics of a specific time series data.

To perform long-term time series prediction, we adopt the direct prediction strategy, which trains an individual model for each prediction horizon respectively. A k-NN component is introduced to select the instances from the selected training dataset for training the LS-SVM regressors, so as to reduce the computation complexity of the training phase and to improve the prediction precision. We also propose a new distance function for the k-NN approach, which considers both the Euclidean distance and the dissimilarity of the trend between the time series. The proposed k-NN based LS-SVM approach for long-term time series prediction is also the first attempt to utilize the k-NN technique in LS-SVM for long-term time series prediction.

4. Design a unique post-prediction process, which includes boundary constraint, multi-value integration and vertical correction components.

   The post-prediction process is designed to further improve the prediction precision, and makes sure that the results are valid in the application domain. The boundary constraint component sets a dynamic constraint, including the upper bound and lower bound, for the predicted values. The multi-value integration

component combines prediction results from several individual models to generate an output. The vertical correction component is designed specially for intrauterine contraction time series, which detects and smoothes out the irregular sharp pulses and peaks with very low heights in the prediction results. The output of this component is the final output of the proposed framework.

Scope and limitations of the proposed framework are listed below.

1. The current work mainly focuses on the intrauterine pressure time series data. Even though IUPC is a more accurate measure than tocodynamometer, there are some intrauterine pressure time series that do not show the typical regular contraction pattern (*i.e.*, they are not composed of intermittent peaks one after another with rest periods in between). The application of the proposed framework is limited to those patients whose intrauterine pressure time series are regular, *i.e.*, containing a sequence of recognizable peaks.

   In addition, IUPC is not available in some third world countries, where an automatic anesthesia equipment is the most in need due to the lack of experienced anesthetists. Consequently, the next step of this work includes designing a more advanced noise filtering or signal recovering technique to recognize contractions from signals obtained by external instruments, so as to enable further analysis.

2. Some parameters and thresholds are determined based on an iterative search.

   In addition to the two hyper-parameters for LS-SVM, we also have some other parameters and thresholds in the proposed framework, such as the length of the input vector, the length of the training time series, and the minimum support.

Each of these parameters and thresholds is determined according to an iterative process that is carried out in order to find the most suitable value for the training dataset. In each iteration, different values are applied to the training dataset, and the one with the lowest error measure is selected as the threshold value.

Meanwhile, it is difficult to decide the optimal searching range and iteration step size, because if the searching range is large and the step size is small, the computational cost might be too high. In contrast, if the searching range is small and step size is large, the computational cost is low, but it might skip the global optimum. There might also be an overfitting problem that the values are too fitted to the training dataset.

1.3 Organization of the Dissertation

The dissertation is organized as follows. Chapter 2 provides a literature review on the physiology of labor contractions, time series prediction, and knowledge discovery and data mining techniques. Their general applications are included as well.

Chapter 3 describes the structure of the proposed knowledge-assisted sequential pattern analysis framework, and also presents some related techniques, including the long-term time series prediction and LS-SVM for regression. This chapter details the preprocess procedure for the inputs of the framework, k-NN based LS-SVM for long-term time series prediction, and the post-prediction process. It also presents the criteria for evaluating the proposed framework.

Chapter 4 presents the proposed patient selection and collaborative training dataset selection components. First, the contraction feature extraction method is introduced

to map the intrauterine pressure time series to the feature space. Second, a discretization method is developed to nominalize the extracted numerical features for the following sequential association rule mining algorithm. Third, a sequential association rule mining algorithm is proposed to discover the interesting sequential patterns among the contractions. Based on the sequential association rule mining approach, a patient selection approach is proposed to select some patients from the stored records, who share similar demographic and obstetrical information with the current patient of interest. Also, a rule-based collaborative training dataset selection method is proposed to dynamically generate a training dataset from the selected patients' intrauterine pressure database and current patient's own past tracing.

Chapter 5 focuses on the heuristic parameter tuning strategies for LS-SVM. A novel heuristic method is designed to decide the searching range of the Gaussian kernel parameter based on the information extracted from the training time series. A strategy for efficiently locating the regularization factor is also proposed. Experiments were conducted to evaluate the efficacy of the proposed strategies.

Chapter 6 describes the dataset used for the experiments, and provides experimental results for evaluation and comparison.

Chapter 7 concludes the dissertation by highlighting the achievements of the proposed approach and introduces avenues for future work.

CHAPTER 2

Literature Review

In this chapter, the studies on the coordination of uterine contractions are reviewed. The detailed discussion is also provided on time series prediction algorithms, knowledge discovery and data mining techniques, and their applications.

2.1 Coordination of Uterine Contractions

Uterine contraction is a direct consequence of the underlying electrical activities in the myometrial cells. Electrical and contractile activities in the myometrial cells are controlled by myogenic, neurogenic, and hormonal control systems (Challis and Lye 1994; Garfield 1987).

Myogenic activity, the spontaneous activity of the myometrium that occurs in the absence of any neural or hormonal input, includes the intrinsic excitability of the muscle cell, the ability of the muscle to contract spontaneously, and the mechanisms that produce rhythmic contractions. A biopsy specimen of uterine tissue, placed in a physiological solution, will contract involuntarily every 2 to 5 minutes without stimulation (Coad and Dunstall 2001). It is important to notice the ability of the muscle to contract spontaneously without any neural or hormonal input. As a type

of contractions, it is also caused by the movement of ions. However, it is initiated by the cell itself, not by an outside occurrence or stimulus. How can we evaluate the myogenic activity? Can it be measured? The myogenic property typically varies between patients and during parturition. Based on the published literature, it is hard to know when the ion channels in the cell membrane would spontaneously open and close, which is described more as a spontaneous activity.

On top of the muscle's inherent myogenic properties, neurogenic and hormonal control systems are superimposed to initiate, augment, and suppress myometrial activities (Challis and Lye 1994; Garfield 1987). Myogenic control is dominated by hormonal influences, including estrogen, oxytocin, progesterone, relaxin, *etc.*, especially those of estrogen and progesterone, which influence myogenic characteristics through their generally opposing actions (Garfield, Ali, Yallampalli, and Izumi 1995). In summary, progesterone and relaxin are found to inhibit myometrial contractility, while estrogen and oxytocin increase uterine activities. However, measuring absolute hormone levels may be misleading, because the biological effects are also determined by the receptor density, levels of proteins, or postreceptor changes (Coad and Dunstall 2001). In addition, hormone concentrations change as labor progresses. Meanwhile, local changes in hormone concentration, not reflected in peripheral blood, would also affect myometrial activities. In order to utilize hormonal information in predicting the coming contractions, we would need to have access to the peripheral and local hormone concentrations, receptor density, levels of proteins, and postreceptor changes in real time. However, this is not achievable based on the current monitoring techniques. On the other hand, neurogenic control is not very critical, since labors can occur in women with spinal injury, although the length of the labor may be altered in

these women (Coad and Dunstall 2001). Contractions can be temporarily abolished by emotional disturbances, such as moving from the home to the hospital, a change in staff shifts, and some other neurogenic factors. The frequency and strength of the contractions can be increased by stretching of the cervix or pelvic floor by the presenting part of the baby.

The coordination of contractions occurs by coupling of myometrial cells via electrical (*e.g.*, gap junctions) and chemical (*e.g.*, oxytocin) mechanisms (Baxi and Petrie 1987). Polarization and depolarization of the cell membranes move the electrical signals across the myometrium (Akerlund 1997). These electrical signals are generated by the movement of calcium through ion channels into the myometrial cell. Action potentials propagate rapidly throughout the uterus, initiating the movement of calcium into the cells. These intracellular calcium waves propagate more slowly than the action potentials, gradually increasing the number of bundles involved in the contractions (Garfield, Blennerhassett, and Miller 1988).

As action potentials are conducted to the neighboring myometrial cells, the groups of cells contract, leading to what is perceived by the woman as a uterine contraction. The coordination of uterine contractions occurs when all myometrial cells contract nearly simultaneously. Coordination and synchronization of contractions are facilitated by the low-resistance gap junctions between myometrial cells. These junctions allow the propagation of the action potentials between cells and thus throughout the uterus.

Furthermore, the gap junction mechanism explains the changes in contraction patterns during pregnancy and delivery. The number of gap junctions influences the areas of the uterus that contract, which determines the intrauterine pressure levels.

In the literature, it is reported that the gap junction density can be determined by measuring the electrical resistance of tissues (Coad and Dunstall 2001), but this measurement cannot be done in real time. Therefore, an appropriate way for us to study the contraction pattern is to analyze the intrauterine pressure time series, and undertake the prediction of contractions based on the learned pattern. The timing and dosage of the remifentanil can be determined accordingly based on the prediction results to relieve the labor pain.

2.2 Time Series Prediction

Time series prediction is to predict the future values based on the current and previous values of a time series. Time series modeling and prediction are very attractive topics, which play an important role in many fields such as transportation prediction (Crone 2010), power prediction (Kusiak, Zheng, and Song 2009; Maralloo, Koushki, Lucas, and Kalhor 2009; Varadan, Leung, and Bosse 2006), and health care (Homma, Sakai, and Takai 2009; Coyle 2009). Most of the studies on time series prediction focus on one-step ahead prediction, *i.e.*, short-term time series prediction. A more challenging task in the time series prediction domain is long-term time series prediction, in which prediction must be done multiple steps ahead. When the prediction horizon increases, the uncertainty of the future trend also increases, and it is harder to model and capture the inherent relationship of a time series.

The prediction process is commonly performed by observing and modeling past values, and assuming that future values will follow the same trend. For short-term time series prediction, there are plenty of classical time series prediction approaches,

such as exponential smoothing (Jones 1966), linear regression (Lin, Lin, Zhou, and Yao 2007), autoregressive model (AR) (Soltani, Boichu, Simard, and Canu 2000), autoregressive integrated moving average (ARIMA) (Zhang 2003), support vector machines (SVM) (Sapankevych and Sankar 2009), artificial neural networks (ANN) (Jang 1993; Zhang 2003), Kalman filter (Cristi and Tummala 2000), and fuzzy logic (Jang 1993). In order to utilize these short-term time series prediction models for long-term time series prediction, there are two approaches: the recursive approach and the direct approach (Herrera, Pomares, Rojas, Guilln, Prieto, and Valenzuela 2007; Nguyen and Chan 2004).

The recursive approach trains one prediction model by optimizing the prediction performance at the next time step, and then iterates the same model using the previously predicted values as part of the input to generate a prediction for a higher horizon. The recursive approach approach suffers from the error propagation problem. On the other hand, the direct approach trains one prediction model for each prediction horizon by optimizing the prediction performance at each prediction horizon. The direct approach needs to train multiple models, so it takes a longer time in the training stage while it avoids the error accumulation problem. The direct approach usually outperforms the recursive approach on the prediction accuracy aspect. A multi-input multi-output local learning (LL-MIMO) approach (Bontempi 2008; Taieb, Bontempi, Sorjamaa, and Lendasse 2009), which is used as one of the comparison approaches in the experiment, predicts the future values as a whole simultaneously. However, it could still be decomposed into multiple independent models, thus it can be considered as a direct approach.

Due to the difficulties that arise in long-term time series prediction, not every model that works for short-term time series prediction would work well in long-term prediction. For example, the autoregressive model is a linear prediction method that attempts to predict the next value based on the previous observations (Soltani, Boichu, Simard, and Canu 2000), which is typically applied to autocorrelated time series data. Because of its linear nature, it is not able to achieve a good prediction precision if the time series contain non-linear components. In the case of long-term time series prediction, the mapping function is usually non-linear. Therefore, the autoregressive model is not preferable. In addition, the Kalman filter is an optimal recursive filter for linear functions subjected to Gaussian noise (Cristi and Tummala 2000). In order to build a suitable Kalman filter, the mechanism which generates the time series should be known, or at least enough information of the mechanism should be available to model the dynamics of the target. For example, the motion of a missile follows Newton's laws of motion, and so the missile tracking problem can be modeled by constructing the relationships among external force, position, velocity, and acceleration of the missile. However, this is not the case for intrauterine pressure tracing. We are not aware of the mechanism of the labor contractions. Meanwhile, the recursive approach has to be applied to conduct long-term time series prediction in the case of the Kalman filter. Error propagation makes the Kalman filter unreliable, especially when the prediction horizon is high. Therefore, the Kalman filter is not included in the comparison experiments.

For long-term time series prediction (Herrera, Pomares, Rojas, Guilln, Prieto, and Valenzuela 2007; Liu and Yao 2009; Maralloo, Koushki, Lucas, and Kalhor 2009; Meng, Dong, and Wong 2009; Sfetsos and Siriopoulos 2004), a lot of efforts

have been placed on deriving variations of least squares support vector machine (LS-SVM) (Huang and Shyu 2010; Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002) and ANN (Yegnanarayana 2004) approaches, which are the fundamental approaches for nonlinear classification and function estimation, and are successfully applied to time series prediction. In order to extend the linear LS-SVM to a nonlinear technique, kernel functions were introduced to implicitly map the input data into a high dimensional feature space, which can have infinite dimensions. The most frequently used kernels in LS-SVM are linear kernel, polynomial kernel, radial basis function (RBF) kernel, and multilayer perceptron (MLP) kernel. The kernel trick extends the LS-SVM theory to a nonlinear technique without an explicit construction of the nonlinear mapping function. Composite kernels (Jiang, Wang, and Wei 2007) have been studied, which combine both global kernel (e.g., polynomial kernel) and local kernel (e.g., RBF kernel) to balance the characteristics of fitting and generalization of these two kernels. Meanwhile, a prior knowledge-based Green's kernel (Farooq, Guergachi, and Krishnan 2007) was used in chaotic time series prediction by using the concept of matched filters. The generality of its application, however, is not proven. To further improve the long-term prediction performance, an input feature selection strategy was also combined with LS-SVM (Sorjamaa, Hao, Reyhani, Ji, and Lendasse 2007) and ANN (Puma-Villanueva, dos Santos, and Von Zuben 2007). By using the features selectively, it better utilizes the information from the past, and reduces the computational complexity of the predictor. However, there have been limited research efforts focusing on choosing the instances adaptively from the training dataset to reduce the input dataset for training a prediction model.

Wavelet methods are widely used in noise removal in both one-dimensional signals and image data. In addition, there has been an increasing interest in wavelet transformation for time series prediction. Many approaches have been proposed for time-series filtering and prediction by combining the wavelet transformation with the prediction models, such as neural networks (Lotric 2004; Menezes and Barreto 2008), Kalman filtering (Cristi and Tummala 2000), and autoregressive models (Soltani, Boichu, Simard, and Canu 2000). Renaud et al. (2005) demonstrated how multiresolution prediction can capture short-range and long-term dependencies by combining a wavelet denoising technique and a wavelet predictive method. Wei and Billings (2006) proposed a direct modeling approach for long-term non-linear time series predictions by introducing the multiresolution wavelet-based non-linear auto-regressive moving average (NARMA) models.

2.3 Knowledge Discovery and Data Mining

The knowledge discovery in databases (KDD) process is usually composed of five stages as shown in Fig. 2.1, including selection, preprocessing, transformation, data mining, and interpretation/evaluation (Fayyad, Piatetsky-shapiro, and Smyth 1996). Data mining is the analysis step of the knowledge discovery in databases process. As an interdisciplinary field of computer science, data mining is the process of discovering new patterns from large datasets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. With the amount of data stored in the databases continuously growing, there is an urge to discover the

valuable hidden knowledge in a large amount of data. Consequently, data mining has been attracting a significant amount of research, industry, and media attention lately.

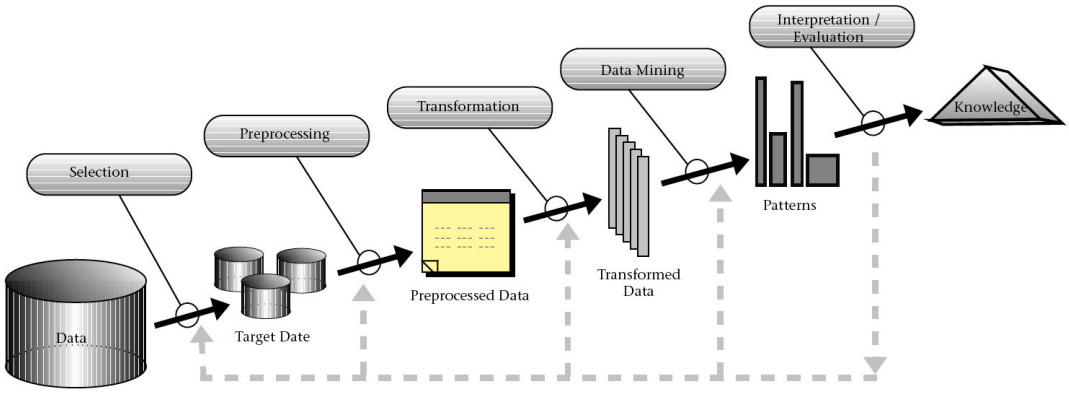

Figure 2.1: An Overview of the Steps That Compose the KDD Process (Fayyad, Piatetsky-shapiro, and Smyth 1996)

A wide variety of data mining methods exist, such as association analysis, classification, clustering, statistical learning, bagging and boosting, sequential pattern mining, integrated mining, rough set mining, link mining, and graph mining (Wu, Kumar, Ross Quinlan, Ghosh, Yang, Motoda, McLachlan, Ng, Liu, Yu, Zhou, Steinbach, Hand, and Steinberg 2008). In this section, we briefly introduce a subset of the techniques that are related to our work, including association rule mining and temporal pattern analysis.

2.3.1 Association Rule Mining

Association rule mining, first introduced by Agrawal et al. (1993), is one of the most important and well studied data mining techniques (Srikant and Agrawal 1996; Jiang

and Gruenwald 2006; Srikant and Agrawal 1995). Its goal is to extract interesting correlations, frequent patterns, associations, or casual structures among sets of items in the transaction databases or other data repositories. Association rule mining is to discover the association rules that satisfy the predefined minimum support and confidence measures from a given database. The problem can be decomposed into two subproblems. The first subproblem is to find those itemsets whose occurrences exceed a predefined threshold (*i.e.*, minimum support) in the database. Those itemsets are called frequent itemsets. The second subproblem is to generate the association rules from those frequent itemsets with the constraints of a minimal confidence threshold. Since the second subproblem is quite straightforward, most of the research has been focusing on the first subproblem. Agrawal and Srikant (1994) proposed an algorithm (named Apriori), which is more efficient during the candidate generation process. Apriori uses pruning techniques to avoid measuring certain itemsets, while guaranteeing the correctness and completeness. The principle of the Apriori algorithm is that if an itemset is frequent, all of its subsets must also be frequent (Agrawal and Srikant 1994).

The main bottlenecks of the Apriori algorithm are 1) the complex candidate generation process that uses most of the time, space, and memory, and 2) the multiple scans of the database. Based on the Apriori algorithm, there have been many new algorithms designed to reduce the computational cost of the association rule mining technique. The computational cost can be reduced 1) by reducing the number of passes over the database, 2) by sampling the database, 3) by adding extra constraints on the structure of patterns, and 4) through parallelization (Kotsiantis and Kanellopoulos 2006). Some representing approaches are introduced below.

The FP-Tree frequent pattern mining algorithm (Han and Pei 2000) passes over the database only twice to generate the frequent itemsets without any candidate generation process. The frequent pattern generation process includes two steps, which are constructing the FP-Tree and generating the frequent patterns from the constructed FP-Tree. The mining result is the same as the Apriori algorithm. FP-Tree is a compressed representation of the original database, because only the frequent items are used to construct the tree, and other infrequent items are pruned. It uses a divide and conquer method that significantly reduces the size of the FP-Tree. However, FP-Tree is not suitable for incremental frequent pattern mining. If there is a new added transaction, it is required to rebuild the whole tree.

For frequent pattern mining on stream data, especially when the stream flows at a faster rate than the processing speed, it is necessary to employ sampling techniques. How to sample the data stream and how to decide the sample rate become the key issues. Chuang et al. (2005) proposed a progressive sampling algorithm, called sampling error estimation (SEE), which aims to identify an appropriate sample size for mining association rules. SEE has two advantages. First, the sample size can be determined without the need of executing association rules. Second, the identified sample size of SEE is accurate, meaning that the association rules can be efficiently executed on a sample of this size to obtain a sufficiently accurate result. This is attributed to the merit of SEE for being able to reduce the influence of randomness by examining several samples with the same size in one database scan.

Usually, the goal of association rule mining is to discover all the patterns whose frequency exceeds a predefined threshold. However, sometimes it is required to add some constraints on the structure of the patterns based on domain knowledge. As-

sociation rule mining should be able to utilize such constraints, so as to speed up the mining process. Wojciechowski and Zakrzewicz (2002) focused on improving the efficiency of constraint-based frequent pattern mining by using the dataset filtering techniques. Dataset filtering conceptually transforms a given data mining task into an equivalent one operating on a smaller dataset. Rapid association rule mining (RARM) (Das, Ng, and Woon 2001) is an association rule mining method that uses the tree structure to represent the original database and avoids the candidate generation process, which is similar to FP-Tree. In order to improve the efficiency, some constraints were applied in RARM during the mining process to generate only those association rules that are interesting to the users, instead of all the association rules.

In order to take advantage of the higher speed and greater storage of the parallel systems, it is required to partition the database among the processors. Accordingly, the association rule mining technique should adapt to the distributed memory system (Schuster and Wolff 2004). An efficient parallel algorithm FPM (Fast Parallel Mining) for mining association rules on a shared-nothing parallel system has been proposed by Cheung and Xiao (1998). It adopts the count distribution approach and has incorporated two powerful candidate pruning techniques, *i.e.*, distributed pruning and global pruning. It has a simple communication scheme which performs only one round of message exchanges in each iteration. Parthasarathy et al. (2001) present a comprehensive literature survey on parallel association rule mining with a shared-memory architecture covering most trends, challenges, and approaches adopted for parallel data mining. All approaches compared in this survey are Apriori-based.

2.3.2 Temporal Pattern Analysis

Temporal pattern analysis is an important topic in data mining. Different from association rule mining, which attempts to find the correlative items occurring simultaneously in one transaction, temporal pattern mining searches correlative items occurring at different time instances (*i.e.*, asynchronously). The existing algorithms in this area can be divided into several categories: sequential association rules (Boonjing and Songram 2007; Jiang and Gruenwald 2006), cyclic association rules (Ozden, Ramaswamy, and Silberschatz 1998), frequent episodes (Wang, Hou, and Zhou 2006), segment-wise periodic patterns (Han, Gong, and Yin 1998), and inter-transaction association rule mining (Han, Lu, and Feng 1998; Tung, Lu, Han, and Feng 2003).

Even though there are temporal components in all these patterns, mining sequential patterns, cyclic association rules, frequent episodes, and segment-wise periodic patterns can also be categorized as intra-transaction association rule mining, in contrast to inter-transaction association rule mining. For example, each sequence is actually taken as one transaction in sequential pattern mining, and then it finds the similar itemsets with timestamps, which satisfy the minimum support requirement. The discovered association rule is intra-transactional. The challenge of inter-transaction rule mining is that it breaks the boundary of the transactions, which leads to an increasing number of potential itemsets and rules. FITI finds the inter-transactional association rules based on intra-transactional associations, and uses a special data structure for efficient mining inter-transactional frequent itemsets (Tung, Lu, Han, and Feng 2003). PROWL uses a projected window method and a depth-first enumeration approach to discover frequent patterns quickly, which has shown to outperform FITI (Huang,

Chang, and Lin 2004). Algorithms for mining follow-up correlation patterns from time-related databases (Zhang, Huang, Zhang, and Zhu 2008; Zhang, Zhang, Zhu, and Huang 2006) generate inter-transactional rules with quantity constraints.

Meanwhile, online mining for frequent itemsets, *i.e.*, mining frequent itemsets over data streams, has received much attention due to the increasing prominence of data streams in a wide range of applications (Cheng, Ke, and Ng 2008). Unlike mining static databases, mining frequent itemsets over data streams poses many challenges, including its one-scan nature, unbounded memory requirement, the high data arrival rate of the data stream, and the combinatorial explosion of the itemsets. Due to these challenges, research studies have been conducted on approximating mining results, along with some reasonable guarantees on the quality of the approximation. Existing algorithms can be generally divided into two categories based on the adopted window model: the landmark window (Chang and Lee 2003; Giannella, Han, Pei, Yan, and Yu 2003; Lee and Lee 2005) and the sliding window (Chang and Lee 2004; Chi, Wang, Yu, and Muntz 2004). The sliding window model captures recent pattern changes and trends. However, performing the update for each incoming and expiring transaction is usually much less efficient than batch-processing, especially in a large search space. Thus, these methods may not be able to cope with high-speed data streams that involve millions of transactions generated from some real-life applications.

CHAPTER 3

Overview of the Proposed Framework

In this chapter, the detailed structure of the proposed framework is presented. The details of the data preprocessing approach are introduced. The related techniques, including long-term time series prediction and LS-SVM for regression, are then discussed. A novel k-NN based LS-SVM for long-term time series prediction approach and a unique post-prediction process are proposed in Section 3.4 and Section 3.5, respectively. Framework evaluation criteria used in the experiments are included in the end of this chapter.

3.1 Structure of the Proposed Framework

Figure 3.1 shows the system architecture of the proposed prediction framework in detail. The inputs to the framework include a training time series d, historical patient information HI, current patient's demographic and obstetrical information, and a testing time series td. Both the training time series d and the testing time series t are from the preprocessed intrauterine pressure time series of the current patient of interest. The historical patient information HI contains multiple past patients' preprocessed intrauterine pressure time series and their demographic and obstetrical

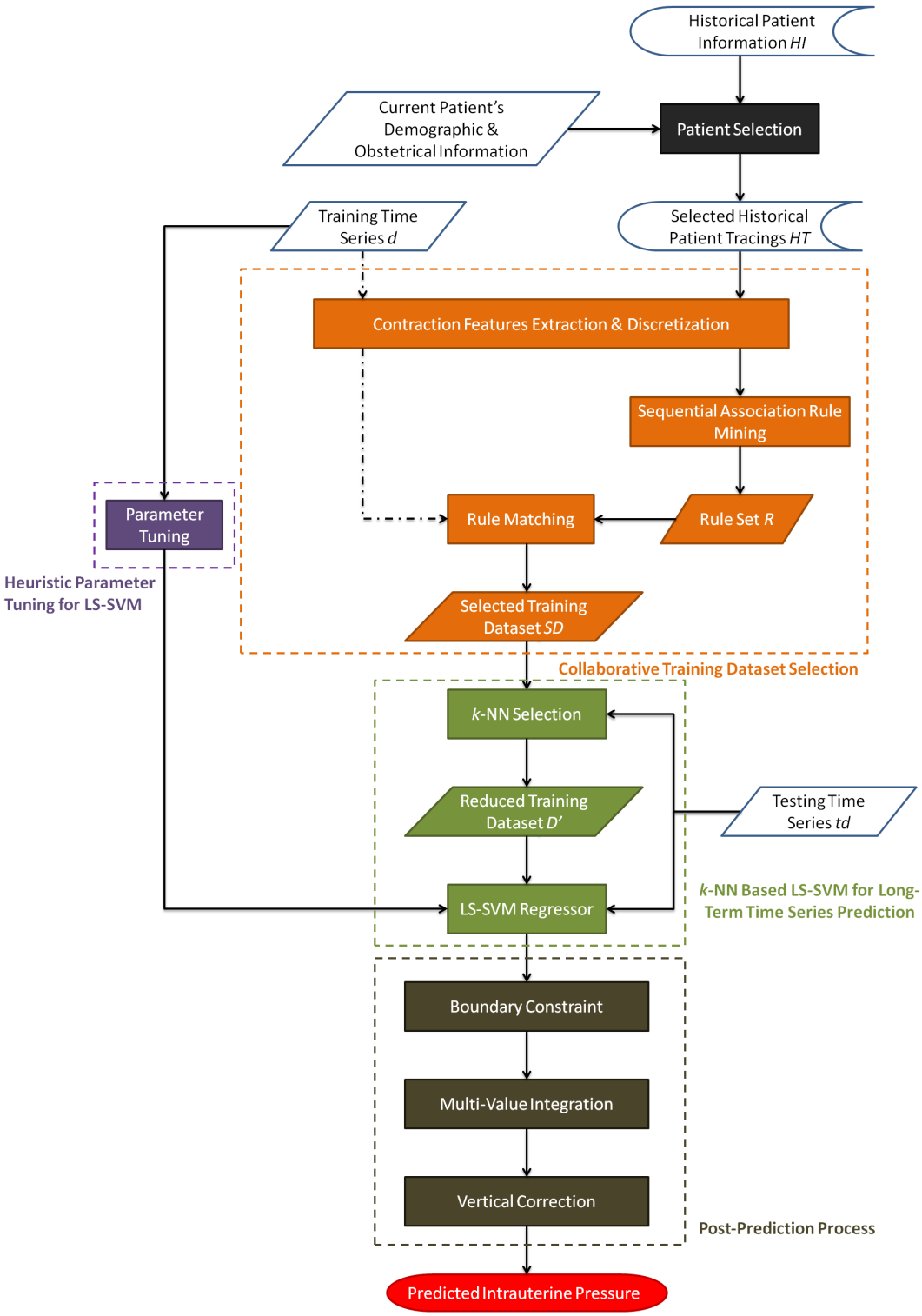

Figure 3.1: System Architecture of the Proposed Framework

information. The output of the framework is the intrauterine pressure values predicted multiple seconds ahead.

The framework mainly contains five parts according to the functionalities: 1) patient selection, 2) collaborative training dataset selection, 3) heuristic parameter tuning for LS-SVM, 4) k-NN based LS-SVM for long-term time series prediction, and 5) the post-prediction process, which includes boundary constraint, multi-value integration and vertical correction components. The five parts are highlighted in different colors and enclosed by the dashed lines. The details of patient selection and collaborative training dataset selection are described in Chapter 4, and heuristic parameter tuning method for LS-SVM is described in Chapter 5. The remaining components are presented in the following sections.

3.2 Inputs to the Framework

The intrauterine pressure signal is measured by the intrauterine pressure catheter as shown in Fig. 3.2, and collected by the QS perinatal clinical information system manufactured by General Electric Company. Usually, an intrauterine pressure tracing is composed of intermittent peaks one after another with rest periods in between. The start of a peak corresponds to the start of a contraction. The time from the beginning to the end of one contraction is called the duration. The time from the beginning of one contraction to the beginning of the next contraction is called the period. Figure 3.3 shows an example of the intrauterine pressure time series. In this example, the duration of the contractions is approximately 1 minute, and the period is approximately 5 minutes.

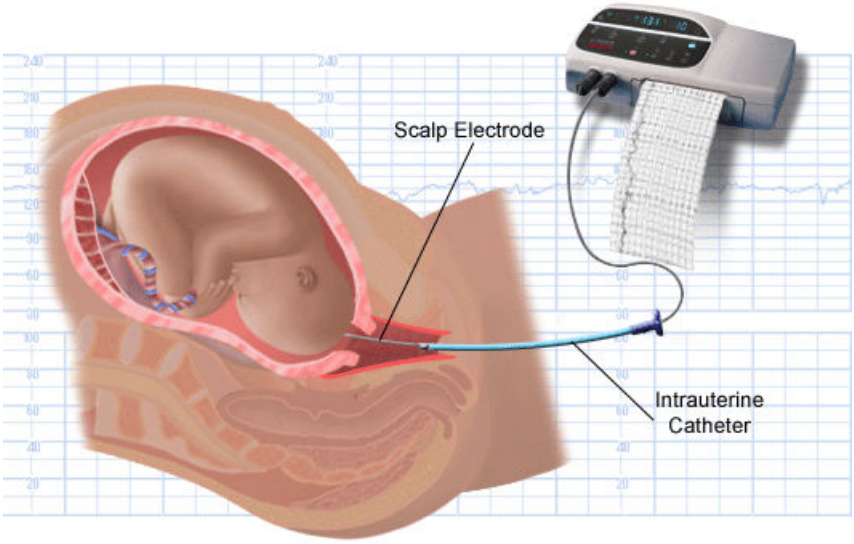

Figure 3.2: Intrauterine Pressure Catheter (Palomar Pomerado Health 2011)

The original intrauterine pressure time series are sampled four times each second within the value range [0,100]mmHg collected by the QS perinatal clinical information system. Due to some system constraints, patients' movements, or other unforeseen interference, the intrauterine pressure time series are often contaminated by noise, and may suffer from loss of data. Poor data quality significantly influences the performance of the prediction. Therefore, a preprocessing step is necessary to reduce the

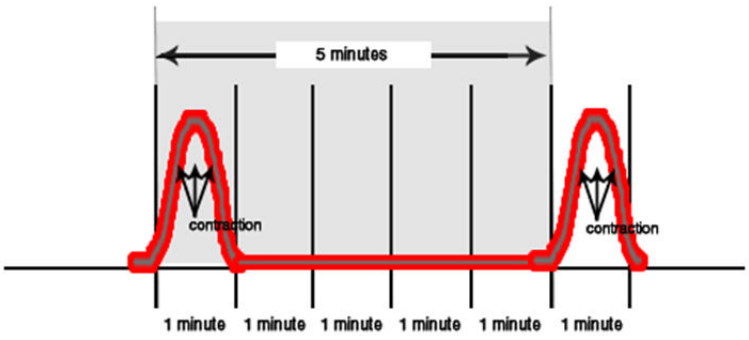

Figure 3.3: Intrauterine Pressure Time Series

noise as much as possible. We subsample the data once each second, adjust all the values that are beyond the given range [0,100]mmHg caused by system imperfections, and propose a new interpolation-based technique to reduce the noise. In this section, we present the details of the denoising method.

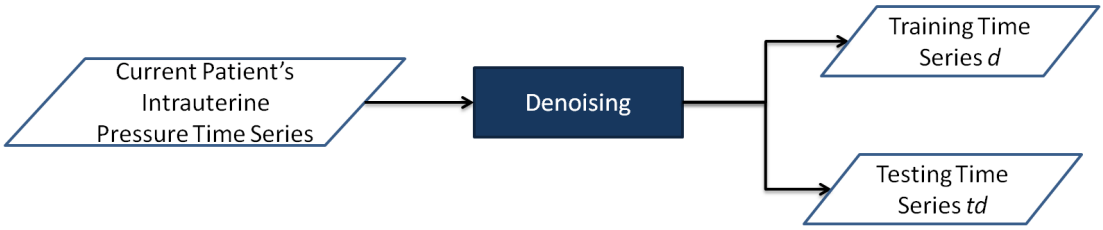

Figure 3.4: Data Preprocessing Step

The pattern of the noise in intrauterine contraction time series varies, making it difficult to filter by a regular low pass filter. In our work, we further subsample the data and detect the peak points, which are very crucial to determine the shape of the curve. Based on these selected points, an interpolation method is applied to rebuild the signal. Two techniques are detailed in the following subsections: 1) peak point detection, and 2) interpolation. As shown in Fig. 3.4, both the training time series d and the testing time series td are derived from the intrauterine pressure time series of the current patient of interest through the denoising process. The intrauterine pressure time series in the HI are also denoised using the same approach.

3.2.1 Peak Point Detection

The following method is designed to locate and measure the positive peaks in a noisy signal introduced by O'Haver (2011). Four parameters should be predefined: $SlopeThreshold$, $AmpThreshold$, $SmoothWidth$, and $FitWidth$, which can be varied

depending on the characteristics of the signal (O'Haver 2011). It detects the peaks by looking for the downward zero-crossings in the smoothed first derivative that exceed $SlopeThreshold$ and the peak amplitudes that exceed $AmpThreshold$, and determines the position, height, and approximate width of each peak by least-square fitting the top part of the peak. The four parameters are defined as follows.

- $SlopeThreshold$: The slope of the smoothed first-derivative that is taken to indicate a peak. This discriminates on the basis of the peak width values. Larger values of this parameter will neglect broad features of the signal.

- $AmpThreshold$: Discriminates on the basis of the peak height values. Any peak with a height smaller than this value is ignored.

- $SmoothWidth$: The width of the smooth function that is applied to the data before the slope is measured. Larger values of $SmoothWidth$ will neglect those small and sharp features. A reasonable value is typically about 1/2 of the number of data points in the half-width of the peaks.

- $FitWidth$: The number of points around the top part of the (unsmoothed) peak that are taken to estimate the peak heights, positions, and widths. A reasonable value is typically about equal to 1/2 of the number of data points in the half-width of the peaks. The minimum value is 3.

In order to measure the peaks in the intrauterine pressure time series that correspond to contractions, we prefer to know the maximum value of the peak. The maximum value is more meaningful than the measured height by least-square fitting,

because the maximum value is the real height of the peak. When the shape of the top part of the peak is irregular or non-parabolic, the calculated height might deviate significantly from the maximum value. Therefore, we modify the peak detection method as follows. Detect the peaks by looking for the downward zero-crossings in the smoothed first derivative that exceed $SlopeThreshold$ and the peak amplitudes that exceed $AmpThreshold$, and determine the position and height of each peak by locating the maximum value of the top part of the peak. In this case, there is no minimum value requirement for $FitWidth$. The detected peak points together with the subsampled points at each s seconds as a whole are the input for interpolation to rebuild the signal.

3.2.2 Interpolation

Interpolation is the process of defining a function that takes on specified values at specified points. The shape-preserving piecewise cubic Hermite interpolation (Mathworks.com 2008) is suitable in our application, and it works better than piecewise linear interpolation, interpolating polynomial, and the piecewise cubic spline interpolation, because it follows the main trend of the time series and preserves the shape of the curve smoothly. Piecewise linear approach cannot provide smooth interpolation. Interpolating polynomial and the piecewise cubic spline interpolation are able to generate interpolation smoothly, but they do not guarantee preserving the main trend.

The key idea of the shape-preserving piecewise cubic Hermite interpolation is to determine the slopes at the given points so that the interpolation function values do

not overshoot the data values. Given N points in the plane, $(x_k,\ y_k)$, $k = 1, \cdots, N$, with distinct x_ks, let h_k denote the length of the kth subinterval:

$$h_k = x_{k+1} - x_k; \tag{3.1}$$

then the first divided difference, δ_k, is given in Equation (3.2):

$$\delta_k = (y_{k+1} - y_k)/h_k. \tag{3.2}$$

Let d_k be the slope of the interpolation at x_k. If δ_k and δ_{k-1} have opposite signs or if either of them is zero, then x_k is a discrete local minimum or maximum, in this case we set $d_k = 0$.This is illustrated in the first half of Fig. 3.5. The lower solid line is the piecewise linear interpolation. Its slopes on both sides of the breakpoint have opposite signs. Consequently, the dashed line has slope zero. The curved line is the shape-preserving interpolation, formed from two different cubics. The two cubics interpolate the center value and their derivatives are both zero. If δ_k and δ_{k-1} have the same sign and the two intervals have the same length, then d_k is set to be the harmonic mean of the two discrete slopes as in Equation (3.3) (Mathworks.com 2008). This is shown in the later half of Fig. 3.5.

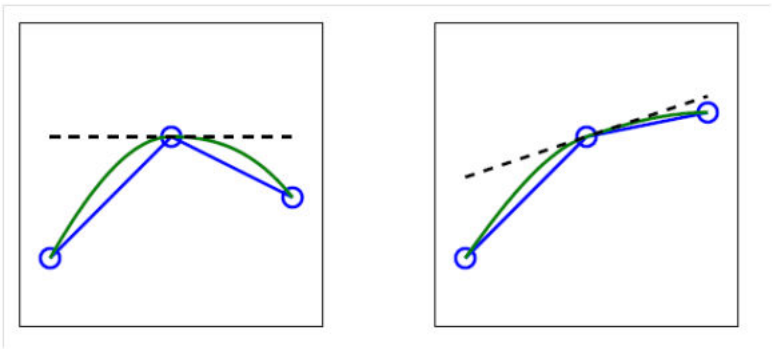

Figure 3.5: Slopes for Shape-Preserving Hermite Interpolation

$$\frac{1}{d_k} = \frac{1}{2}(\frac{1}{\delta_{k-1}} + \frac{1}{\delta_k}) \tag{3.3}$$

If δ_k and δ_{k-1} have the same sign, but the two intervals have different lengths, then d_k is set to be a weighted harmonic mean of the two discrete slopes, with weights determined by the lengths of the two intervals.

$$\frac{w_1 + w_2}{d_k} = \frac{w_1}{\delta_{k-1}} + \frac{w_2}{\delta_k}, \tag{3.4}$$

where $w_1 = 2h_k + h_{k-1}$ and $w_2 = h_k + 2h_{k-1}$. This proposed preprocess step, which combines data subsampling, peak detection and interpolation, filters much noise and preserves the main shape of the curve. It is a necessary step before doing prediction to reduce the influence of the noise and ensure a good performance.

3.3 Related Techniques

In this section, we briefly describe the problem of long-term time series prediction and two strategies to solve the problem. An introduction to the LS-SVM approach is also included.

3.3.1 Long-Term Time Series Prediction

Long-term time series prediction is used to estimate future values multiple steps ahead. In real world applications, it is usually necessary to predict multiple steps ahead, instead of only predicting the value of the next step. In the case of our task, long-term time series prediction should be done to predict the intrauterine pressure values 30 seconds in advance. Prediction is not the ultimate goal, and it is commonly

followed by decision making based on the predicted results. Decision making and the corresponding action take time and might have some delays on their effects. The capability of looking multiple steps ahead enables proactive reaction. Therefore, long-term time series prediction is more useful and more important than the short-term time series prediction. However, long-term time series prediction also brings more difficulties because of the increasing uncertainty and the lack of information about the future trend. Methods for long-term time series prediction can be broadly categorized into two trends: recursive prediction strategy and direct prediction strategy (Herrera, Pomares, Rojas, Guilln, Prieto, and Valenzuela 2007). In this section, we provide a detailed description of these two strategies.

Recursive Prediction Strategy

A time series A is a sequence of data points a_t, usually collected at uniform time intervals consecutively. Each data point, a_t, is an observation at time t. Long-term time series prediction is to predict the future n values based on the previous p observations, where $n > 1$ and $p \geq 1$. Recursive prediction strategy first trains a one-step ahead prediction model, f_1, as shown in (3.5), and then iterates the same model multiple times taking the previously predicted values as a part of the input. For example, $\overline{a}_{t+2}$ is calculated in (3.6).

$$\overline{a}_{t+1} = f_1(a_t, a_{t-1}, \ldots, a_{t-p+1}); \tag{3.5}$$

$$\overline{a}_{t+2} = f_1(\overline{a}_{t+1}, a_t, \ldots, a_{t-p+2}). \tag{3.6}$$

Here, $\overline{a}_{t+1}$ is an estimated value, and its true value a_{t+1} is not available at time t. Iterate f_1 in the similar manner to predict $\overline{a}_{t+n}$, which is n-step ahead prediction.

This approach is simple and computationally inexpensive, because it only needs to train one prediction model. However, it takes output from the prediction model as part of the input vector repeatedly, so it suffers from error accumulation problem. Especially, when the prediction horizon is higher, the error propagation is more severe. Therefore, the prediction accuracy of this approach is usually low.

Direct Prediction Strategy

Direct prediction strategy (Herrera, Pomares, Rojas, Guilln, Prieto, and Valenzuela 2007) trains one prediction model for each prediction horizon based on the historical data, respectively. For n-step ahead prediction, n models should be trained as described in Equation (3.7). The n models share the same set of past true values as the input, but they use different outputs.

$$\overline{a}_{t+i} = f_i(a_t, a_{t-1}, \ldots, a_{t-p+1}), 1 \leq i \leq n. \tag{3.7}$$

The computational cost of the direct prediction strategy is higher, because it needs to train n different models. Its merit is that there is no error accumulation problem, so the prediction accuracy of this approach is usually higher than that of the recursive prediction strategy. In the proposed framework, we adopt this direct prediction strategy to achieve a higher prediction accuracy value for long-term time series prediction, *i.e.*, we train n different models to predict n-step ahead.

3.3.2 LS-SVM for Regression

The LS-SVM method is widely used in nonlinear regression and classification (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002). Here, we provide an overview of its

application in solving the nonlinear regression problem. Given a set of training instances $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where N is the number of training instances. Let $x \in \mathbb{R}^p$, $y \in \mathbb{R}$, and $\varphi(x) : \mathbb{R}^p \to \mathbb{R}^{p_h}$ be a mapping function which maps the input vector x, of which the dimensionality is p, to a high dimensional feature space at dimension p_h. The LS-SVM model can be described as: $y(x) = w^T\varphi(x) + b$, where $w \in \mathbb{R}^{p_h}$ is a weight vector and b is the bias.

It is an optimization problem to calculate w and b based on the given training instances. Let e_j be an error variable, which is the approximate error for the jth training instance. Let γ be a regularization parameter, the optimization problem can be formulated in the primal weight space as follows.

$$\begin{aligned} &\min_{w,b,e} \mathcal{J}_P(w, e) = \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{j=1}^{N} e_j^2, \\ &\text{subject to } y_j = w^T\varphi(x_j) + b + e_j, j = 1, \ldots, N. \end{aligned} \tag{3.8}$$

When p_h becomes infinite, it is unsolvable in the primal weight space, because the dimension of w is infinite. The problem can be solved by converting the problem into a dual space. First, construct the Lagrangian as in Equation (3.9).

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}_P(w, e) - \sum_{j=1}^{N} \alpha_j\{w^T\varphi(x_j) + b + e_j - y_j\}, \tag{3.9}$$

where α_j are Lagrange multipliers. The dual problem is then derived as in Equation (3.10) by eliminating the variables w and e under the conditions for optimality.

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{3.10}$$

where $y =< y_1; \ldots; y_N >$, $1_v =< 1; \ldots; 1 >$, $\alpha = < \alpha_1; \ldots; \alpha_N >$, and the kernel function is given in Equation (3.11).

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j), \quad i, j = 1, \ldots, N. \tag{3.11}$$

Then the estimated function can be derived as in Equation (3.12).

$$y(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j) + b. \tag{3.12}$$

The kernel trick enables us to map the input vector into a huge dimensional feature space without explicitly computing in that space. The kernel function should satisfy Mercer's condition (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002). The most commonly used kernel functions are linear kernel, polynomial kernel, radial basis function (RBF) kernel, and multilayer perception (MLP) kernel as described below.

- **Linear kernel**: $K(x_i, x_j) = {x_i}^T x_j$.
- **Polynomial kernel**: $K(x_i, x_j) = ({x_i}^T x_j + t)^m$, $t \geq 0$, where t is the intercept and m is the degree of the polynomial.
- **RBF kernel**: $K(x_i, x_j) = exp(-\|x_i - x_j\|_2^2/\sigma^2)$, where σ is the variance of the Gaussian kernel.
- **MLP kernel**: $K(x_i, x_j) = tanh(s{x_i}^T x_j + \theta)$, where s and θ are tuning parameters.

In our study, the RBF kernel is used because of its suitability in non-linear modeling and the time series prediction application. There are two tuning parameters, γ and σ, which need to be decided. γ is the regularization factor, which determines the trade-off between the training error minimization and smoothness of the estimated

function. σ is the bandwidth of the Gaussian kernel. The values of these two parameters are usually selected empirically in different ways. In Chapter 5, we propose a novel heuristic method of tuning these two parameters utilizing information extracted from the training time series. More details about the LS-SVM method can be found in (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002).

3.4 k-NN Based LS-SVM Method

The most common way of training a prediction model is to use the entire available training dataset as the input and treat every instance in the dataset equally. It is observed that training from the instances, which have similar inputs, could render a better model, and it better captures the correlation between the inputs and the corresponding outputs. Therefore, instead of using the whole training dataset, we bring in the k-NN method to dynamically select training instances from the selected training dataset SD, which are closer to the testing instance, to train an LS-SVM model for each testing instance. To our best knowledge, this is the first attempt to do instance selection using the k-NN method in long-term time series prediction modeling. The proposed approach is a lazy algorithm. A prediction is made only after a testing instance comes. We propose a new distance function, which incorporates the Euclidean distance and the dissimilarity of the trend of a time series, to calculate the distance of the instances (Huang and Shyu 2012). Figure 3.6 shows the structure of the proposed k-NN based LS-SVM for long-term time series prediction.

The inputs to this component are the selected training dataset SD generated by the collaborative training dataset selection component and the testing time series td.

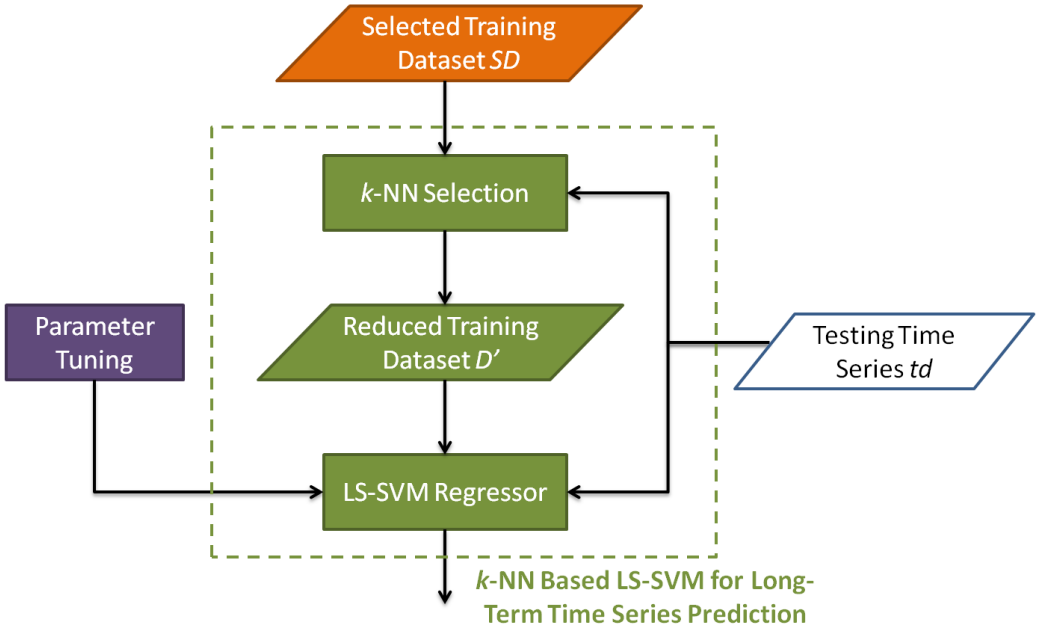

Figure 3.6: k-NN Based LS-SVM Long-Term Time Series Prediction

The output of this component is the predicted values for the testing time series. The k-NN based LS-SVM for long-term time series prediction component contains four steps.

1. Step one: Obtain the hyper-parameters tuned by the method described in Chapter 5 using the first set of the training dataset. We obtain the regularization factor γ and kernel parameter σ, which are customized according to the information extracted from the current time series, prior to doing time series prediction. The tuning process is done only once.

   The parameter tuning for LS-SVM is crucial. Both kernel parameter and regularization parameter impact the performance significantly. However, no consensus has been reached so far regarding how to decide the reasonable searching range and how to adaptively locate the optimal combination of the parameter

values for different datasets. In Chapter 5, we propose a heuristic way of tuning the parameters of LS-SVM with RBF kernel for regression modeling, which include the kernel parameter σ and the regularization factor γ.

2. Step two: k-nearest neighbors are selected from the training dataset SD for each instance in the testing dataset, using a proposed distance measure, which is detailed in the rest of this section. The distance measure considers both the Euclidean distance and the dissimilarity of the trend between the time series. The trend of a time series is described by a vector which is the first order difference of the time series. The selected training dataset SD is updated for each testing instance. This strategy makes sure that the model is adaptive to the latest pattern of the time series. The selected k instances form the training dataset D' for LS-SVM method.

   The integration of the k-NN approach is based upon the assumption that similar inputs commonly share a closer model to correlate with the corresponding outputs. Instead of using the whole training dataset to train an LS-SVM model, it would be more precise and prudent to train a prediction model from the instances in the training dataset that are closer to the testing instance. For each testing instance, the prediction model is re-trained, which makes it more adaptive than using a constant model. Meanwhile, k is usually much smaller than the number of instances in the training dataset. By selecting only those k training instances that are closest to the testing instance, it reduces the size of the input data for training the LS-SVM regressor, thus dramatically decreacing the complexity of building the LS-SVM regressor. Even though the prediction

model needs to be re-trained for each testing instance, the time consumption is still acceptable, and this claim has been validated by experimental results.

3. Step three: Take the reduced training dataset D' to train an LS-SVM regressor for each prediction horizon respectively based on the direct prediction strategy described in Section 3.3.1. For an n-step ahead time series prediction problem, n regressors would be trained. n regressors are taking the same set of the previous true values as the inputs, while generating different outputs according to the prediction horizon.

4. Step four: Take the testing instance as the input to the obtained n LS-SVM regressors and retrieve n predicted values. This step is fast and straightforward. The testing instance is utilized first to search for its k nearest neighbors from the training dataset to form a reduced training dataset, and then it is utilized here as the input to the LS-SVM regressors. As it is shown in Fig. 3.6, there are two flows coming from testing time series td towards k-NN and LS-SVM regressor components respectively.

The length of the input vector in each instance is p, while we use the most recent p_1 values as the input to this k-NN selection component, where $p_1 \leq p$. Similarly, we use the most recent p_2 values as the input to the following LS-SVM regressor component, where $p_2 \leq p$ and $p = MAX(p_1, p_2)$.

Let the jth row in SD be $< x_j, x_{j+1}, \ldots, x_{j+p-1}, x_{j+p}, \ldots, x_{j+p+n-1} >$, where $1 \leq j \leq size(SD)$, n is the prediction horizon. The first p values in the row is the corresponding input vector: $< x_j, x_{j+1}, \ldots, x_{j+p-1} >$. The input vector to k-NN

selection component is $< x_{j+p-p_1}, x_{j+p-p_1+1}, \ldots, x_{j+p-1} >$, which is the last p_1 values of the input vector to the system. Its first order difference is

$$< z_{j+p-p_1}, \ldots, z_{j+p-2} >=< x_{j+p-p_1+1} - x_{j+p-p_1}, \ldots, x_{j+p-1} - x_{j+p-2} > . \tag{3.13}$$

The size of $< z_{j+p-p_1}, \ldots, z_{j+p-2} >$ is $1 \times (p_1 - 1)$.

Given a testing input vector $< x_{T+1}\ , \ldots, x_{T+p-1}, x_{T+p} >$, which starts at time point $T+1$. The last p_1 values will be used here, *i.e.*, $< x_{T+p-p_1+1}\ , \ldots, x_{T+p-1}, x_{T+p} >$. We first calculate its Euclidean distance with each training instance, denoted as $E(j)$ according to Equation (3.14).

$$E(j) = \sqrt{(x_{T+p-p_1+1} - x_{j+p-p_1})^2 + \ldots + (x_{T+p} - x_{j+p-1})^2}. \tag{3.14}$$

The first order difference of the testing input vector is $< z_{T+p-p_1+1}\ , \ldots, z_{T+p-1} >=< x_{T+p-p_1+2} - x_{T+p-p_1+1}, \ldots, x_{T+p} - x_{T+p-1} >$, whose size is $1 \times (p_1 - 1)$ as well. Calculate the Euclidean distance between the differential testing input vector and each differential training input vector, denoted as $F(j)$, according to Equation (3.15)

$$F(j) = \sqrt{(z_{T+p-p_1+1} - z_{j+p-p_1})^2 + \ldots + (z_{T+p-1} - z_{j+p-2})^2}. \tag{3.15}$$

The length of vector E and vector F is $size(SD)$, which varies from patient to patient. We use a combination of the normalized E and F as the distance metric for the k-NN method. The distance Dis is defined by Equation (3.16).

$$Dis(j) = \frac{E(j) - MIN(E)}{MAX(E) - MIN(E)} + \frac{F(j) - MIN(F)}{MAX(F) - MIN(F)}, \tag{3.16}$$

where $MAX(E)$, $MIN(E)$, $MAX(F)$, and $MIN(F)$ are the maximum and minimum values for E and F, respectively. k instances corresponding to the smallest distance measures are selected as a reduced training dataset for LS-SVM.

Here, we give a simple example to demonstrate how to select the instances. Let $p = 2$ and $n = 3$, which means that it takes the previous 2 values as the input, and predict 3 step ahead. The selected training dataset SD is given in Table 3.1.

Table 3.1: An Example for Instance Selection

1	2	4	6	5
2	4	6	5	3
4	6	5	3	1
6	5	3	1	1

Set $p_1 = 2$ and $k = 2$. Given a testing input vector: $< 2, 3 >$, $E(j)$ is calculated, *i.e.*, the Euclidean distance between the testing input vector and each training instance.

$$E(1) = \sqrt{(2-1)^2 + (3-2)^2} = \sqrt{2},$$
$$E(2) = \sqrt{(2-2)^2 + (3-4)^2} = 1,$$
$$E(3) = \sqrt{(2-4)^2 + (3-6)^2} = \sqrt{13},$$
$$E(4) = \sqrt{(2-6)^2 + (3-5)^2} = \sqrt{20}.$$

The first order difference of the testing input vector is $< 3-2 >=< 1 >$. Calculate $F(j)$, *i.e.*, the Euclidean distance between the differential testing input vector and each differential training input vector.

$$F(1) = \sqrt{(1-(2-1))^2} = 0,$$
$$F(2) = \sqrt{(1-(4-2))^2} = 1,$$
$$F(3) = \sqrt{(1-(6-4))^2} = 1,$$
$$F(4) = \sqrt{(1-(5-6))^2} = 2.$$

Hence, we have $MAX(E) = \sqrt{20}$, $MIN(E) = 1$, $MAX(F) = 2$, and $MIN(F) = 0$. Calculate $Dis(j)$.

$$Dis(1) = \frac{\sqrt{2}-1}{\sqrt{20}-1} + \frac{0-0}{2-0} = 0.12,$$

$Dis(2) = \frac{1-1}{\sqrt{20}-1} + \frac{1-0}{2-0} = 0.5,$

$Dis(3) = \frac{\sqrt{13}-1}{\sqrt{20}-1} + \frac{1-0}{2-0} = 1.25,$

$Dis(4) = \frac{\sqrt{20}-1}{\sqrt{20}-1} + \frac{2-0}{2-0} = 2.$

Given $k = 2$, so 2 instances corresponding to the smallest distance are selected as the reduced training dataset. In this example, the first two instances, with distance measure at 0.12 and 0.5, respectively, are selected as the training dataset to train LS-SVM.

3.5 Post-Prediction Process

The post-prediction process contains three components: boundary constraint, multi-value integration and vertical correction components, as shown in Fig. 3.7. They are used to post-process the prediction results, which further improve the prediction precision, and make sure that the results are valid.

3.5.1 Boundary Constraint Component

When the prediction horizon is very large, the prediction model might not be able to capture the changing patterns of the time series very well. The prediction model is fitted to the training data, and by carrying on the same patterns it might generate a value which is too small or too large. For some real world time series data, their values should fall within a certain reasonable range. Also, we do not expect sharp pulses in the prediction result. Therefore, a boundary constraint component is used to bound the predicted values rendered by LS-SVM regressors. We set the upper and lower bounds, denoted as UpB and $LowB$, respectively, based on the values in the

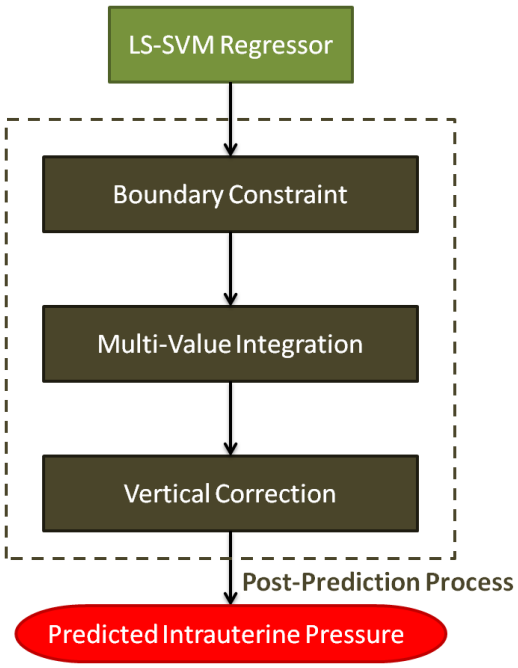

Figure 3.7: Post-Prediction Process

training time series d.

$$UpB = MAX(d) + 0.02 \times STD(d); \tag{3.17}$$

$$LowB = MIN(d) - 0.02 \times STD(d). \tag{3.18}$$

Here, $MAX(d)$ and $MIN(d)$ are the maximum and minimum values of the training time series d, and $STD(d)$ is the standard deviation of d. Thus the boundaries are set to the maximum and minimum values of the training time series data with a moderate extension, where 0.02 is selected empirically. If a predicted value is larger than the upper bound or smaller than the lower bound, we reset the value to the upper bound or lower bound value, respectively. Otherwise, the predicted value is rendered as the final output. Therefore, the final output of the system is limited to the range of $[MIN(d) - 0.02 \times STD(d), MAX(d) + 0.02 \times STD(d)]$. The boundary

is set dynamically according the updating training time series data, and it is domain knowledge free.

3.5.2 Multi-Value Integration Component

There are some existing studies on combining prediction results from individual models. Due to fact that there is no single prediction model that is able to outperform all other methods on any time series, a combination of multiple models becomes the solution to make the prediction model more general, which is able to predict well for a group of time series. There are different opinions on which types of the models should be combined, different methods with distinct nature or very similar models. Some studies claim that these is not much value added in combining the models which are not significantly different, because the models access the same information set and capture similar patterns (Clement and Hendry 1998). On the contrary some other researchers claim that it is also important to combine forecasts from very similar models (Zou and Yang 2004). We agree that even when very similar models are combined, the model uncertainty could be generally reduced. Some of our primary study has validated this claim (Huang and Shyu 2012).

For an n-step ahead prediction problem, the k-NN based LS-SVM component is set to generate a prediction over a horizon n', which is larger than the required prediction horizon n, *i.e.*, $n' > n$. Accordingly, n' regressors are trained in order to predict n'-step ahead. Each predicted value is corresponding to a unique LS-SVM regressor. n' LS-SVM regressors are trained using the same input vectors, but different outputs as described in Equation (3.7). Let the current time be T. The testing input vector is

$< x_{T-p+1}$, ..., x_{T-1}, $x_T >$, and the length of the vector is p. k-NN based LS-SVM component generates n' predicted values, which can be denoted as $< \overline{x}_{T+1}$, ..., $\overline{x}_{T+n}$, ..., $\overline{x}_{T+n'} >$, where the prediction value at time $T+n$ is the expected output. The prediction for the value at time $T+n$ is also conducted in the previous $n'-n$ testing instances. Instead of simply returning $\overline{x}_{T+n}$ as the output for prediction at horizon n, we integrate the previous h predictions of the time series value at time $T+n$, where $0 \leq h \leq n'-n$, together with the current prediction $\overline{x}_{T+n}$ by using an autoregressive model. The autoregressive model takes in total $h+1$ predicted values as input and generates the final output. In order to better illustrate how the approach works, an example is given in Table 3.2.

Table 3.2: An Example for Multi-Value Integration Component

1	x_1	x_2	$\overline{x}_3$	$\overline{x}_4$	$\overline{x}_5$	$\overline{x}_6$
2	x_2	x_3	$\overline{x}'_4$	$\overline{x}'_5$	$\overline{x}'_6$	$\overline{\boldsymbol{x}}'_{\boldsymbol{7}}$
3	x_3	x_4	$\overline{x}''_5$	$\overline{x}''_6$	$\overline{\boldsymbol{x}}''_{\boldsymbol{7}}$	$\overline{x}''_8$
4	x_4	x_5	$\overline{x}'''_6$	$\overline{\boldsymbol{x}}'''_{\boldsymbol{7}}$	$\overline{x}'''_8$	$\overline{x}'''_9$

In this example, the first column is the instance index. The values of the parameters are $p=2$, $n=2$, $n'=4$, and $T=5$, which means that the prediction takes the past 2 values ($p=2$) as the input vector, and predicts the next 4 values ($n'=4$), while the goal is to predict the next 2 values ($n=2$). The current testing instance is instance 4, and the input vector is $(x_{T-1}, x_T) = (x_4, x_5)$. h can be an integer within the range of $[0, 2]$. Let the autoregressive model that is employed to combine the predicted values be f_a.

- If $h=0$, it renders the prediction value at time $T+n=7$, *i.e.*, $\overline{x}'''_7$ as the final prediction value at horizon $n=2$.

- If $h = 1$, it integrates the previous one prediction of the time series value at time $T + n = 7$, *i.e.*, $\overline{x}_7''$, with the current prediction $\overline{x}_7'''$ as the final output, which is $f_a(\overline{x}_7'', \overline{x}_7''')$.

- If $h = 2$, it integrates the previous two predictions of the time series value at time $T + n = 7$ with the current prediction $\overline{x}_7'''$ as the final output, which is $f_a(\overline{x}_7', \overline{x}_7'', \overline{x}_7''')$.

The autoregressive model is a linear prediction formula that is commonly used to predict an output of a system based on the previous outputs. In the proposed framework, parameter h determines the order of the autoregressive model. For a given h, the corresponding order of the autoregressive model should be $h + 1$. An autoregressive model of order $h + 1$ is defined in Equation (3.19).

$$x_t = c + \sum_{i=1}^{h+1} \varphi_i x_{t-i} + \varepsilon, \tag{3.19}$$

where $\varphi_1, \cdots, \varphi_{h+1}$ are the parameters of the model, c is a constant and ε is white noise. There are many ways to estimate the coefficients. In the experiments, we estimate the coefficients by minimizing the root mean square error in the training process. The training instances for the autoregressive model are formed by the predicted values (as the input) and the corresponding true value (as the output). Take the dataset given in Table 3.2 as an example, and let h be 2. Accordingly, the order of the autoregressive model is 3. For instance, the input of one training instance is ($\overline{x}_6'$, $\overline{x}_6''$, $\overline{x}_6'''$), and the output is the real value at time 6, which is x_6. In reality, there could be many more training instances with the progression of time. The time series is much longer than the one given in this simple example. A set of the training

instances can be used to estimate the coefficients for the autoregressive model. Given an input of a testing instance, for example, ($\overline{x}'_7$, $\overline{x}''_7$, $\overline{x}'''_7$), the trained autoregressive model f_a returns a value as the final prediction for time 7.

The input values to this integration component are all predicted values from the k-NN based LS-SVM method, and thus a linear function is more suitable to combine these values than a non-linear function. We compared autoregressive model with some other methods, such as LS-SVM, nonlinear autoregressive moving average, and the autoregressive model is able to achieve the highest prediction precision.

3.5.3 Vertical Correction Component

The proposed vertical correction component contains two steps. The first step is to smooth the prediction results from the multi-value integration component. According to the proposed approach, the prediction model is retrained at each second. Smoothing can reduce the inconsistency of the models. We employed the preprocessing technique introduced in Section 3.2 to do smoothing, which combines data subsampling, peak detection and interpolation processes.

The second step is to smooth out the sharp pulses and peaks with very low heights. It is based on the domain knowledge that for a normal contraction, the corresponding uterine pressure peak should exceed certain height and duration. Therefore, we ignore the vibrations which are not corresponding to contractions. The peaks are detected using the peak detection method introduced in Section 3.2. Three conditions are set to determine if a peak should be smoothed out or not.

- Condition 1: if the height of a peak is lower than a given threshold, $minHeight$, it is considered as candidate noise.
- Condition 2: if the width of a peak is smaller than a given threshold, $minWidth$, it is considered as candidate noise.
- Condition 3: for the candidate noise that satisfies either of the previous conditions, if the starting point value of the peak is smaller than a threshold, $minSPV$, it is considered as noise.

To smooth out a noise peak, we reset the peak area to be the value of the starting point value of the peak. Finally, the output from the vertical correction component is the final prediction results that the framework generates as the output. They are the sequences of the intrauterine pressure predicted multiple seconds ahead.

3.6 Framework Evaluation Criteria

To evaluate the performance of the prediction models, We employ two error measurements, root mean squared error ($RMSE$) and symmetric mean absolute percentage error ($SMAPE$).

Let X be the real time series, and $\overline{X}$ be the predicted time series obtained at prediction horizon n. The length of both X and $\overline{X}$ is m. $RMSE$ is the square root of the variance, which is defined in Equation (3.20). $RMSE$ is closely related to the value range of the time series data.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{m} (x_t - \overline{x}_t)^2}{m}}. \tag{3.20}$$

In order to compare the performance over different datasets, we calculate $SMAPE$, which is based on relative errors as defined in Equation (3.21). $SMAPE$ is the mean value of the difference between x_t and $\overline{x}_t$ divided by their average. It has a lower bound and an upper bound at 0 and 2, respectively.

$$SMAPE = \frac{1}{m}\sum_{t=1}^{m}\frac{|x_t - \overline{x}_t|}{(x_t + \overline{x}_t)/2}. \tag{3.21}$$

A fit measure is used as one of the measurements as well to evaluate how fit the prediction is to the real time series. Let the mean value of X be $mean(X)$. The fit measure is defined in Equation (3.22).

$$FIT = 100 \times (1 - \frac{\|X - \overline{X}\|_2}{\|X - mean(X) \cdot 1_v\|_2})\%, \tag{3.22}$$

where $1_v =< 1; \ldots; 1 >$, and the length of 1_v is m. FIT reaches the maximum value (100%) when the prediction $\overline{X}$ exactly matches with the real time series X, *i.e.*, the prediction error $RMSE$ is 0. Other then this ideal situation, FIT is always a number smaller than 100%. A larger FIT value implies a better performance.

The purpose of the prediction is to predict the starting points of the coming contractions, so as to give the signal for analgesia injection ahead of time to relieve the labor pain. Therefore, we also conducted comparison experiments to evaluate the abilities of the prediction models in predicting the starting points of the contractions. In order to make the comparison fair, we applied the post-prediction process to every compared prediction model. Each of the testing time series contains more than one contraction. The prediction for each contraction is assigned an accuracy weight based on the distance between the predicted starting point and the true starting point of the contraction, and then the average of the weights for all the contractions in a testing

time series is calculated as the accuracy measure. Table 3.3 shows the weights for different distance ranges, which are defined based on domain knowledge and empirical studies.

Table 3.3: Weight Distance Mapping Table

Distance (de) (s)	Weight
$0 \leq de < 10$	1
$10 \leq de < 20$	0.8
$20 \leq de < 30$	0.6
$30 \leq de < 40$	0.4
$40 \leq de < 50$	0.2
$de \geq 50$	0

For the error measurements, including $RMSE$ and $SMAPE$, when the values are smaller the prediction models perform better. Contrarily, for the fit and accuracy measurement, when the values are larger, the prediction models perform better.

CHAPTER 4

Patient Selection and Collaborative Training Dataset Selection

In this chapter, we introduce a discretization method for discretizing the numerical contraction features (height and period), describe a sequential association rule mining approach for mining the frequent sequential contraction patterns, present the patient selection strategy, and finally, specify the process of the rule matching based collaborative training dataset selection.

4.1 Discretization

A contraction can be described by the corresponding peak height and period. The features are extracted using the peak detection method introduced in Section 3.2. The contraction pattern is determined by the combination of both the height and the period of a contraction. The aim of sequential association rule mining algorithm is to discover interesting sequential relationships between categorical variables in large databases. The features extracted for contractions are numerical. In order to discover the sequential association relationships among contractions, it is necessary to discretize the features.

The equal-width discretization method is adopted due to the fact that the discretization method should facilitate the association rule mining process. On one hand, there should be a certain number of rules generated. On the other hand, the generated rules should be meaningful and reasonable. Even though the equal-width discretization method is relatively straightforward, it suits our application better than other discretization methods. Table 4.1 and Table 4.2 show the mapping tables used in the experiments, which are determined based on domain knowledge. Contractions detected from both the training time series d and the tracings in the historical patient information HI are discretized using the same mapping table.

Table 4.1: Discretization Mapping Table for Height

Height (ht) range (mmHg)	Nominal marker
$ht < 40$	a
$40 \leq ht < 55$	b
$55 \leq ht < 70$	c
$70 \leq ht < 85$	d
$85 \leq ht < 100$	e
$ht \geq 100$	f

Table 4.2: Discretization Mapping Table for Period

Period (pd) range (s)	Nominal marker
$pd < 25$	A
$25 \leq pd < 85$	B
$85 \leq pd < 145$	C
$145 \leq pd < 205$	D
$205 \leq pd < 265$	E
$265 \leq pd < 325$	F
$325 \leq pd < 385$	G
$pd \geq 385$	H

For the intrauterine pressure time series given in Fig. 3.3, the period is 5 minutes, (*i.e.*, 300 seconds). Consequently, the period is discretized as 'F'. Let the height of

the peak be 75 mmHg, then it is discretized as 'd'. Therefore, the contraction shown in Fig. 3.3 can be described as 'dF'. For consecutive contractions one after the other, they can be represented as a sequence of paired letters in the similar manner, *i.e.*, a discretized contraction feature series.

4.2 Sequential Association Rule Mining Algorithm

A brief introduction about the basic concepts for association rule mining algorithm is first given here. Let $I=\{I_1,\ I_2,\cdots,\ I_m\}$ be a set of m distinct items, T be a transaction that contains a set of items such that $T \subseteq I$, and D be a database with different transaction records Ts. An association rule is an implication in the form of $X \rightarrow Y$, where $X \subset I$ and $Y \subset I$ are sets of items, which are called itemsets, and $X \cap Y = \emptyset$. X is called antecedent/condition, and Y is called consequent. The number of items in an itemset is called the length of an itemset. Itemsets of length m are called m-itemsets. The rule implies that the transactions of the database that contain X tend to contain Y.

There are two important basic measures for association rules, support (sup) and confidence ($conf$). Support determines how often a rule is applicable to a given database, while confidence determines how frequently Y appears in transactions that contain X. Confidence is a measure of the strength of the association rules. sup and $conf$ are calculated using Equation (4.1) and Equation (4.2), respectively.

$$sup(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}; \tag{4.1}$$

$$conf(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}, \tag{4.2}$$

where N is the total number of transactions in the given database, $\sigma(X \cup Y)$ is the number of transactions that contain both X and Y, and $\sigma(X)$ is the number of transactions that contain X. For example, if the support of an item is 2%, it means only 2 percent of the transactions contain this item. If the confidence of an association rule $X \rightarrow Y$ is 80%, it means that 80% of the transactions that contain X contain Y as well. Since the database is usually large, there could be many rules generated. We are only interested in those frequently appeared itemsets and rules with high confidence measures. Therefore, thresholds of support and confidence are predefined to discard those rules that are not so interesting or useful. The two thresholds are minimal support ($minSup$) and minimal confidence ($minConf$), respectively.

In order to discover interesting sequential patterns from the intrauterine pressure time series, we develop our own sequential association rule mining algorithm to cater the needs. In our approach, an item is one contraction, and it contains two letters, representing its height and period, respectively. The sequential association rule mining algorithm takes the discretized contraction feature series derived from a group of historical patient tracings as the input, and returns a set of rules as the output.

We first search for the frequent sequential itemsets for each individual patient separately, called local frequent sequential itemsets, and then generate the overall sequential itemsets, called global frequent sequential itemsets, based on the local frequent sequential itemsets. Three metrics are defined below to evaluate the interest of a sequential association rule in the form of $wX \rightarrow yZ$, where w and y are discretized heights, X and Z are discretized periods.

Definition 1 *supL: the local support. Let N_c be the total number of contractions in one patient's intrauterine pressure tracing. Let $\sigma(wX \cup yZ)$ be the occurrences of the local sequential itemset $wX \cup yZ$. supL is defined in Equation (4.3).*

$$supL(wX \rightarrow yZ) = \frac{\sigma(wX \cup yZ)}{N_c}. \tag{4.3}$$

Definition 2 *supG: the global support. Let M be the total number of patients. Let N_p be the number of patients who have the sequential itemset as one of the local frequent itemsets. supG is defined in Equation (4.4).*

$$supG(wX \rightarrow yZ) = \frac{N_p}{M}. \tag{4.4}$$

Definition 3 *PS: PS measure of a rule $wX \rightarrow yZ$, which is derived from a global frequent sequential itemset. Let $P(wX \cup yZ)$ be the probability of $wX \cup yZ$. Let $P(wX)$ and $P(yZ)$ be the probabilities of wX and yZ, respectively. PS is defined in Equation (4.5).*

$$PS(wX \rightarrow yZ) = P(wX \cup yZ) - P(wX)P(yZ). \tag{4.5}$$

The PS metric was first introduced by Piatetsky-Shapiro (1991). PS is 0 if the variables, wX and yZ, are statistically independent. It is monotonically increasing if the variables occur more often together, and it is monotonically decreasing if one of the variables alone occurs more often. The PS metric is employed here to evaluate the strength of the sequential association rule. Accordingly, three thresholds should be predefined, which are listed below.

Definition 4 *minSupL: minimum support for deciding local frequent sequential itemsets. If the frequency of one itemset in one patient's discretized tracing,* i.e., *supL, is*

no less than minSupL, the itemset is considered as a local frequent sequential itemset of the patient.

Definition 5 *minSupG: minimum support for deciding global frequent sequential itemsets. If the percentage of patients who share one local frequent sequential itemset,* i.e.*, supG, is no less than minSupG, the itemset is considered as a global frequent sequential itemset of the patient.*

Definition 6 *minPS: minimum PS measure. If the PS measure of a sequential association rule derived from a global frequent sequential itemset is no less than minPS, the rule is considered as an interesting rule.*

The proposed sequential association rule mining approach contains three steps. First, generate local frequent sequential itemsets. Given the local support threshold, we first find the items that occur no less than *minSupL* in each patient's tracing. The set of candidate m-itemsets is generated by connecting the frequent (m-1)-itemsets generated in the previous iteration. We bring in the principle of the Apriori algorithm to reduce the computation cost of the iterations, which is that if an itemset is frequent, all of its subsets must also be frequent. It significantly reduces the number of the generated candidate m-itemsets. If the local support of one candidate m-itemset is no less than *minSupL*, it is recognized as a frequent local m-itemset. For sequential association rule mining, the sequence of the occurrence is important. For example, a 2-itemset 'cAdB' is different from the 2-itemset 'dBcA'. Their frequency should be calculated separately. As mentioned earlier, an item in our problem contains two letters, representing the peak height and period, respectively. In the given example,

'cA' represents one contraction, where 'c' specifies the discretized height and 'A' specifies the discretized period. Similarly, 'dB' represents another contraction with height in range 'd' and period in range 'B'.

Second, generate global frequent sequential itemsets. Combine the local frequent sequential itemsets derived from all the historical patient tracings, and generate candidate global frequent m-itemsets based on local frequent m-itemsets. If the global support of one candidate m-itemset is no less than $minSupG$, it is recognized as a frequent global m-itemset.

Third, generate sequential association rules from the frequent global itemsets. In this algorithm, only one item consequent sequential association rules are generated, *i.e.*, one contraction in the consequent part. This is because we are trying to predict the occurrence of the next one coming contraction. For example, for a frequent global 3-itemset 'cAdBcC', we only generate the rule $cA \cup dB \rightarrow cC$, but not the rule $cA \rightarrow dB \cup cC$. Again, the sequence of the items differentiates the rules. For example, $cA \cup dB \rightarrow cC$ is different from $dB \cup cA \rightarrow cC$. If the PS measure of one derived rule is no less than $minPS$, it is recognized as an interesting sequential association rule.

The interesting sequential association rules of various lengths form the rule set R. The rules are sorted according to their PS values in the descending order. A larger PS value means that the rule is more interesting, *i.e.*, when the condition part of the rule occurs, the chance of the consequent part occurring as well is higher.

4.3 Patient Selection

To the best of our knowledge, there are no prior studies that try to analyze how to link demographic or obstetrical features with the uterine contraction pattern in labor. Even though obstetricians have some empirical observation on how the demographic and obstetrical features impact the uterine contraction pattern, systemic and theoretical study is challenging due to the difficulties in formulizing the problem. Sequential pattern analysis on intrauterine pressure tracings provides a way to characterize the pattern of contractions. This enables the analysis on whether and how the demographic and obstetrical features impact the sequential uterine contraction pattern.

Based on domain knowledge, we summarize some demographic and obstetrical features that might have some impacts on the uterine contraction pattern in labor. The features include maternal age, body mass index (BMI), gestational age, number of pregnancies, living children pregnancy history, labor anesthesia, and indication of oxytocin. The meaning of each variable and its possible values are given below.

- The maternal age is a numerical value, and it is stored as an integer in the database where we retrieve the information from. The maternal age determines the contractility of the muscles, so it might influence the uterine contraction pattern as well.

- The BMI is a heuristic proxy for human body fat based on an individual's weight and height. The BMI is a numerical value as well.

- The gestational age is defined as the time elapsed since 14 days prior to fertilization, and it is usually recorded in terms of the number of weeks. For normal pregnancies, women get into labor when the gestational age reaches 38 weeks to 40 weeks. If the gestational age is less than 38 weeks, it is considered as pre-term labor. If the gestational age is larger than 40 weeks, it is considered as post-term labor.

- The number of pregnancies means the number of babies that the parturient is carrying. For example, if a woman is pregnant with twins, the number of pregnancies equals 2.

- The living children pregnancy history indicates if the woman have ever given birth before. The multiparous tend to have different contraction pattern from the primiparous, because the previous pregnancy influences the contractility of the myometrium. Therefore, we differentiate the multiparous from the primiparous.

- For labor anesthesia, most of the women in the study have chosen epidural, and some other chose anesthesia other than epidural, for example, intravenous sedation or no anesthesia.

- Oxytocin can be used for induction or labor augmentation. The amount of oxytocin for induction is much larger than it is for augmentation. Oxytocin promotes and accelerates uterine contraction.

We divide the patients into several groups with different ranges of value for one feature, and then analyze the sequential uterine contraction pattern of each group.

If the sequential uterine contraction patterns of these groups are different, it means that this feature has a strong impact or determines the sequential uterine contraction pattern. Thus the feature is kept for grouping the patients for later analysis; otherwise, the feature is discarded. This process is repeated for each of the considered feature.

Based on the analysis on 611 patients' records, we observe that maternal age, gestational age, labor anesthesia, and indication of oxytocin, have an impact on the sequential uterine contraction pattern, and the impact is more significant than the one generated by the rest of the features. Therefore, these four features are used to select patients for later analysis. Based on the generated sequential uterine contraction pattern and domain knowledge, we use three of the chosen features and design a tree, as shown in Fig. 4.1, to classify the patients in historical patient information HI into 7 groups. The three features are gestational age, labor anesthesia, and indication of oxytocin.

It is believed that induced labor significantly differs from spontaneous labor. A large amount of oxytocin alters how frequently the uterus contracts. The retrieved sequential uterine contraction patterns are also different. Therefore, we choose indication for oxytocin as the root node to separate the patients who had induction from the patients who gave natural birth. For patients who did not experience induction, labor anesthesia is used to differentiate the patients who had epidural from the patients who had analgesic other than epidural. For induced labor, it usually takes a longer time. All the patients who had induced labor chose to have epidural. Therefore, we omit the labor anesthesia node in the left branch. We further divide the groups by gestational age to differentiate pre-term labor (gestational age < 38)

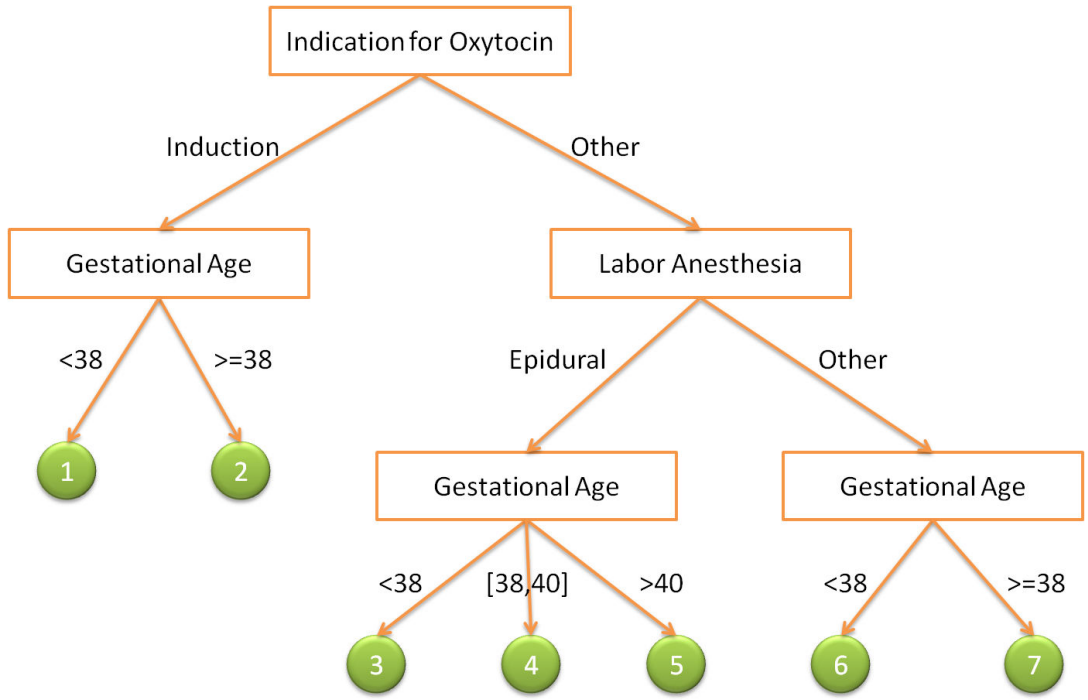

Figure 4.1: A Tree for Patient Selection

from term and post-term labor (gestational age ≥ 38). For patients who gave natural birth and had epidural (majority of patients are in this case), we also separate term labor ($38 \leq$ gestational age ≤ 40) from post-term labor (gestational age > 40). Accordingly, we can divide the patients into 7 groups based on this built tree.

The flowchart for patient selection is shown in Fig. 4.2. The inputs are historical patient information HI and the current patient's demographic and obstetrical information. HI includes multiple past patients' preprocessed intrauterine pressure time series and their demographic and obstetrical information. The patients in HI are divided into 7 groups according to the tree shown in Fig. 4.1. The output is selected historical patient tracing HT. The purpose of patient selection is to select some pa-

tients who share similar demographic and obstetrical features with the current patient of interest. Based on our analysis, the chosen demographic and obstetrical features have a strong impact on the sequential contraction patterns.

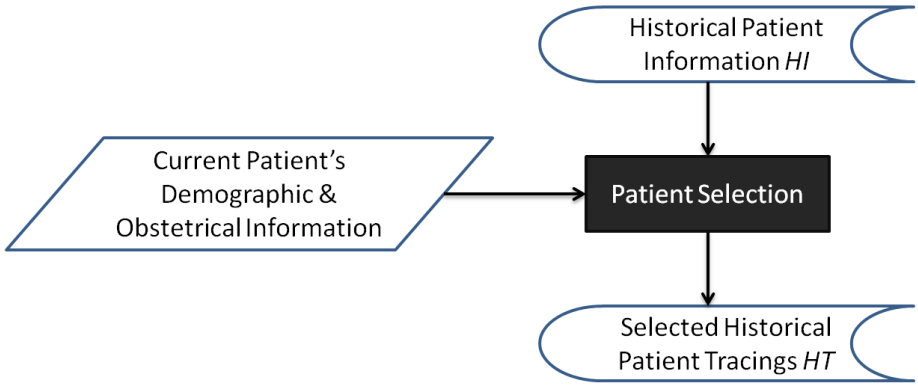

Figure 4.2: Patient Selection Component

First, we determine which group the current patient should belong to according to her gestational age, labor anesthesia, and indication of oxytocin using the tree in Fig. 4.1. Let the current patient belong to group i, where $1 \leq i \leq 7$. Second, select $nPatient$ patients from group i whose age are closest to the current patient's age. Thus, four features in total are used in this selection, which are maternal age, gestational age, labor anesthesia, and indication of oxytocin. The selected patients' intrauterine tracings are the output of this component, and are passed to collaborative training dataset selection component as one of its inputs to form a training dataset for building the prediction models.

4.4 Collaborative Training Dataset Selection

Collaborative filtering (CF) is a technique that uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users (Su and Khoshgoftaar 2009). In our framework, a collaborative filtering component is proposed to use the known sequential contraction patterns of a group of past patients to assist in the prediction of unknown upcoming contraction for other patients. The purpose of the collaborative training dataset selection process is to selectively employ available contractions from both the selected historical patient tracings and the current patient's most recent intrauterine pressure tracing to train the prediction model. This dynamic selection makes the prediction model adaptive to the changing contraction pattern of the patient. Figure 4.3 shows the processes of the collaborative training dataset selection component (enclosed by the dashed lines). This component takes 1) the current patient's training time series d and 2) selected historical patient tracings HT as the inputs, and gives selected training dataset (SD) as the output.

The dataset selection is based on a rule matching process. The rule matching process takes the inputs from two branches as shown in Fig. 4.3. The right branch derives sequential association rules from the selected historical patient tracings HT. For each time series in HT, we detect all the contractions and corresponding peak height and period, and then discretize these features as described in Section 4.1 for the rule mining purpose. The sequential association rule mining method described in Section 4.2 is used to extract the interesting rule set R. In the left branch, training time series d is the intrauterine pressure sequence of the current patient's most recent

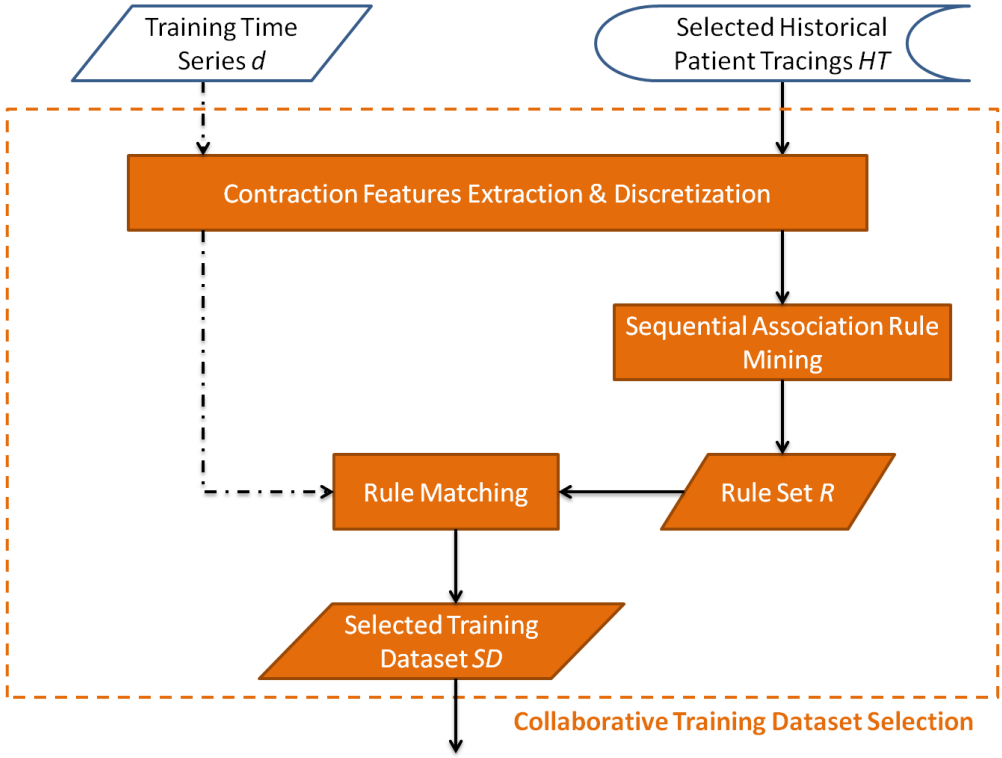

Figure 4.3: Collaborative Training Dataset Selection Component

CN contractions. We measure the peak height and contraction period, and discretize them in the same way as the process in the right branch.

The current patient's recent contraction pattern is compared with the condition part of the generated sequential rules. If there is a matched rule, tracings of the same number (CN) of contractions from the historical patient tracings is selected (which correspond to the consequent part of the rule), and is combined with the current patient's most recent CN contractions tracing in d to form the selected training dataset SD. However, if no matched rule is found, the intrauterine pressure sequence of the current patient's most recent contractions in d is employed as the training dataset SD.

The rules in R are sorted according to their PS values in the descending order. The maximum length of the rules, $maxL$, is dependent on the sequential pattern of the historical patient tracings. Because we always keep one item in the consequent part, accordingly, the maximum length of the rules is $maxL - 1$. On the other hand, there are CN contractions in the training time series d. The rule matching process is to compare the current patient's recent contraction pattern with the condition part of the generated sequential rules. Consequently, the maximum searching range is $min(maxL - 1, CN)$, *i.e.*, the possible maximum length of the condition part of a matched rule is the smaller value between $maxL - 1$ and CN.

The first matched rule is kept for instance selection, and then it stops looking for other matched rules. If no matched rule is found after searching through the whole rule set, the intrauterine pressure sequence of the current patient's most recent CN contractions in d is employed to form the selected training dataset using a sliding window technique. Sliding window technique is used to prepare training and testing instances for the prediction model. Let the length of the time series d be $Tlength$. The sliding window approach is used to re-align the one-dimensional time series into a matrix. The matrix, denoted as SD, is the selected training dataset, where each row in the matrix is one training instance. The size of the sliding window is set to $p + n$, and the size of SD is $(Tlength - p - n + 1) \times (p + n)$, where the first p columns are the input for training the models, the $(p+i)th$ column is the output for the model f_i, and n is the prediction horizon. An individual model is trained for each prediction horizon, respectively to achieve higher prediction accuracy for long-term time series prediction. For example, if we have a training time series $d : \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, where $Tlength = 8$. Set $p = 2$ and $n = 3$. The corresponding matrix SD is shown in

Table 4.3: An Example for Constructing Training Dataset

x_1	x_2	x_3	x_4	x_5
x_2	x_3	x_4	x_5	x_6
x_3	x_4	x_5	x_6	x_7
x_4	x_5	x_6	x_7	x_8

Table 4.3. The size of SD is $(Tlength - p - n + 1) \times (p + n) = 4 \times 5$, where the first 2 columns are the input for training the models, and the $(2 + i)th$ column is the output for the model f_i.

If there is a matched rule, *i.e.*, the current patient's contraction pattern matches with the condition part of one rule, the same number (CN) of contraction tracings from the historical patient tracings are selected (which correspond to the consequent part of the rule), and are combined with the current patient's most recent CN contractions tracing in d to form the selected training dataset SD using the sliding window technique described above in a similar manner. Usually, the number of contraction tracings that follow one interesting sequential association rule is much larger than CN. We calculate the Euclidean distance between the current patient's recent feature vector and each of the feature vectors in the selected historical patient tracings that follows the selected sequential association rule. The feature vector of one contraction contains the normalized height and normalized period of the corresponding peak. The CN contraction tracings with the smallest distance are selected. The selected training dataset SD is then one of the inputs of the k-NN based LS-SVM component for long-term time series prediction modeling.

By using this proposed collaborative filtering approach, we selectively utilize the data from the selected historical patient tracings, and combine them with the current patient's most recent intrauterine contraction tracing. On one hand, a sequential

association rule-based dataset selection enables incorporating the personalized recommendations based on the pattern/knowledge learned from similar patients. On the other hand, using the current patient's most recent intrauterine contraction tracing helps to discover and preserve the current patient's own contraction patterns that are different from the sequential contraction patterns learned from the database. The combination of these two sources facilitates adaptive prediction, and enhances the prediction ability and robustness of the obtained model.

CHAPTER 5

Heuristic Parameter Tuning for LS-SVM

Least squares support vector machines (LS-SVM) with radial basis function (RBF) kernels are proven to be efficient regressors for time series (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002). LS-SVM and its variants are commonly used in time series modeling and prediction (Rubio, Pomares, Rojas, and Herrera 2011; Ormandi 2008). The main bottleneck of this group of approaches is that there are two parameters to be tuned, *i.e.*, the regularization parameter (γ) and the bandwidth (σ) for the RBF kernel. The selection of the parameters for LS-SVM is crucial. Both kernel parameter and regularization parameter may significantly impact the performance. However, it has not reached a consensus yet in how to decide the reasonable search ranges, and how to adaptively find the optimal combination of the parameter values for different datasets. In this chapter, we present a novel heuristic method to tune these two parameters utilizing the information extracted from the training dataset (Huang, Shyu, and Tien 2012). As shown in Fig. 5.1, the heuristic parameter tuning component takes training time series d as the input, and returns the selected parameters to the LS-SVM regressor for modeling.

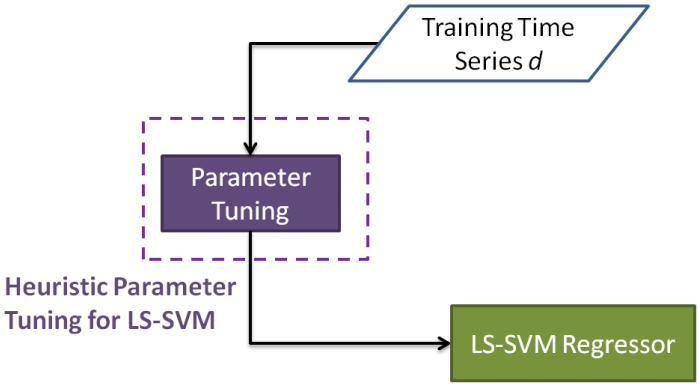

Figure 5.1: Heuristic Parameter Tuning for LS-SVM

5.1 Related Work

As defined in Section 3.3.2, γ is the regularization factor, which determines the trade-off between the training error minimization and smoothness of the estimated function. σ is the bandwidth of the Gaussian kernel. The values of these two parameters are usually selected empirically in different ways. What are the proper value ranges for these two parameters to reach the optimal (or close to optimal) prediction results? How to optimize the values of these two parameters for a specific time series? To the best of our knowledge, no consensus has been reached so far to answer these questions, though there have been some attempts to solve the problems.

Lendasse et al. (2005) propose a method to determine the γ value for a given σ value. A nonparametric noise estimator (Jones 2004) is used to estimate the variance of the noise in the outputs. Lendasse et al. (2005) propose an approach that keeps increasing the value of γ until the training error does not exceed the noise variance estimated by the nonparametric noise estimator. It is believed that a larger γ value leads to a smaller training error. Therefore, when the value of γ is large enough, the

training error could be less than the variance of the noise of the outputs. After the γ value is selected for each given σ value, they calculate the leave-one-out (LOO) error to estimate the modeling error with each parameter pair. The σ value and the corresponding γ value, which minimize the LOO error, are selected. In their approach, both the training error and the LOO error are calculated in two stages to select these two parameters. This way of selecting the γ value is suspected to lead to overfitting to the training dataset. Also, the authors do not specify the starting values for σ and γ, and there is no discussion about how to select the range of σ, and how to increase the value of γ. At the same time, it is a fact that the prediction performance is closely related to the preset range of σ.

Another approach based on the nonparametric noise estimator is that the initial value and the maximum value for the regularization factor γ is suggested to be the ratio of the variance of the output to the estimated variance of the noise, and the minimum γ value is set to 1 (Rubio, Pomares, Rojas, and Herrera 2011). It is an interesting guess of the starting value for γ, and their experimental results demonstrate that the value is a reasonable selection. The authors, however, do not give sufficient explanations on the selection of the range of γ. The suggested range for σ is $[\sigma_{min}, \sigma_{max}]$, where σ_{min} is the minimum distance between two training instances and σ_{max} is the maximum distance between two training instances. However, there is no theoretical analysis on the reason behind this selection. In their paper, the evaluation and comparison are done only experimentally.

The aforementioned studies mainly focus on the selection of the initial values and/or the ranges of the parameters. On the other hand, other studies pay more attention to how to search for the optimum in an efficient way. One trend is to use the

traditional gradient-based approaches (Chang and Lin 2005). Another trend is to employ computational intelligence techniques, such as the particle swarm optimization (PSO) (Guo, Yang, Wu, Wang, and Liang 2008) and evolution approach (Friedrichs and Igel 2004) to determine the hyper-parameters. When there are more than two hyper-parameters to be tuned, the selection of the parameters might be correlated to each other. It is preferable to tune the multiple parameters at the same time. For example, PSO considers a combination of the values of parameters as a point in a hyperspace. It searches for the optima by updating generations and relocating the positions of the points. In this way, all the parameters are updated at the same time, correspondingly. This method is not restricted to any assumptions, and can be generally applied to tune the parameters of different models.

In the following sections, a novel heuristic method is proposed to decide the searching range of the Gaussian kernel parameter based on the information extracted from the training time series. A strategy for efficiently locating the regularization factor is also presented.

5.2 Selection of the Kernel Parameter σ

The kernel trick enables us to work in large dimensional feature spaces without doing explicit computations in the large dimensional spaces (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002). In the case of the RBF kernel, the underlying mapping function actually maps the input vectors into an infinite dimensional space. Even though LS-SVM with the RBF kernel has been commonly used for regression, there are not sufficient studies on how to find a reasonable range for tuning the kernel

parameter σ. Given two input vectors in the training dataset, x and z, the RBF kernel function can be expressed as shown in Equation (5.1) (Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002).

$$K(x, z) = exp(-\|x - z\|_2^2/\sigma^2). \tag{5.1}$$

Let $s = \|x - z\|_2$, which is the Euclidean norm of the vector $(x-z)$, and hence $K(x, z)$ $= exp(-s^2/\sigma^2)$. For a given training dataset, the values of s are bounded in a specific range for any possible combination of x and z values. Assume that the minimum distance between two training instances is s_{min}, and the maximum distance between two training instances is s_{max}. Figure 5.2 is a plot of the kernel function which shows how $K(x, z)$ decreases when s increases.

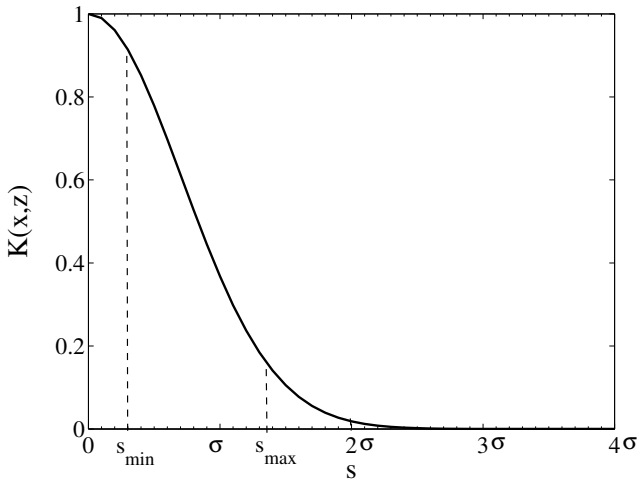

Figure 5.2: RBF Kernel Function

In order to obtain a good regression model for the training dataset as described in Equation (3.12), we propose that the range of s, which is $[s_{min}, s_{max}]$, should be located at a position where the slope of the kernel function is large. The reason for this assumption is that we want to differentiate the instances as much as possible

through using the kernel trick. By differentiating the training instances through the kernel trick, the non-linear regression approach is able to capture more information from the training dataset for modeling. On the other hand, if many training instances are mapped to the flat part of the kernel function, $K(x, z)$ will be a constant, which will not be able to construct a reasonable regression model. Based on the observation that the $K(x, z)$ curve becomes flat when s is larger than 2σ, it is recommended to bound s within 2σ. Thus, we have $s_{max} \leq 2\sigma$. It can be deduced that the slope of the point at $s = 0.0367\sigma$ is the same as the slope of the point at $s = 2\sigma$. Following the same reasoning, we have $s_{max} \geq 0.0367\sigma$. Therefore, a heuristic value range for the kernel parameter σ is shown in Equation (5.2).

$$\frac{s_{max}}{2} \leq \sigma \leq \frac{s_{max}}{0.0367}. \tag{5.2}$$

Meanwhile, through empirical studies, we observe that the value of the optimal kernel parameter is usually closer to $s_{max}/2$ than to its upper bound $s_{max}/0.0367$. Therefore, in the experiments, we select $s_{max}/2$ as the initial value of σ, and then search for the optimal value within the suggested range.

5.3 Selection of the Regularization Factor γ

The regularization Factor γ introduced in Equation (3.8) controls the trade-off between the training error minimization and the smoothness of the model. A larger regularization factor gives more weights to the modeling error in solving the optimal problem. In this case, the corresponding training error will be smaller, while the smoothness of the model is lower. When the value of γ is too large, it leads to the

overfitting problem. On the other hand, if the value of γ is too small, the model will be smoother but it cannot fully capture the relationship between the inputs and outputs in the training dataset, because the training errors are too large. Therefore, it is important to select an appropriate value for γ, so that the estimated model will not overfit the training dataset but being able to model the underlying function.

The problem is how to determine if the model is overfitting to the training dataset, and when the training error is small enough. We do not expect the training error to be zero or very close to zero, because that probably means the model is overfitting to the training dataset. Here, we utilize Gamma test (Stefánsson, Koncar, and Jones 1997) to estimate the variance of the noise σ_r^2. If the signal does not contain noise, the component estimated by the Gamma test can also be considered as a part of the signal that cannot be captured by regression modeling. A brief introduction of the Gamma test is included in Appendix A. We compute leave-one-out (LOO) error to estimate the training error σ_t^2. Leave-one-out cross-validation (LOOCV) involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. It can evaluate the model more generally and avoid overfitting. If the LOO error is larger than the variance of the noise, we increase the value of γ. Otherwise, the value of γ is decreased. γ is updated according to Equation (5.3).

$$\gamma(i+1) = \gamma(i) \times \frac{\sigma_t^2(i)}{\sigma_r^2}, \tag{5.3}$$

where i is the index of the iterations. $\sigma_t^2(i)$ is the LOO error at the ith iteration. According to Equation (3.8), γ is a weight coefficient of the squared training error.

Therefore, a variance ratio-based updating strategy is suitable to quickly locate the optimal value for γ. The searching process stops when the training error changes within 5%. The selection of 5% is another attempt to avoid overfitting, which can be adjusted if necessary. In this way, the trained model is not overfitted to the training dataset, and is able to model the underlying function between the inputs and outputs. The tuning process converges quickly because of utilizing the error feedback to adjust the value of γ. As suggested in (Rubio, Pomares, Rojas, and Herrera 2011), the initial value of γ, denoted as $\gamma(1)$, is given by Equation (5.4).

$$\gamma(1) = \frac{\sigma_y^2}{\sigma_r^2}, \tag{5.4}$$

where σ_y^2 is the variance of the output in the training dataset.

The pseudo-code of the algorithm is shown in Fig. 5.3. The inputs are the training time series d and a set of values for σ within the range $[s_{max}/2, s_{max}/0.0367]$, denoted as $SIGMA$. The outputs of the tuning process are the σ and γ values corresponding to the minimum LOO error. The processes of tuning σ and γ are intertwined, and are based on the training dataset only. Line 1 calculates the variance of the training time series and estimates the variance of the noise. Lines 2-3 initialize the variables *cost* and *gam*. The for loop from line 4 to line 17 is to select a γ value for each value of σ in $SIGMA$, according to Equation (5.3), and the initial value of γ is set to $\frac{\sigma_y^2}{\sigma_r^2}$. In lines 8 and 13, *leaveoneout*($Training\ dataset$, $\gamma(i)$, σ^2) is a function which takes the training dataset as the input, uses $\gamma(i)$ and σ^2 as the hyper-parameters of the LS-SVM method with the RBF kernel, and returns the LOO error. As stated in line 9, when the LOO error is 0, or the LOO error decreases less than 5% comparing to the error measure from the previous iteration, the value of γ to pair with the given

PARAMETER TUNING

1 calculate σ_y^2 and σ_r^2;
2 $cost = [\,]$;
3 $gam = [\,]$;
4 **for** each σ in $SIGMA$
5 $i = 1$;
6 $tcost = 0$;
7 $\gamma(1) = \sigma_y^2/\sigma_r^2$;
8 $\sigma_t^2(i) = leaveoneout(Training\ dataset,\ \gamma(i),\ \sigma^2)$;
9 **while** $((tcost == 0)||((tcost \sim= 0)\&\&(tcost - \sigma_t^2(i))/tcost >= 0.05))$ **do**
10 $tcost = \sigma_t^2(i)$;
11 $\gamma(i+1) = \gamma(i) \times \frac{\sigma_t^2(i)}{\sigma_r^2}$;
12 $i = i + 1$;
13 $\sigma_t^2(i) = leaveoneout(Training\ dataset,\ \gamma(i), \sigma^2)$;
14 **end while**
15 $cost = [cost; \sigma_t^2(i)]$;
16 $gam = [gam; \gamma(i)]$;
17 **end for**
18 find the minimum cost: $cost(j) = min(cost)$;
19 output: $[\sigma, \gamma] = [SIGMA(j), gam(j)]$.

Figure 5.3: Pseudo-Code for Heuristic Parameter Tuning

σ is returned. In lines 18 and 19, the combination of the parameters which gives the lowest LOO error among the paired parameters is returned as the output.

5.4 Comparative Analysis

In this section, we theoretically compare the proposed strategies with some existing approaches: the LS-SVM approach with hyper-parameters tuned by the approach

in (Lendasse, Ji, Reyhani, and Verleysen 2005) (denoted as LS-SVMNNE), the LS-SVM approach with hyper-parameters tuned by the approach in (Rubio, Pomares, Rojas, and Herrera 2011) (denoted as LS-SVMheu), and the LL-MIMO algorithm (Bontempi 2008) (denoted as LL-MIMO).

LS-SVMNNE (Lendasse, Ji, Reyhani, and Verleysen 2005) and LS-SVMheu (Rubio, Pomares, Rojas, and Herrera 2011) are both LS-SVM based approaches with tuned hyper-parameters using different strategies. LS-SVMNNE keeps increasing the value of γ by doubling its previous value until the training error does not exceed the noise variance estimated by the nonparametric noise estimator (Lendasse, Ji, Reyhani, and Verleysen 2005). As analyzed earlier, a larger γ value leads to a smaller training error. The training error could be smaller than the variance of the noise of the outputs when the value of γ is large enough. However, LS-SVMNNE does not provide the initial value for γ, and does not discuss the selection of the range for σ^2, while the prediction performance is closely related to these preset values. Meanwhile, the strategy that increases the value of γ by doubling its previous value might not be appropriate. Given an initial γ value, $\gamma(1)$, let the number of iterations be m, *i.e.*, the searching is terminated after increasing the γ value m times. The updating step size is increased after each iteration, and the maximum deviation of the selected γ value from the true optimal γ value is $2^{m-1}\gamma(1)$, which increases exponentially. Therefore, the deviation from the optimal value can be very large when m is large. The proposed variance ratio-based updating strategy, according to Equation (5.3), is able to locate the optimal value for γ quickly. The number of iterations is expected to be small, which will be validated through experiments. In addition, the updating step size is decreased after each iteration, and it does not suffer from the problem

that the returned value deviates from the optimal value significantly. The limitation of the proposed strategy is that the estimated variance of the noise cannot be 0, *i.e.*, the denominator in Equation (5.3) cannot be 0. This criterion is satisfied by most of the time series.

LS-SVMheu recommends the minimum and maximum values of both γ and σ^2, while LS-SVMheu does not give a solution of finding the optimal values of these parameters (Rubio, Pomares, Rojas, and Herrera 2011). For the regularization factor γ, both the initial and maximum values are suggested to be the ratio of the variance of the output to the estimated variance of the noise (Rubio, Pomares, Rojas, and Herrera 2011), and its minimum value is set to 1. However, there is no sufficient reasoning on this selection of the range of γ. In addition, a fixed minimum γ value for all time series is not preferable. The suggested range for σ is $[s_{min}, s_{max}]$, where s_{min} is the minimum distance between two training instances and s_{max} is the maximum distance between two training instances. However, there is no theoretical basis for this selection. The recommended range for σ by our strategy is $[s_{max}/2, s_{max}/0.0367]$, which is based on the property of the RBF kernel function. The descriptive comparison regarding parameter tuning is summarized in Table 5.1.

LL-MIMO algorithm described in (Bontempi 2008) is a multi-input multi-output local learning approach for long-term time series prediction. It is not an LS-SVM related approach, so it does not have the hyper-parameters that need to be tuned. LL-MIMO predicts the future multiple values as a whole simultaneously using a linear combination. Comparing to the LS-SVM based approaches, LL-MIMO is weaker in non-linear modeling.

Table 5.1: Descriptive Comparison

Methods	γ	σ
LS-SVMNNE	Double the value of γ until the training error does not exceed the noise variance.	N/A
LS-SVMheu	Recommended range: $[1, \sigma_y^2/\sigma_r^2]$.	Recommended range: $[s_{min}, s_{max}]$.
The proposed strategies	Initial value is $\gamma(1) = \sigma_y^2/\sigma_r^2$, and the value updates according to $\gamma(i+1) = \gamma(i) \times \sigma_t^2(i)/\sigma_r^2$.	Recommended range: $[s_{max}/2, s_{max}/0.0367]$.

5.5 Experiment and Results

In order to evaluate the performance of the proposed heuristic parameter tuning method, we performed experiments on the framework shown in Fig. 5.4, which is only part of the overall proposed framework for time-series prediction. The framework shown in Fig. 5.4 takes the tuned parameters to the k-NN based LS-SVM method for long-term time series prediction, and then returns the predicted values as the output. A sliding window technique is applied to prepare the training and testing instances. Three types of datasets are used in the experiments. Section 5.5.1 describes the datasets used in the experiments and the measurement employed to evaluate the performance. Comparisons with four approaches: LS-SVMNNE, LS-SVMheu, LL-MIMO, and the LS-SVM approach with hyper-parameters tuned by the proposed method (denoted as LS-SVM) are shown in Section 5.5.2. The proposed framework is denoted as k-NN based LS-SVM. The experiments were conducted on an Intel Core 2 machine with two 2.66 GHz CPUs and 3.25 GB of RAM.

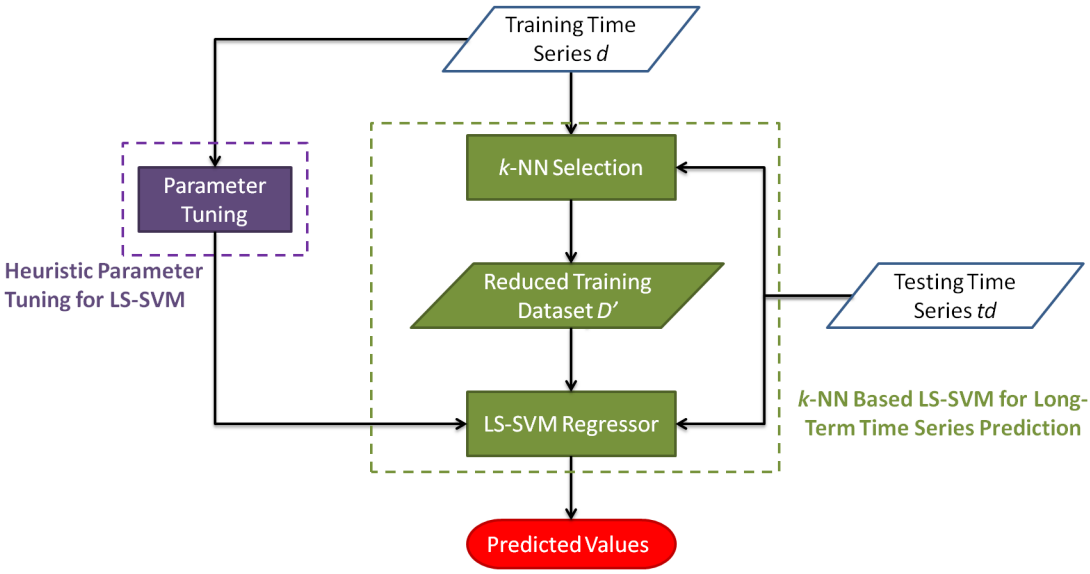

Figure 5.4: Framework for Testing the Heuristic Parameter Tuning Component

5.5.1 Datasets and Error Measurement

Three types of datasets were used in the comparative experiments, including a time series provided by NNGC1 competition, a chaotic laser time series, and a sunspot area data time series provided by National Aeronautics and Space Administration (NASA). The datasets exhibit diverse patterns. Experimental results on intrauterine pressure time series will be shown in Chapter 6.

NNGC1 competition (Crone 2010) provided diverse non-stationary, heteroscedastic transportation time series data with different structures and frequencies. The datasets are frequently used in publications as well. The time series used in our experiments is one of the longest series collected hourly. The length of the time series is 1742. Figure 5.5 is a plot of a segment of the selected time series. It is observed that the time series presents a nearly weekly repeated pattern with the variation in the amplitude.

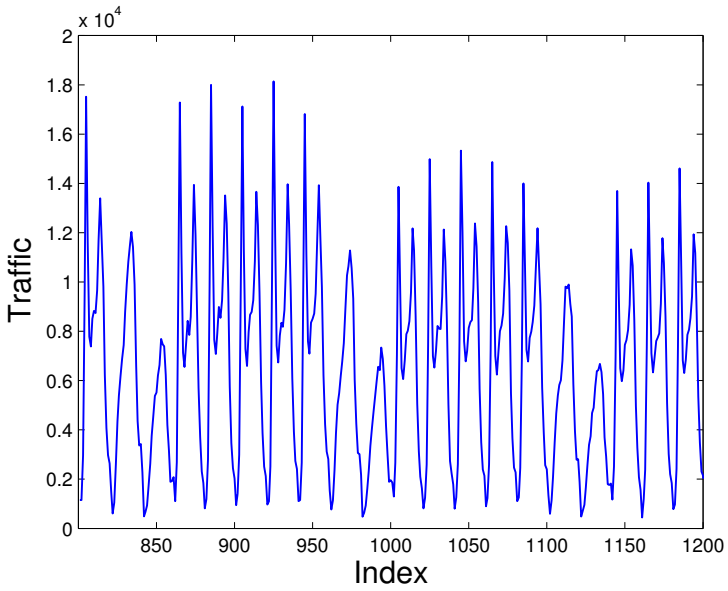

Figure 5.5: A segment of the NNGC1 time series

The chaotic laser time series is a univariate time record of a single observed quantity, measured in a physics laboratory experiment. It comprises measurements of the intensity pulsations of a single-mode Far-Infrared-Laser NH_3 in a chaotic state (Huebner, Abraham, and Weiss 1989). It has been used in the benchmarking studies after the time series prediction competition organized by the Santa Fe Institute. The length of the time series is 1000. Figure 5.6 is a plot of a segment of the chaotic laser time series. The pattern of the chaotic laser time series is composed of gradual variations and sudden changes.

The Royal Greenwich Observatory (RGO) (Hathaway 2011) compiled sunspot observations from a small network of observatories to produce a dataset of daily observations starting in May of 1874. Sunspots appear as dark spots on the surface of the Sun. Sunspots are magnetic regions on the Sun with magnetic field strengths thousands of times stronger than the Earth's magnetic field. The time series used in

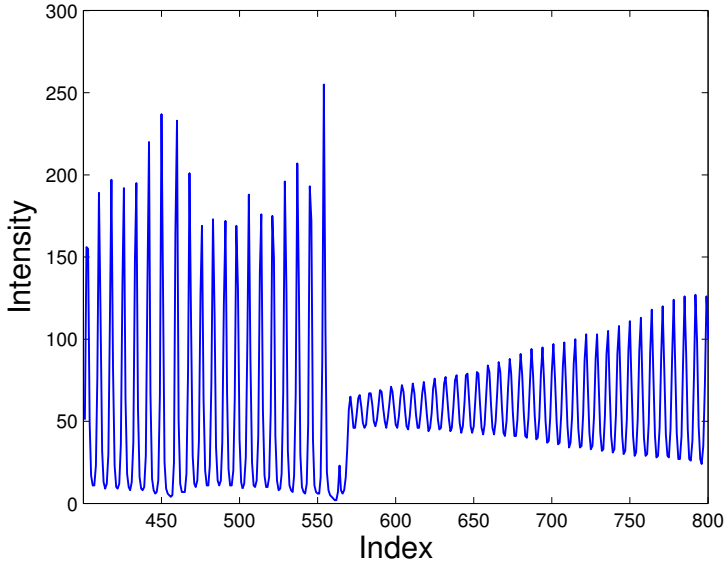

Figure 5.6: A segment of the chaotic laser time series

the experiment contains the monthly averages of the daily sunspot areas in units of millionths of a hemisphere. The length of the time series is 1639. Figure 5.7 is a plot of a segment of the sunspot area time series. This time series contains much more high frequency components comparing to the previous two time series datasets.

We employed root mean squared error ($RMSE$) metrics to evaluate the performance of the prediction models. $RMSE$ is the square root of the variance, which is defined in Equation (3.20).

5.5.2 Experimental Results and Analysis

We compare the proposed framework with four methods, 1) the LS-SVM approach with hyper-parameters tuned by the approach in (Lendasse, Ji, Reyhani, and Verleysen 2005), 2) the LS-SVM approach with hyper-parameters tuned using the parameters selection strategy described in (Rubio, Pomares, Rojas, and Herrera 2011),

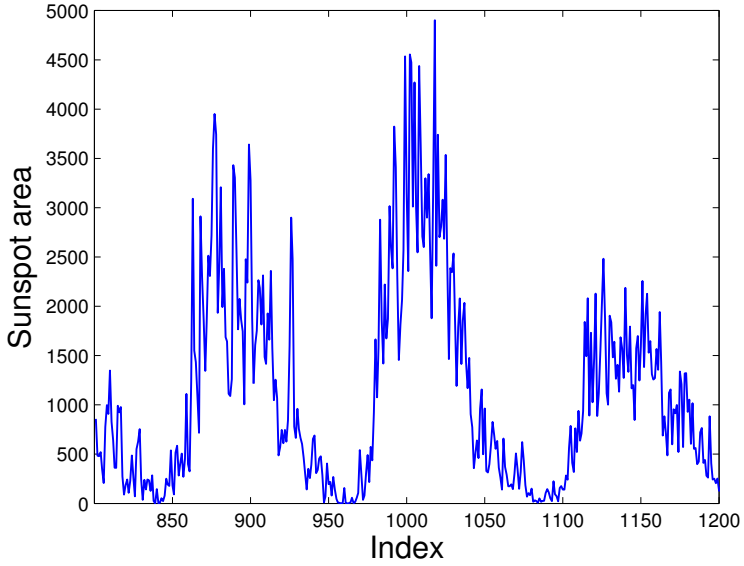

Figure 5.7: A segment of the sunspot area data time series

3) the LL-MIMO algorithm (Bontempi 2008), and 4) the LS-SVM approach with hyper-parameters tuned using the proposed method. Besides the hyper-parameters, there are some other parameters need to be decided for the compared approaches. Even though parameters like the length of the input vector (p) and the number of neighbor points (k) can also influence the prediction performance, the study is mainly focusing on the impact of the hyper-parameters on the prediction accuracy. In order to provide more informative results to evaluate the effect of the proposed heuristic parameter tuning for those hyper-parameters, we only presented results on one set of pre-defined values for the prediction horizon (n), input vector (p), and number of neighbor points (k) in this chapter. Considering the high complexity of the LS-SVM approaches: LS-SVMNNE, LS-SVMheu and LS-SVM, we set the prediction horizon n to 20 for all the comparative experiments. The length of the training time series is set to 600, the length of the input vector p is set to 30. Parameter k in the k-NN

approach is set to 80. Additional tuning process on these parameters will also help to increase the prediction performance, particularly there are plenty of efforts on the input selection, while it is beyond the scope of this study. We select the values of these parameters based on our empirical study.

LS-SVMNNE, introduced in (Lendasse, Ji, Reyhani, and Verleysen 2005), employs bisection method to search for the optimal γ for a given σ^2, but the authors did not provide the initial value for γ, and did not discuss the selection of the range for σ^2. In the experiments, we set the initial value of γ to 1, and the searching range of σ^2 to $[50, 500]$ with a step size at 50. LS-SVMheu presented in (Rubio, Pomares, Rojas, and Herrera 2011) gives the minimum and maximum values of both γ and σ^2, while the authors did not give a solution of finding the optimal values of the parameters. For the comparison purpose, the value of σ^2 is set to the mid-point of the range recommended, which was suggested by the experiments in (Rubio, Pomares, Rojas, and Herrera 2011). The value of γ is set to the recommended initial value, because the authors proved that the initial value is very close to the optimal value. LL-MIMO algorithm described in (Bontempi 2008) is a multi-input multi-output local learning approach for long-term time series prediction. It is not an LS-SVM related approach, so it does not have the hyper-parameters that need to be tuned. LL-MIMO predicts the future multiple values as a whole simultaneously. The comparison with the LS-SVM approach with hyper-parameters tuned by the proposed method is also included.

The comparison results and the tuned hyper-parameters of the NNGC1 time series at prediction horizon $n = 20$ are reported in Table 5.2. There are 1123 testing instances used in the experiments. For LL-MIMO approach, it does not have hyper-

parameters, so its corresponding γ and σ^2 values are not available, and are denoted as N/A in the tables. As we could notice from Table 5.2, the proposed k-NN based LS-SVM framework achieved a very low $RMSE$. Its performance is 61.05% higher than LS-SVMheu, 43.52% higher than LL-MIMO, and 38.81% higher than LS-SVM algorithm. To give a closer look, a part of the prediction results are shown in Fig. 5.8. Because the space is limited, only 25 consecutive points are plotted to ensure the quality of the figure. It can be observed that the predicted curve is very close to the real time series. However, the results predicted by LS-SVMhue and LL-MIMO are away from the real time series, especially when the curve is reaching the peaks.

Table 5.2: Prediction Results for NNGC1 Time Series

Methods	$RMSE$	γ	σ^2
LS-SVMNNE	1675.3	1024	450
LS-SVMheu	2687.2	11.8	23191.5
LL-MIMO	2394.7	N/A	N/A
LS-SVM	2316.1	1106.8	22772.0
k-NN based LS-SVM	**1668.5**	1106.8	22772.0

Figure 5.9 shows the prediction errors of the compared five approaches. The prediction error plotted is the difference between the predicted values and the real values. Prediction errors of LS-SVMhue and LL-MIMO significantly deviate from 0. Even though the tuned σ^2 values for the proposed framework and LS-SVMheu are close, the values of γ differ a lot. Based on the results, we can deduce that LS-SVMheu is under-learning in this case, which means the selected γ value is too small for the NNGC1 time series. For LS-SVM related approaches, it is possible to have multiple local minima. The tuned hyper-parameters for LS-SVMNNE are very different from that of the proposed framework, but their prediction performance is very close, even though the prediction models are different. The results also show that by including

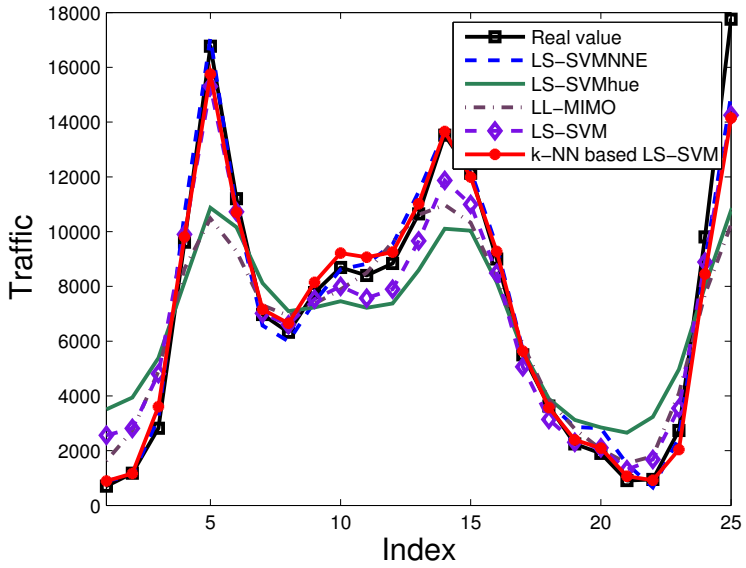

Figure 5.8: Prediction Results for NNGC1 Time Series

the k-NN component, it significantly improves the prediction precision.

Table 5.3 shows the prediction results and the tuned hyper-parameters on the chaotic laser time series at prediction horizon $n = 20$. There are 381 testing instances used in the experiments. In this case, LS-SVMNNE returns a very large prediction error comparing to other approaches, and the predicted curve is off the pattern for some testing instances as it is shown in Fig. 5.10. The huge prediction errors from these testing instances make the average RMSE a very large number. The selection of σ^2 is important. Using a fixed searching range for all the time series is not an ideal approach. Although sometimes it might turn out to be a good prediction for some time series, which is the case for the NNGC1 time series, it just happens coincidentally. It is necessary to adjust the searching range according to the characteristics of the given time series.

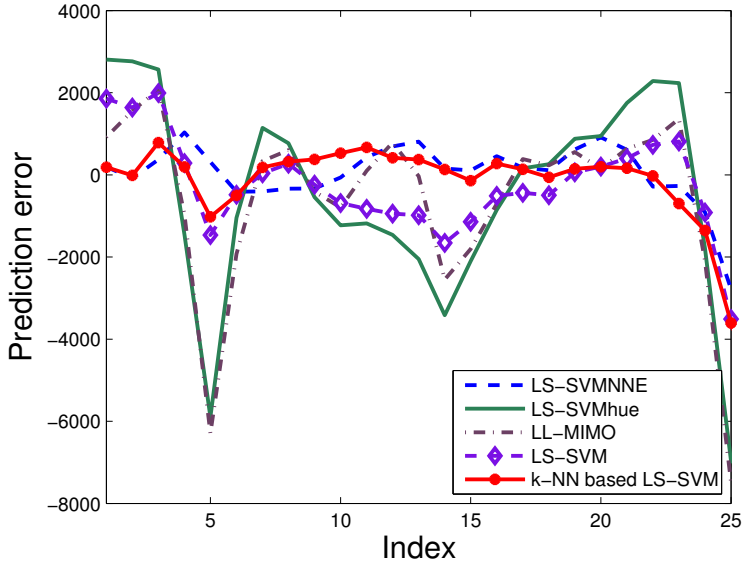

Figure 5.9: Prediction Errors for NNGC1 Time Series

Table 5.3: Prediction Results for Chaotic Laser Time Series

Methods	$RMSE$	γ	σ^2
LS-SVMNNE	64.3	16384	50
LS-SVMheu	18.2	47.3	360.6
LL-MIMO	21.8	N/A	N/A
LS-SVM	20.1	35148.9	1426.3
k-NN based LS-SVM	**16.9**	35148.9	1426.3

Both γ and σ^2 values obtained by LS-SVMheu and the k-NN based LS-SVM framework are quite different, but their $RMSE$s do not differ much. Again, it indicates the existence of the multiple local minima. Another part of the prediction result is shown in Fig. 5.11. The curve predicted by LS-SVMhue contains a constant delay in this case, while the proposed approach is able to follow the changing pattern well and almost overlaps the real time series. The prediction errors of the proposed approach variate around 0 slightly as shown in Fig. 5.12. It has also shown a clear improvement by the k-NN component comparing to the prediction result of LS-SVM

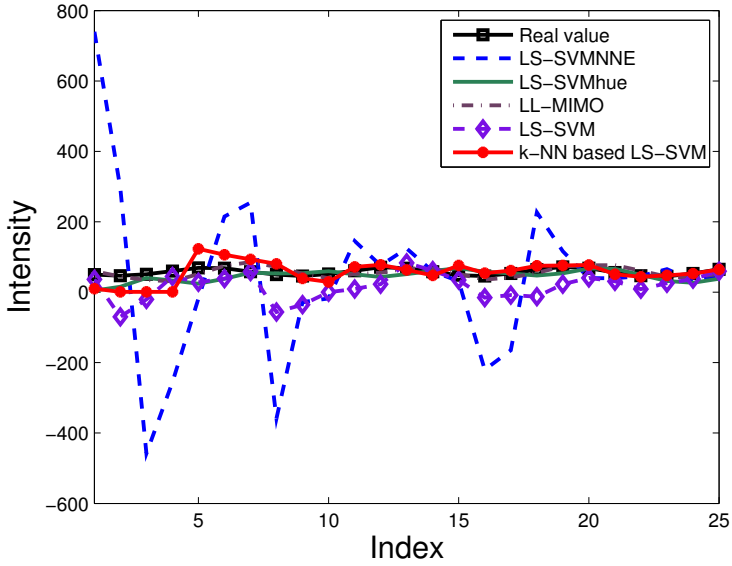

Figure 5.10: Prediction Results for Chaotic Laser Time Series

method alone.

The analysis results on the sunspot area data time series at prediction horizon $n = 20$ are shown in Table 5.4. There are 1020 testing instances used in the experiments. LL-MIMO does not preform as well for this time series with the highest $RMSE$, while it is a simple and fast method for long-term time series prediction. LL-MIMO approach selects k-NN of the testing instance from the training dataset, and calculates the average outputs of the selected instances as the prediction results. When the time series presents a more dynamic pattern, such as the case of the sunspot area data time series, LL-MIMO is not able to learn the changing pattern. On the other hand, the LS-SVM related approaches take the advantage of LS-SVM's non-linear modeling capability, and are able to conduct a better prediction. Figure 5.13 shows a part of the prediction results, and Figure 5.14 is the corresponding prediction error plot. The sunspot area data time series is the monthly averages of the daily sunspot areas.

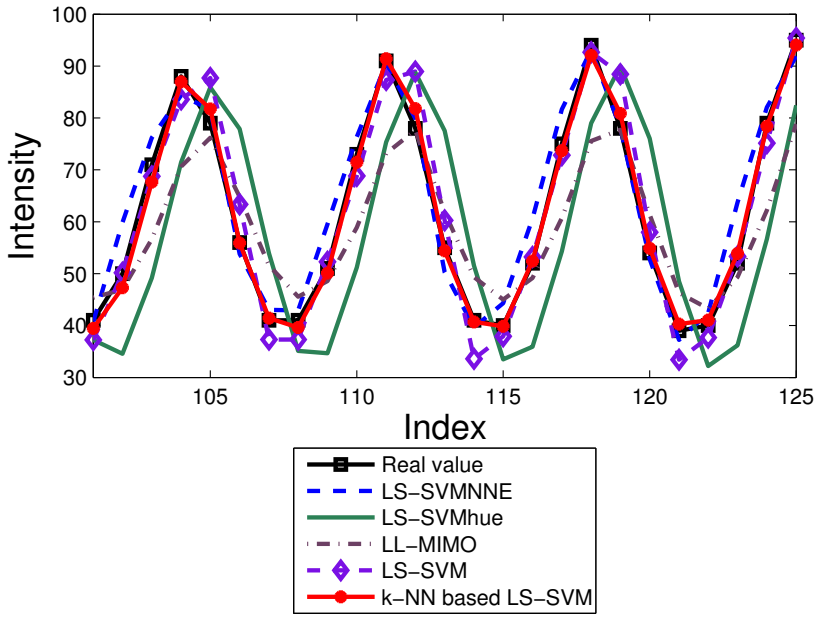

Figure 5.11: Prediction Results for Chaotic Laser Time Series

Its values change frequently at a fast pace. From the plot, we can observe that the original signal contains some high frequency components. It is a very challenging task to perform long-term time series prediction on this time series. The curve predicted by the k-NN based LS-SVM approach is able to capture and follow the main trend of the time series.

Table 5.4: Prediction Results for Sunspot Area Data Time Series

Methods	$RMSE$	γ	σ^2
LS-SVMNNE	754.3	8	50
LS-SVMheu	751.4	4.3	3956.4
LL-MIMO	773.2	N/A	N/A
LS-SVM	736.6	10.0	3862.1
k-NN based LS-SVM	**736.2**	10.0	3862.1

The time consumption is shown in Table 5.5, which shows the average time it costs to do n-step ahead prediction once. LS-SVMNNE, LS-SVMheu, and LS-SVM take a

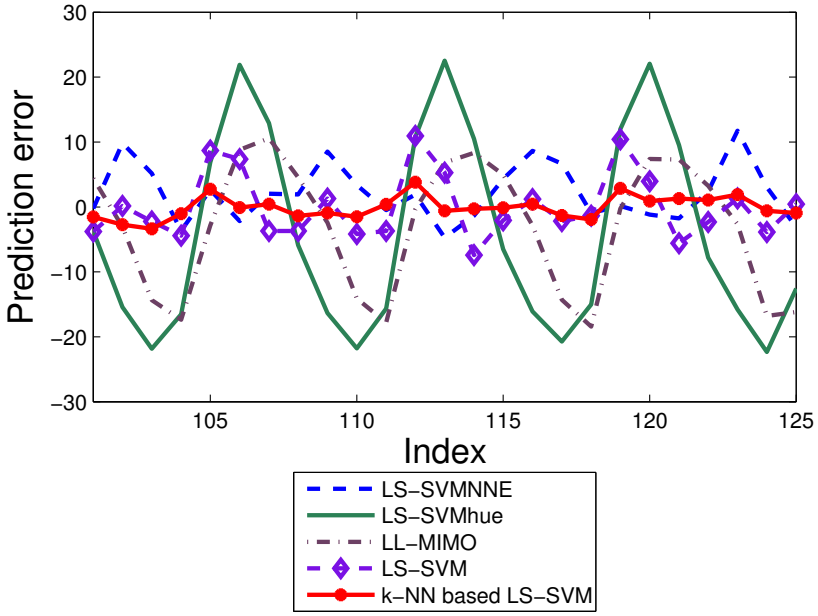

Figure 5.12: Prediction Errors for Chaotic Laser Time Series

longer time because of the high complexity of the LS-SVM model. Although including the k-NN component does require some additional time before training the LS-SVM regressors, the time cost is not comparable with the time needed for training the LS-SVM regressors with the whole training dataset. The k-NN component reduces the number of training instances significantly, and therefore dramatically reduces the complexity and the time cost for training one LS-SVM regressor (even more saving for n LS-SVM regressors). The proposed k-NN based LS-SVM approach is able to achieve a smaller overall time cost than the compared LS-SVM based approaches. If the dataset is large and n is large, the proposed framework will benefit even more from the k-NN component. In order to achieve a better prediction result, the prediction model is re-trained for every testing instance, so it takes a longer time than LL-MIMO which does not have the training process. Overall, the proposed k-NN based

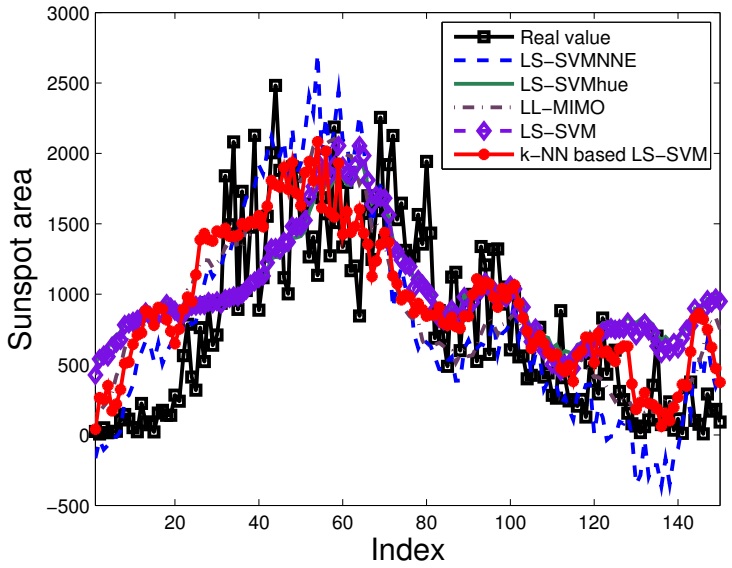

Figure 5.13: Prediction Results for Sunspot Area Data Time Series

LS-SVM framework achieved the lowest prediction error, and it is much faster than LS-SVMNNE, LS-SVMheu, and LS-SVM.

Table 5.5: Time Consumption

Methods	Time Cost (s)
LS-SVMNNE	4.42
LS-SVMheu	4.85
LL-MIMO	0.01
LS-SVM	4.85
k-NN based LS-SVM	1.14

We conducted more experiments to demonstrate that the proposed strategy for selecting γ is effective. Figure 5.15 to Fig. 5.17 show the prediction errors for a fixed σ^2 value with different γ values for all three time series. The σ^2 values are set to the tuned values for the three time series respectively, *i.e.*, 22772.0 for NNGC1 time series, 1426.3 for chaotic laser time series, and 3862.1 for sunspot area data time series. The red dashed lines in the figures are pointing out the suggested values for

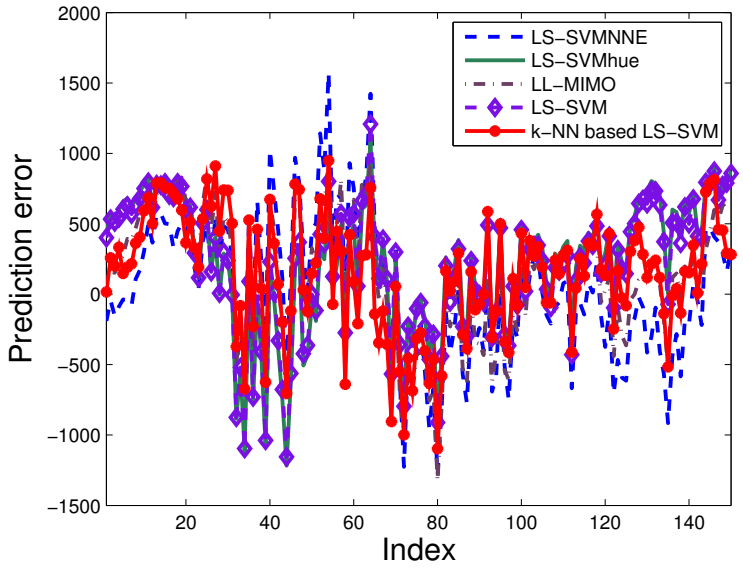

Figure 5.14: Prediction Errors for Sunspot Area Data Time Series

γ by the proposed framework. As it can be observed from the figures, the suggested values are very close to the optimal values. Furthermore, the proposed strategy is very efficient. The average number of the iterations for locating γ is 3 for NNGC1 time series, 2 for chaotic laser time series, and 1.86 for sunspot area data time series. The overall average number of the iterations is 2.29. It is much more efficient than the exhaustive grid search. Appropriately utilizing the information extracted from the training time series enables us to tune the parameters efficiently and effectively.

It is true that both the parameter tuning strategy and the k-NN component can be applied to other types of prediction tasks, for example, one-step ahead time series prediction. However, we believe that the effects of the proposed parameter tuning strategy and the k-NN component to the prediction performance will be more observable and significant in the application of long-term time series prediction. The reason is that for long-term time series prediction, the prediction needs to be done multiple

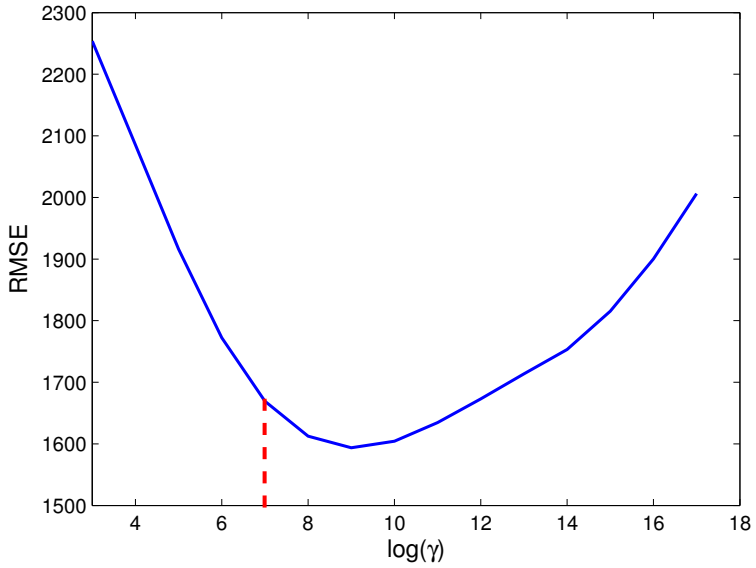

Figure 5.15: Prediction Error for a Fixed σ^2 with Different γ for NNGC1 Time Series

steps ahead, but the uncertainties of the future trend and the lack of information make the prediction task very difficult. These challenges in turn imply more opportunities for the improvement. Therefore, the significance and merits of the proposed parameter tuning strategy and the k-NN component can be better demonstrated.

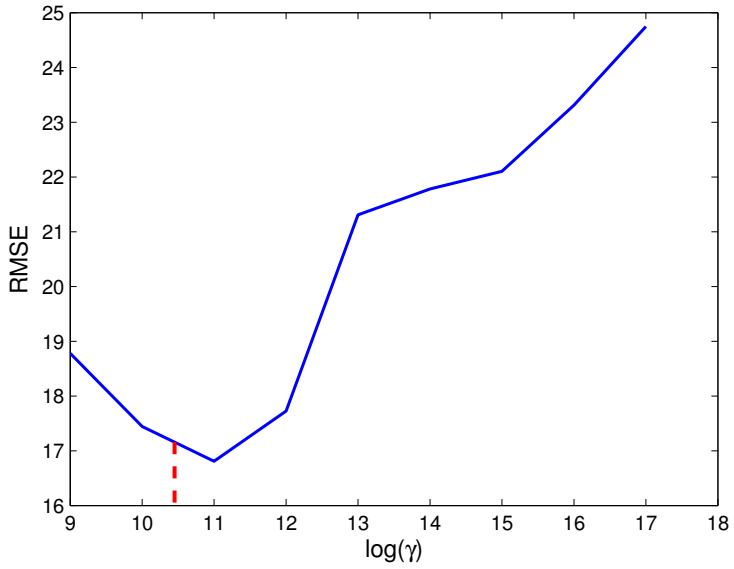

Figure 5.16: Prediction Error for a Fixed σ^2 with Different γ for Chaotic Laser Time Series

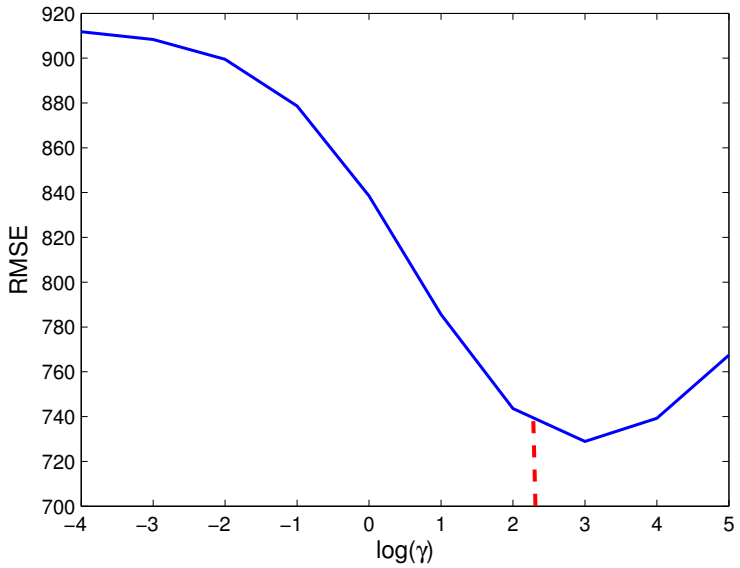

Figure 5.17: Prediction Error for a Fixed σ^2 with Different γ for Sunspot Area Data Time Series

CHAPTER 6

Framework Evaluation

In this chapter, to demonstrate the merits of the proposed framework, we present the experimental results on multiple de-identified labor tracings. On one hand, in order to show the prediction ability of the proposed prediction model, we compare the output from the k-NN based LS-SVM long-term time series prediction with four existing methods, namely: LL-MIMO (Bontempi 2008), LS-SVM (Pelckmans, Suykens, Gestel, Brabanter, Lukas, Moor, and Vandewalle 2003; Suykens, Gestel, Brabanter, Moor, and Vandewalle 2002), AR (Soltani, Boichu, Simard, and Canu 2000), and autoregressive moving average (ARMA) (Zhang 2003). The compared values are the outputs from the prediction models without any adjustment by the post-prediction components. The metrics used to evaluate the performance of the algorithms are $RMSE$, $SMAPE$, and FIT (defined in Section 3.6). On the other hand, the purpose of the proposed framework is to predict the starting points of the coming contractions, so as to give the signal for analgesia injection ahead of time to relieve the labor pain. Therefore, a criterion is proposed in Section 3.6 to evaluate the prediction accuracy of the proposed framework.

6.1 Dataset

We received de-identified pressure data files of 8460 women who gave birth between April 2010 and September 2011. Each patient's tracing is stored in a single file. In each file, there is a matrix with interleaved rows representing intrauterine pressure data and baby's heart rate. Accordingly, the first cell of each row contains the day and time information. The second cell specifies if the row contains 'UA' (uterine activity) or 'HR2' (heart rate) data. The third cell specifies the type/method/status of the measurement, such as 'IUP' for intrauterine pressure and 'TOCO' for external uterine pressure. The remainder of the row, a total of 240 cells, contains the numerical data corresponding to one-minute record. Because the data are sampled once each 1/4 second, there are $4 \times 60 = 240$ data points in one minute.

The uterine pressure signal collected externally is usually contaminated by lots of noise. It is hard to recognize contractions from a noisy signal. In contrast, the 'IUP' signal is collected internally, which has a much clearer contraction pattern as compared to the 'TOCO' signal. Therefore, we only consider 'IUP' signals at this stage. The intrauterine pressure measure is an invasive measurement, and it can be applied only after rupture of the membranes. Most of the files we received do not contain 'IUP' signals, because those data were collected externally using a tocodynamometer. Out of the 8460 files, 2422 files contain 'IUP' signals, but some of them are shorter than 30 minutes, which are too short for the useful analysis. We select 611 women's 'IUP' tracings, together with their demographic and obstetrical information, to form the historical patient information HI. We randomly select 11 patients' records for testing. The corresponding demographic and obstetrical information for each of the

Table 6.1: Patient Demographic and Obstetrical Information for Testing

Patient ID	Indication for Oxytocin	Gestational Age (week)	Labor Anesthesia	Maternal Age (year)
1	Augmentation	40	Epidural	22
2	Augmentation	39	Epidural	18
3	N/A	41	Epidural	34
4	Augmentation	39	Epidural	23
5	Induction	38	Epidural	18
6	Induction	38	Epidural	18
7	Augmentation	38	Epidural	29
8	Augmentation	38	Epidural	34
9	Induction	38	Epidural	43
10	Augmentation	41	Epidural	27
11	Augmentation	35	Epidural	35

selected patients for testing is shown in Table 6.1. The selected tracings, either for training or for testing, contain a sequence of recognizable peaks.

6.2 Experiment and Results

There are some parameters and thresholds that need to be determined for the compared approaches. An intrauterine time series is subsampled once every second during the data-preprocessing step, so the time interval of the preprocessed time series is one second. The onset of the remifentanil's effect is approximately 30 seconds, so the prediction should be made approximately 30 seconds (adjustable according to the requirement) ahead of the next contraction to accurately match the effect of analgesia. Accordingly, we set the prediction horizon to 30. We present the results using one set of pre-defined values for the length of the training time series ($Tlength$), the length of input vector (p), and the number of neighbor points (k), which are shared by the evaluated approaches. The training time series is the intrauterine pressure time series

of the most recent 6 contractions, so the exact length of the training time series varies from patient to patient, which depends on how long the most recent 6 contractions last. The length of the input vector p is set to 5. Parameter k in the k-NN approach is set to 80. An additional tuning process on these parameters will also help to increase the prediction performance, particularly there are plenty of efforts on the input selection, while it is beyond the scope of this study. We select the values of these parameters based on our empirical study. In addition, we set $minSupL$, $minSupG$, and $minPS$ to 0.03, 0.1, and 0.01, respectively for the sequential association rule mining process. These thresholds can be adjusted if we get more patient tracings or a different set of patient tracings in the selected historical patient tracings HT. Parameters γ and σ are obtained from the parameter tuning process as described in Chapter 5.

The comparison results are shown in Table 6.2, Table 6.3, and Table 6.4 in terms of $RMSE$, $SMAPE$, and FIT, respectively. As we can see from the results, the proposed approach achieves the lowest prediction error in terms of both $RMSE$ and $SMAPE$. It also achieves the highest FIT measure. The experimental results show that the proposed framework is superior to the compared four methods, on average 64.2%, 15.9%, 106.2%, and 51.3% better in terms of $RMSE$, and 100.5%, 36.2%, 163.5%, and 61.1% better in terms of $SMAPE$, than LL-MIMO, LS-SVM, AR, and ARMA, respectively. The intrauterine pressure time series are rather dynamic and complex, thus we do not expect the prediction error to be zero. When the prediction significantly deviates from the true value, it is possible that the FIT measure is a negative number according to its definition in Equation (3.22). As shown in Table 6.4, the FIT measures of the prediction results of AR and ARMA are negative numbers

Table 6.2: Experimental Results in Terms of Root Mean Squared Error

Patient ID	LL-MIMO	LS-SVM	AR	ARMA	Proposed Framework
1	16.61	10.28	16.60	13.36	**8.11**
2	16.98	11.12	21.81	10.72	**10.27**
3	15.99	12.16	20.36	18.03	**9.83**
4	10.87	6.78	14.12	8.41	**5.79**
5	15.66	11.43	19.03	18.49	**9.98**
6	15.67	11.04	19.07	15.76	**8.45**
7	21.98	15.76	26.57	22.05	**13.74**
8	20.24	16.07	28.28	15.64	**15.17**
9	15.99	10.46	22.78	11.37	**9.21**
10	25.82	19.28	33.19	20.98	**16.08**
11	12.67	8.69	14.93	14.37	**8.17**

for some patients.

As analyzed in Chapter 2, not every model that works for short-term time series prediction would work well for long-term prediction. The long-term time series prediction is a much more challenging task due to the ever-changing labor contraction pattern. LL-MIMO is a simple and low computational cost algorithm. It predicts the future values by calculating the average of the training instances. LL-MIMO does not train a model to describe the inherent relationship between inputs and outputs, and its prediction ability is very limited as analyzed above.

The LS-SVM method gives fair prediction results, but it is computationally expensive due to using a large training dataset and complex computing. On average, it takes 20.4 seconds to perform a single 30 seconds-ahead prediction using LS-SVM, while it only takes 1.0 second for the proposed framework, which is about 19 times faster than LS-SVM. Therefore, LS-SVM is not applicable for real-time prediction, even though it gives better prediction than LL-MIMO.

Table 6.3: Experimental Results in Terms of Symmetric Mean Absolute Percentage Error

Patient ID	LL-MIMO	LS-SVM	AR	ARMA	Proposed Framework
1	0.302	0.174	0.254	0.174	**0.120**
2	0.468	0.298	0.646	0.139	**0.234**
3	0.406	0.295	0.535	0.484	**0.222**
4	0.272	0.157	0.314	0.161	**0.123**
5	0.425	0.275	0.413	0.416	**0.225**
6	0.404	0.250	0.436	0.341	**0.166**
7	0.537	0.520	0.916	0.589	**0.236**
8	0.441	0.306	0.668	0.319	**0.270**
9	0.411	0.238	0.603	0.262	**0.209**
10	0.549	0.370	0.822	0.447	**0.268**
11	0.325	0.202	0.360	0.315	**0.192**

The autoregressive model is a linear prediction method that attempts to predict the next value based on the previous observations. Because of its linear nature, it is not able to achieve good prediction precision if the time series contains non-linear components. In the case of long-term time series prediction, the mapping function is usually non-linear. Therefore, the autoregressive model is not preferable. The prediction errors of AR are usually the highest among the compared algorithms, and the FIT measures are negative for most of patients.

Comparing to AR, ARMA includes a moving average part, which incorporates the prediction error to build the prediction model. This makes ARMA more capable and faster in following a time series trend. It can be also observed from Table 6.2, Table 6.3 and Table 6.4 that ARMA always outperforms AR.

The proposed post-prediction process aims to further improve the prediction precision. It smoothes out the predicted curves and facilitates detecting starting points. We include some plots of the predicted intrauterine pressure time series together with the original tracings in Fig. 6.1 - Fig. 6.11 to demonstrate the prediction performance

Table 6.4: Experimental Results in Terms of the FIT Measure

Patient ID	LL-MIMO	LS-SVM	AR	ARMA	Proposed Framework
1	8.08	43.15	8.15	26.06	**58.15**
2	2.01	35.85	-25.86	38.12	**42.72**
3	4.77	27.60	-21.23	-7.38	**46.07**
4	6.47	41.66	-21.46	27.65	**47.57**
5	1.84	28.38	-19.26	-15.88	**40.16**
6	5.88	33.69	-14.52	5.37	**53.78**
7	8.23	34.19	-10.89	7.94	**47.28**
8	6.63	25.84	-30.50	27.83	**40.16**
9	9.68	40.92	-28.72	35.77	**46.57**
10	1.84	26.71	-26.14	20.26	**41.39**
11	6.70	36.00	-9.95	-5.78	**46.46**

of the complete proposed framework. The accuracy (defined in Section 3.6) for each tracing is included in the title of the plot. It can be observed that the prediction is able to anticipate and preserve the trend of the intrauterine pressure. High accuracies demonstrate that the proposed framework is capable of predicting the start of the upcoming contractions consistently.

In statistical test theory, two types of statistical errors are distinguished, type I error and type II error. A type I error, also known as an error of the first kind, is the wrong decision that is made when a test rejects a true null hypothesis (H_0). A type I error is also referred to as false positive in some test situations. In contraction prediction, type I error is the wrong prediction of contraction that is made when there is no upcoming contraction.

A type II error, also known as an error of the second kind, is a wrong decision that is made when a test fails to reject a false null hypothesis. A type II error is also referred to as false negative in some test situations. In contraction prediction, type II error is when there is a forthcoming contraction, but the prediction model fails to

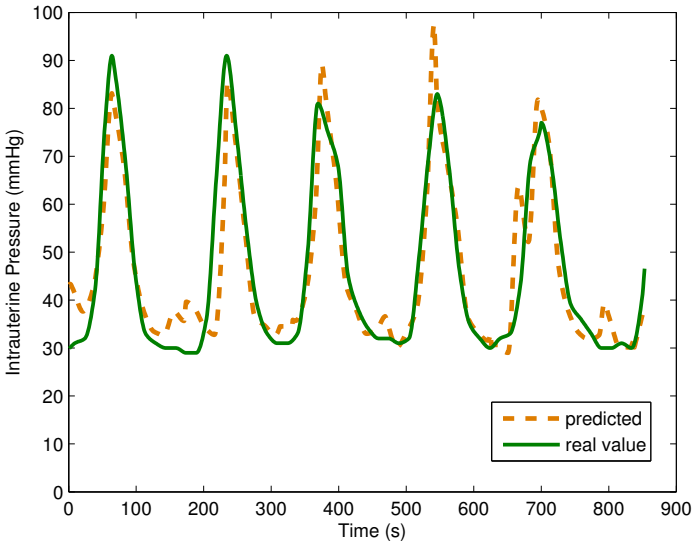

Figure 6.1: Prediction Results for Patient 1 (Accuracy = 0.92)

anticipate it.

Both types of errors are unwanted. Based on the real-life consequences of an error, one type may be more serious than the other. A type I error in contraction prediction gives the parturient more amount of remifentanil than it is needed, while a type II error causes no or insufficient remifentanil infusion. Because of the merits of remifentanil (*i.e.*, fast on, fast off, and almost none of it gets to baby, etc.), there is no serious side effect if the parturient gets a few more boluses of remifentanil. On the other hand, if there is no or insufficient remifentanil available when a contraction occurs, the parturient will experience severe labor pain, which significantly lowers the parturient's satisfaction. Therefore, the type II error in contraction prediction is more serious regarding the consequences.

As the results show in Fig. 6.1 - Fig. 6.11, the proposed prediction framework did not fail to predict any contractions (*i.e.*, there is no type II error), although it

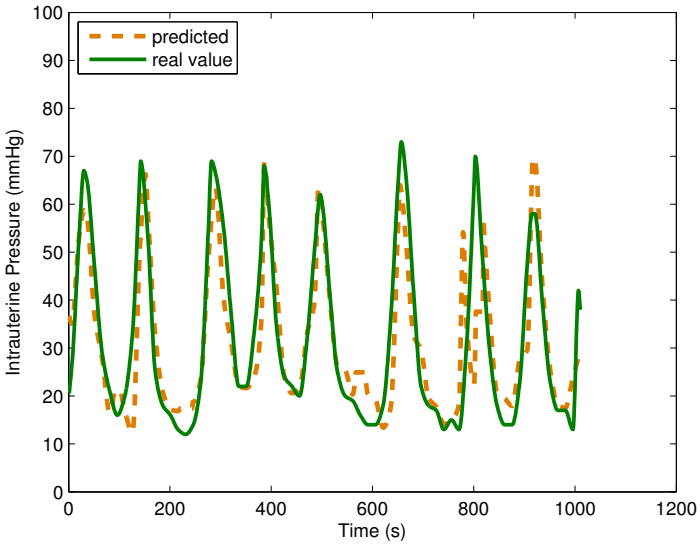

Figure 6.2: Prediction Results for Patient 2 (Accuracy = 0.91)

predicts a few extra contractions (*i.e.*, type I errors, which are tolerable) as can be seen in Fig. 6.7 and Fig. 6.10. In real applications, patient-controlled analgesia (PCA) with remifentanil can be incorporated in order to decrease the risk of type II errors. If the parturient needs more analgesia than the recommended dosing regimen based on the contraction prediction, she can start or increase a remifentanil bolus at the first subjective sign of an upcoming uterine contraction. Meanwhile, the PCA device should be well-programmed to make sure that the parturient will not overdose.

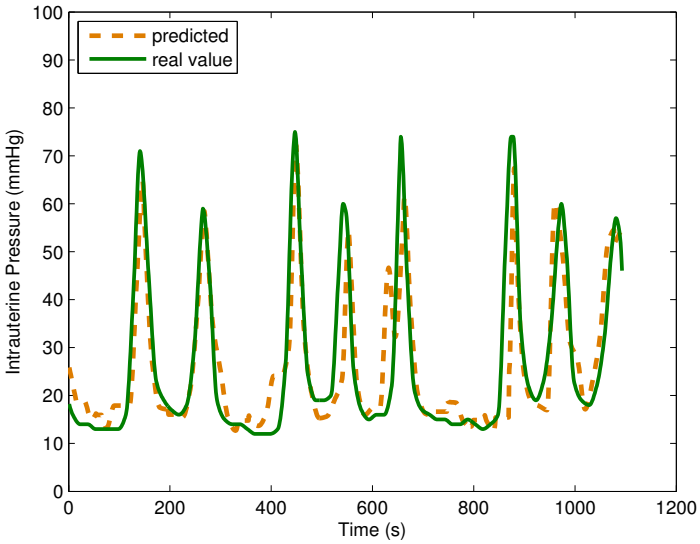

Figure 6.3: Prediction Results for Patient 3 (Accuracy = 0.83)

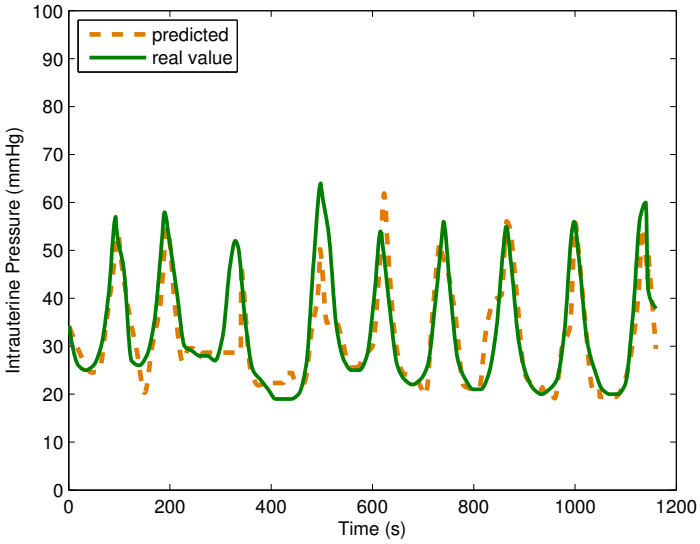

Figure 6.4: Prediction Results for Patient 4 (Accuracy = 0.83)

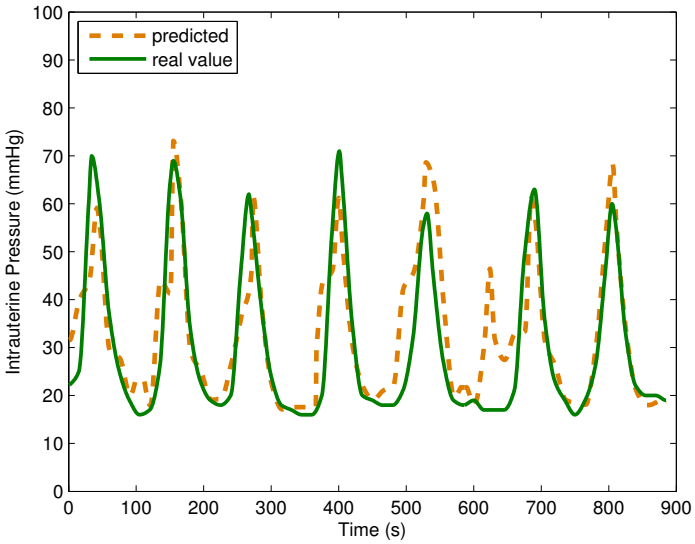

Figure 6.5: Prediction Results for Patient 5 (Accuracy = 0.86)

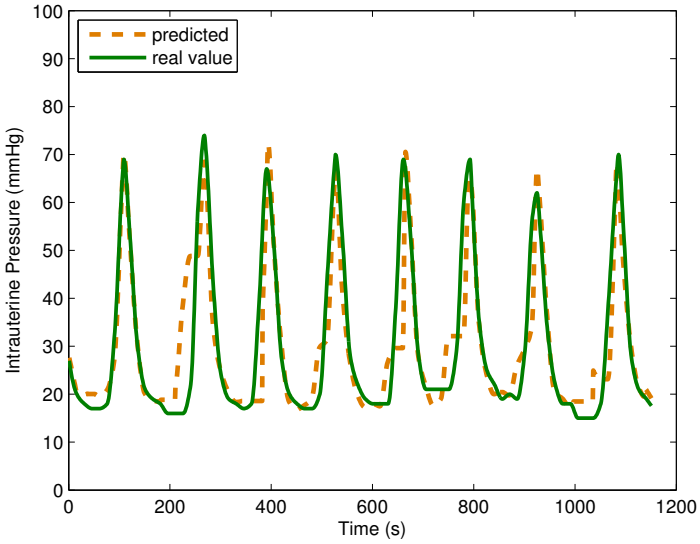

Figure 6.6: Prediction Results for Patient 6 (Accuracy = 0.78)

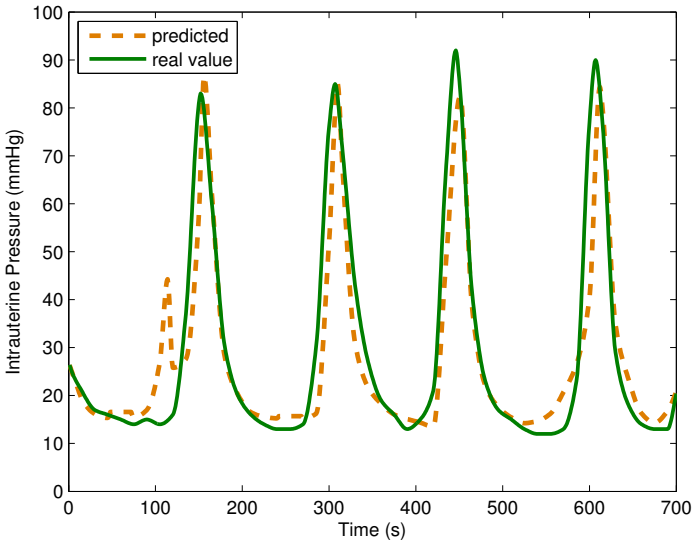

Figure 6.7: Prediction Results for Patient 7 (Accuracy = 0.80)

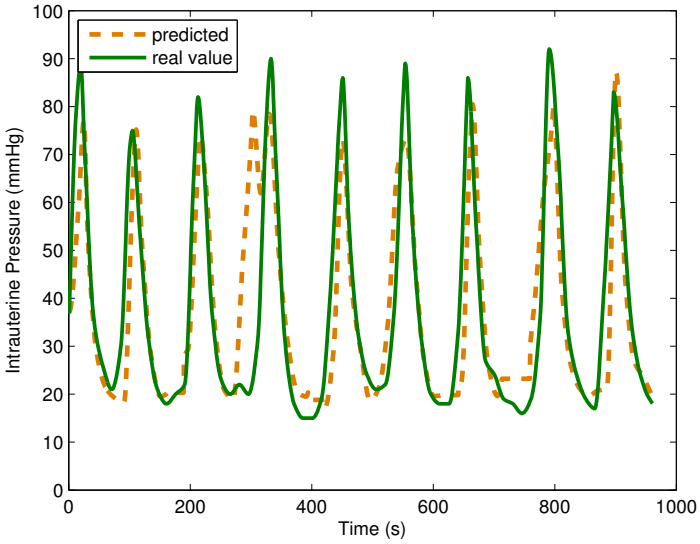

Figure 6.8: Prediction Results for Patient 8 (Accuracy = 0.86)

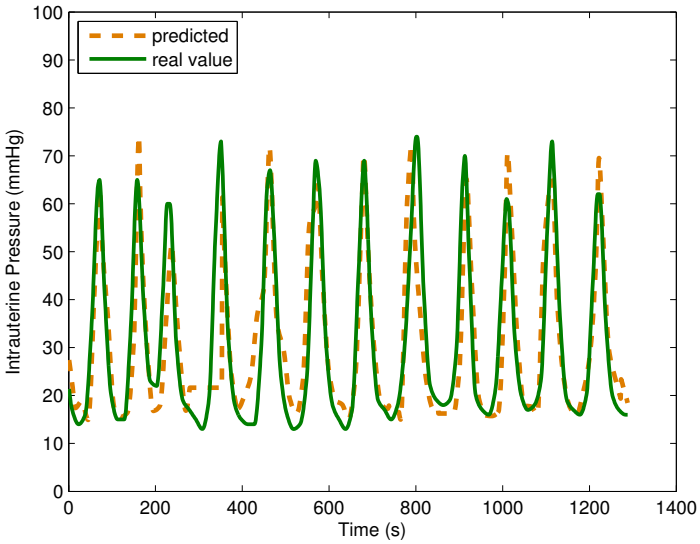

Figure 6.9: Prediction Results for Patient 9 (Accuracy = 0.95)

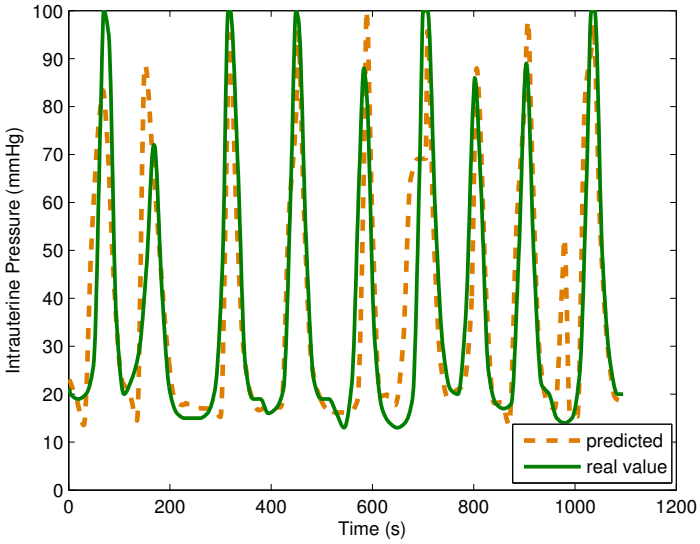

Figure 6.10: Prediction Results for Patient 10 (Accuracy = 0.76)

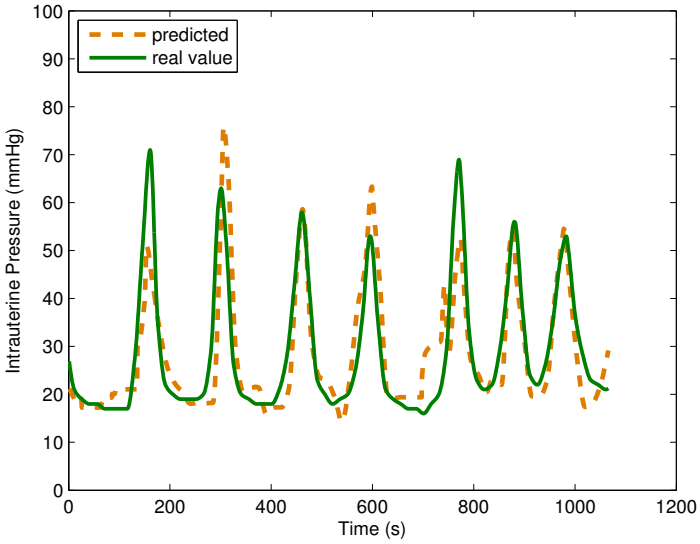

Figure 6.11: Prediction Results for Patient 11 (Accuracy = 0.7)

CHAPTER 7

Conclusion and Future Work

In Chapter 1 and Chapter 2, the background of this work and the related state-of-art research are discussed. In the next four chapters, the proposed system is presented in detail, and the performance analysis is provided. In this concluding chapter, a summary of the contributions of the proposed system is given, followed by the discussion of the future research directions.

7.1 Conclusion

Designing an optimal dosing regimen for remifentanil necessitates the prediction of the pace of contractions, so that the drug can be given shortly before the pain caused by a contraction begins. Meanwhile, the prediction should be made early enough to allow for an administration of remifentanil that will have maximal efficacy during contractions, little effect between contractions, and minimal impact on the fetus. In this dissertation, we introduce the novel knowledge-assisted sequential pattern analysis framework to tackle the difficult task of predicting the timing of the upcoming labor contractions. The framework reveals interesting sequential patterns in the in-

trauterine pressure sequences, which we employ to assist contraction prediction using the sequential association rule-based collaborative filtering strategy.

- The five-step framework is introduced in this dissertation. The steps are 1) patient selection, 2) collaborative training dataset selection, 3) heuristic parameter tuning for LS-SVM, 4) k-NN based LS-SVM for long-term time series prediction, and 5) the post-prediction process, which includes boundary constraint, multi-value integration and vertical correction components.

- The contraction pattern is defined by the combination of the height of the peak that represents the contraction and the period of the contraction. A new sequential association rule mining approach is designed to discover the sequential uterine contraction pattern. Sequential pattern analysis on intrauterine pressure tracings provides a way to characterize the pattern of contractions. This enables analysis on whether and how the demographic and obstetrical features impact the sequential uterine contraction pattern. The analysis allows us to group the patients based on some selected demographic and obstetrical features that impact the sequential uterine contraction pattern.

- A new sequential association rule-based dataset selection method with a collaborative filtering strategy is then proposed to dynamically select the training dataset from the patient intrauterine pressure database and the current patient's own past tracing. The collaborative filtering component uses the known sequential contraction patterns of a group of past patients to assist in the prediction of unknown upcoming contractions for the current patient of interest. We selectively employ available contractions from both the selected historical

patient tracings and the current patient's most recent intrauterine pressure tracing to train the prediction models. This dynamic selection makes the prediction models adaptive to the changing contraction pattern of the patient.

- A novel heuristic method is proposed to decide the search interval of the Gaussian kernel parameter based on the information extracted from the training time series. A strategy for efficiently locating the regularization factor is also proposed. The parameter tuning process guarantees that the proposed k-NN based LS-SVM framework is able to reach its best performance. In the k-NN based LS-SVM framework, the k-NN component reduces the complexity of training the LS-SVM, and also improves the prediction accuracy. We design a new distance function, which incorporates the Euclidean distance and the dissimilarity of the trend of a time series for the k-NN method. The promising experimental results validate the effectiveness and efficiency of the tuning strategy and the modeling for long-term time series prediction.

- The post-prediction process enhances the correctness and smoothness of the predicted results. The boundary constraint component sets a dynamic constraint for the predicted values to make sure that the results are valid in the application domain. The multi-value integration component combines prediction results from several individual models to generate an output, which decreases the uncertainty of the prediction model. The vertical correction component detects and smoothes out the irregular sharp pulses and peaks with very low heights in the prediction results, and generates the final output of the proposed framework.

This dissertation provides a solution for predicting upcoming uterine contractions for women in labor. Prediction accuracy is essential in order to maximize the effect and minimize the side effect profile of remifentanil. The experimental results have shown that the proposed framework outperforms some existing methods in many aspects. Designing the optimum dosing regimen also requires studies on predicting the intensity of upcoming contractions and on the pharmacodynamics of remifentanil, so that we will be able to indicate when and how much remifentanil to inject to relieve a patient's labor pain. Ultimately, success depends on accurate predictions in order to administer an appropriate amount of remifentanil at the appropriate time, thus enhancing current pain management techniques and improving safety for mothers and babies.

7.2 Future Research Direction

We have seen some encouraging results of the proposed knowledge-assisted sequential pattern prediction framework. The following topics are some interesting extensions of the proposed framework.

1. Design a system for remifentanil infusion with optimum dosing regimen.

   In order to design a system for remifentanil infusion with optimum dosing regimen, there are four major components we need to accomplish as shown in Fig. 7.1. Each of the components represents a particular task. The tasks are 1) predicting the timing of the upcoming contraction, 2) predicting the intensity of the upcoming contraction, 3) analyzing the pharmacodynamics of remifentanil,

and 4) designing a device for remifentanil infusion, which takes the information provided by the first three components and interacts with patients and anesthetists.

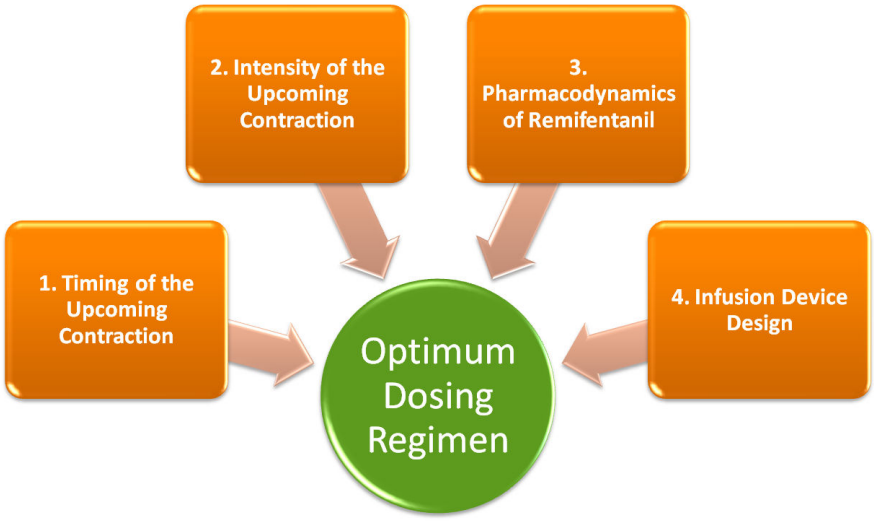

Figure 7.1: Systematic View

The focus of this dissertation is on the first task. We give the prediction of the timing of the upcoming contraction multiple seconds ahead. It also ensures that there is enough time to inject remifentanil and let remifentanil reach the peak of its effect to relieve the pain caused by the upcoming contraction. The prediction of the upcoming contraction helps to determine when remifentanil should be injected. The timing of the injection is also related to how fast remifentanil reaches the peak of the effect after injection, which relates to the pharmacodynamics of remifentanil.

The intensity of a contraction can be described by the height of the peak that represents the contraction. The prediction of the intensity of the upcoming

contraction should be also made at the time when we predict the starting point of a contraction. Usually, after a contraction initiates, it takes approximately 30 seconds to reach the peak of the contraction. Thus, if we predict the starting point and the intensity of a contraction at the same time, it means that the prediction horizon for predicting the intensity is 30 seconds higher than it is for predicting the starting point. It is a more challenging long-term prediction task. One solution is to use the framework presented in this dissertation, and set a higher prediction horizon, *i.e.*, 60 (=30+30) seconds, and return the height of the peak as the indication of the intensity. However, predicting the intrauterine pressure time series 60 seconds in advance is very difficult, and the predicted height might not be very accurate. Another solution could be that for a sequence of contractions, we extract the height feature for each contraction, analyze the sequence of the height feature, and try to conduct prediction on the height of the upcoming contraction directly.

Pharmacodynamics is the study of the biochemical and physiological effects of drugs on the body or on microorganisms or parasites within, the mechanisms of drug action, and the relationship between drug concentration and effect. Research should be conducted to find out when and how much remifentanil should be administered to patients in labor in order to relieve the pain caused by one contraction. Patients' demographic information probably also plays a role in determining these parameters. For example, a patient with larger BMI would need a larger dose to reach the same drug concentration and effect.

Timing and intensity of the upcoming contraction together with pharmacodynamics of remifentanil as a whole determine the timing and dosage of the remifentanil infusion. An infusion device, which takes the information provided by the first three components as the input, should be designed to inject remifentanil to patients according to the given command of timing and dosage. In addition, the infusion device should be able to interact with patients and anesthetists. For example, if the patient still feels much pain when she is receiving the timely infusion of remifentanil, she should be able to manually increase the rate of the infusion. Thus, the infusion device also takes the patient's or anesthetist's feedback as additional input to adjust the infusion rate.

2. Incorporate more detailed patient demographic and obstetrical information to the framework.

   This dissertation suggests a way to utilize the patient demographic and obstetrical information to assist the prediction task. More statistic studies on the relationship between demographic information and the contraction pattern could be conducted.

   In the proposed framework, we use the indication of oxytocin as one of the features to select the patients for the training purpose. We first determine which group the current patient should belong to according to her gestational age, labor anesthesia, and indication of oxytocin using the tree shown in Fig. 7.2. Let the current patient belong to group i, where $1 \leq i \leq 7$. Second, select $nPatient$ patients from group i whose age are closest to the current patient's age. The selected patients' intrauterine tracings are the output of the patient

selection component, and are passed to collaborative training dataset selection component as one of its inputs.

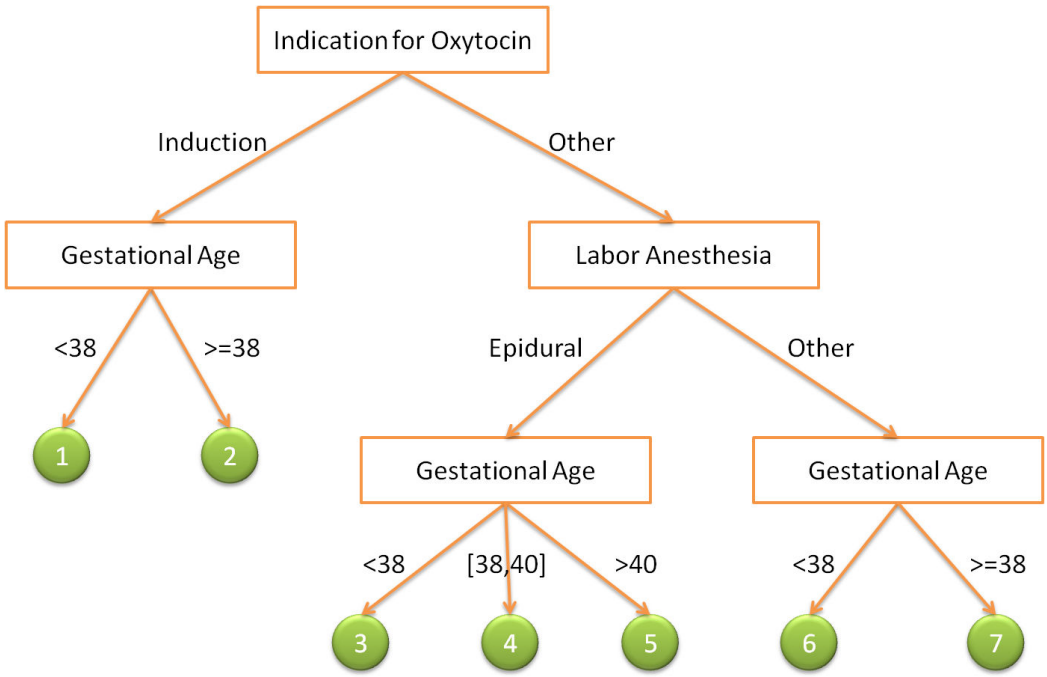

Figure 7.2: A Tree for Patient Selection

In the current framework, every segment of one intrauterine tracing is considered as equal, and become potential training dataset once the patient is selected. For those patients whose indication of oxytocin is 'Induction', we could employ the indication for oxytocin as a time series input to divide the intrauterine tracings. The speed of oxytocin flow injected to the patients is changing throughout the labor precess. We could further divide the intrauterine pressure time series into segments according to the rate of oxytocin infusion.

Table 7.1 gives an example of the indication of oxytocin record for one patient. According to the record, pitocin (oxytocin injection) started at 09:25:00

Table 7.1: The Indication of Oxytocin Record for One Patient

03/01/11 09:25:00	Pitocin Started milliunits/min @ 1
03/01/11 10:10:00	Pitocin Increased milliunits/min @ 2
03/01/11 11:10:00	Pitocin Increased milliunits/min @ 4
03/01/11 11:40:00	Pitocin Increased milliunits/min @ 6
03/01/11 12:10:00	Pitocin Increased milliunits/min @ 8
03/01/11 12:40:00	Pitocin Increased milliunits/min @ 10
03/01/11 13:10:00	Pitocin Increased milliunits/min @ 12
03/01/11 13:40:00	Pitocin Increased milliunits/min @ 14
03/01/11 14:30:00	Pitocin Increased milliunits/min @ 16
03/01/11 15:00:00	Pitocin Increased milliunits/min @ 18
03/01/11 15:32:00	Pitocin Increased milliunits/min @ 20 MU/MIN
03/01/11 15:44:00	Pit_D/C
03/01/11 15:44:00	Pit_D/C

at 1 MU/MIN (milliunits per minute). The rate increased to 2 MU/MIN at 10:10:00. The oxytocin injection was disconnected at 15:44:00. Oxytocin is the major hormone that stimulates the contraction of smooth muscle of the uterus during labor. It speeds up labor contractions and delivery. We propose that the training dataset selection could be better if we also classify the segments of intrauterine tracing according to the amount of oxytocin that the patients were having. In the given example in Table 7.1, we can retrieve 13 segments which were collected before 09:25:00, [09:25:00 - 10:10:00], [10:10:00 - 11:10:00], [11:10:00 - 11:40:00], [11:40:00 - 12:10:00], [12:10:00 - 12:40:00], [12:40:00 - 13:10:00], [13:10:00 - 13:40:00], [13:40:00 - 14:30:00], [14:30:00 - 15:00:00], [15:00:00 - 15:32:00], [15:32:00 - 15:44:00] and after 15:44:00, respectively. Each segment is marked by the corresponding oxytocin injection rate.

If the current patient of interest is having oxytocin injection 2 MU/MIN, than the segment collected in [10:10:00 - 11:10:00] should be considered for the train-

ing instance selection. Currently, the record of the oxytocin infusion is not complete and sometimes contains error. We can have this information fully documented correctly and use it for finer training dataset selection.

3. Enhance the discretization method.

   The discretization method introduced in Section 4.1, which is used for converting numerical features to nominal features, directly influences the sequential pattern discovered by the sequential association rule mining algorithm. How to choose the intervals for discretization is crucial. The discretized intervals should not hide the patterns, thus the intervals should not be either too big or too small. Meanwhile, the intervals should be semantically meaningful.

   Discretization is more often discussed in the supervised classification domain. In our task, it requires an unsupervised discretization method, because there is no class concept and accordingly, and there is no prior knowledge about the class information. Existing unsupervised discretization methods include equal width partitioning, equal frequency partitioning, and k-means based discretization. For equal width partitioning, the size of each interval is the same. For equal frequency partitioning, the number of the instances in each partition is the same. k-means based discretization divides the partitions using the k-means clustering method, which groups the instances that are close to each other in one partition. The number of instances in each group varies. All these existing unsupervised discretization methods do not take the data distribution into consideration. We extract two features from intrauterine pressure tracings, *i.e.*, height and period. The 2-D distribution of height and period is shown in Fig. 7.3 .

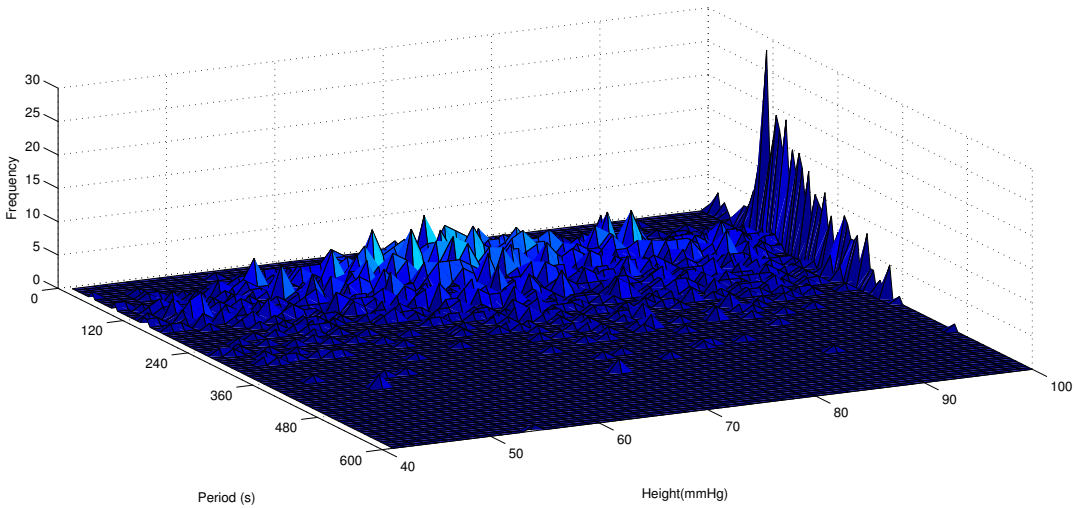

Figure 7.3: Distribution

It is observed that the distribution of the instances is not uniform. A discretization method that considers the 2-D density information is probably more suitable for sequential association rule mining than equal width partitioning. First, normalized both height and period according to Equation (7.1) and Equation (7.2).

$$h = \frac{height - minHeight}{maxHeight - minHeight}, \tag{7.1}$$

$$p = \frac{period - minPeriod}{maxPeriod - minPeriod}, \tag{7.2}$$

where $maxHeight$ and $minHeight$ are the maximum and minimum values of the height; $maxPeriod$ and $minPeriod$ are the maximum and minimum values of the period. Given a point (h, p), if we consider the area with radius r centered on (h, p), a kernel density function is proposed in Equation (7.3).

$$dens = \frac{1}{\pi r^2} \sum_{i=1}^{n} K(\frac{(h,p) - (h_i, p_i)}{r}), \tag{7.3}$$

where $K(\cdot)$ is a kernel function which assigns a weight to the point $(h_i,\ p_i)$ within the area depending on its distance from $(h,\ p)$. The points that are closer to $(h,\ p)$ are assigned larger weights. Different types of the kernel function such as linear kernel, polynomial kernel, radial basis function kernel, and multilayer perceptron kernel can be investigated. Based on the obtained density of each point, we can divide the domain into consecutive blocks, which satisfy the condition that the sum of the density of each point in the block equals to a preset threshold.

4. Analyze signals in addition to IUP.

   The intrauterine catheter is not available in some developing countries, where an automatic anesthesia equipment is needed the most due to the shortage of experienced anesthetists. Therefore, the next step of this work should include designing a more advanced noise filtering or signal recovering technique to recognize contractions from signals obtained by external measurement, such as tocodynamometer, so as to enable further analysis.

   As an alternative and based on literature review, we know that the uterine contraction is directly related to the electrical activities in the myometrial cells. Uterine EMG signals represent electrical activities, which as a whole trigger uterine contractions. As discussed earlier, a single spike is not enough to maintain a forceful contraction. Therefore, if only a single spike is detected in an EMG signal, the spike should be considered as noise. On the other hand, if multiple, high-frequency, and coordinated spikes are detected, a contraction is probably occurring. Compared to the IUPC signal, the EMG signal contains

more frequency information, and it could be analyzed in the frequency domain. In this case, there are a number of sophisticated mathematical methods (power spectrum, wavelets, fractals, and artificial neural networks, to name a few) that could be used to determine the extent of electrochemical preparedness of the myometrium for labor and subsequent delivery. To start the analysis, a fast Fourier transform (FFT) can be used to obtain the frequency spectrum of the EMG signal. Figure 7.4 shows an example.

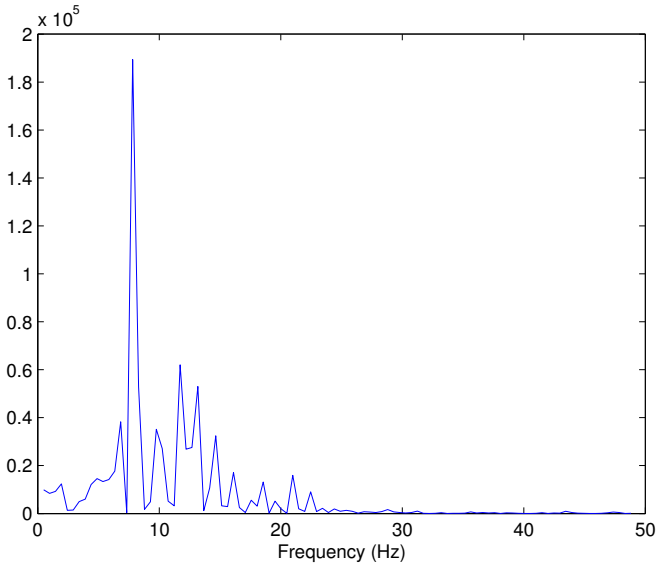

Figure 7.4: Frequency Spectrum

The peaks in the frequency spectrum represent the frequencies with strong intensity. We select N major frequencies from the peaks w_k, where $k = 1, \cdots, N$. The original signal $y(t)$ can be approximated by the selected frequencies according to Equation (7.4).

$$\tilde{y}(t) = a_0 + \sum_{k=1}^{N} a_k \sin(w_k t + \varphi_k), \tag{7.4}$$

where $\tilde{y}(t)$ is the reconstructed signal, and a_0, a_k and φ_k are parameters needed to be obtained through modeling. Equation (7.4) can be rewritten in the following form.

$$\tilde{y}(t) = a_0 + \sum_{k=1}^{N} [a_{k,1} \sin(w_k t) + a_{k,2} \cos(w_k t)] = \theta^T x(t), \tag{7.5}$$

where $a_{k,1}$ and $a_{k,2}$ are coefficients, $\theta^T = [a_0, a_{1,1}, a_{1,2}, \cdots, a_{k,1}, a_{k,2}, \cdots, a_{N,1}, a_{N,2}]$, and

$$x(t) = \begin{bmatrix} 1 \\ \sin(w_1 t) \\ \cos(w_1 t) \\ \vdots \\ \sin(w_k t) \\ \cos(w_k t) \\ \vdots \\ \sin(w_N t) \\ \cos(w_N t) \end{bmatrix}.$$

We can use the least squares algorithm to obtain the optimum θ. Further analysis and prediction can be done based on the obtained model in the frequency domain. Besides using the EMG signals, we could also explore if there are any other alternative methods that could monitor the electrical activities. Understanding these methods is important to build a good predictor, given that the electrical activity is a crucial factor that causes the uterine contractions.

APPENDIX A

Gamma Test

Gamma test (Stefánsson, Koncar, and Jones 1997) is a nonparametric noise estimation technique which estimates the variance of the noise. Gamma test was introduced for model selection and also for input selection (Rubio, Pomares, Rojas, and Herrera 2011; Sorjamaa, Hao, Reyhani, Ji, and Lendasse 2007), while it is employed in our framework on the purpose of parameters selection. Gamma test is based on the assumption that if two points are very close in the input space, the corresponding output should be close enough. Otherwise, the deviation is caused by noise.

Given M data instances: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\}$, where $x \in \mathbb{R}^p$ and $y \in \mathbb{R}$. The relationship between the input and the output can be formulated as in Equation (A.1).

$$y_i = f(x_i) + r_i, i \in [1, M] \tag{A.1}$$

where f is the underlying function which captures the relationship between x_i and y_i. r_i is the noise for the ith data instance. We can assume that the mean of r is zero, because any bias can be considered as a part of the underlying function f. Gamma test is used to estimate the variance of r, denoted as σ_r^2.

Gamma test estimates the variance σ_r^2 by calculating the vertical intercept of the linear regression line: $\eta(k) = A\delta(k) + B$, $1 \leq k \leq p$, where A is the slope parameter and B is the vertical intercept. p is usually set to 10. $\eta(k)$ and $\delta(k)$ are defined in Equation (A.2) and Equation (A.3), respectively.

$$\eta(k) = \frac{1}{2M} \sum_{i=1}^{M} (y_{[i,k]} - y_i)^2; \tag{A.2}$$

$$\delta(k) = \frac{1}{M} \sum_{i=1}^{M} \left| x_{[i,k]} - x_i \right|^2, \tag{A.3}$$

where $x_{[i,k]}$ is the kth nearest neighbor of x_i, and $y_{[i,k]}$ is the output value corresponding to $x_{[i,k]}$. Compute the regression line of the p points $(\delta(k), \eta(k))$, $1 \leq k \leq p$. σ_r^2 is estimated by the vertical intercept B of the regression line. When the number of the data instances is large enough, B is a reliable estimation of the variance.

APPENDIX B

Glossary

Agonist A substance interacting with a receptor molecule that initiates the same response as the hormone/transmitter usually binds to that site.

ARM Association rule mining.

BMI Body mass index.

CA Correspondence analysis.

CF Collaborative filtering.

ECG Electrocardiography.

EMG Electromyography, a technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is performed using an instrument called an electromyograph, to produce a record called an electromyogram. An electromyograph detects the electrical potential generated by muscle cells when these cells are electrically or neurologically activated.

γ Regularization parameter for LS-SVM, which controls the trade-off between the training error minimization and the smoothness of the model.

Gap junctions Gap junctions are formed from bundles of proteins, called connexins, which align forming symmetrical channels protruding through adjacent cells and thus allowing contact and communication (Coad and Dunstall 2001).

KDD Knowledge discovery in databases.

k-NN k-nearest neighbors.

LOOCV Leave-one-out cross-validation, a technique that involves using a single observation from the original sample as the validation data, and the remaining observations as the training data.

LS-SVM Least squares support vector machines.

MCA Multiple correspondence analysis.

MLP Multilayer perception.

Myometrium The myometrium is the middle layer of the uterine wall, consisting mainly of uterine smooth muscle cells, but also of supporting stromal and vascular tissue. Its main function is to induce uterine contractions.

Overfitting When an algorithm searches for the best parameters for one particular model using a set of training data, it can model not only the general patterns in the data but also any noise specific to the training data set, resulting in poor performance of the model on testing data. This issue is called overfitting. Possible solutions include cross-validation, regularization, and other sophisticated statistical strategies.

PCA Patient-controlled analgesia.

RBF kernel Radial basis function kernel.

RMSE Root mean squared error.

σ The bandwidth for the RBF kernel.

SMAPE Symmetric mean absolute percentage error.

TOCO Tocodynamometer.

Bibliography

Agrawal, R., Imieliński, T., and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD International Conference on Management of Data*. Washington, DC, USA, 207–216.

Agrawal, R. and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proc. of the 20th International Conference on Very Large Data Bases*. Santiago, Chile, 487–499.

Akerlund, M. 1997. Contractility in the nonpregnant uterus. *Annals of the New York Academy of Sciences 828*, 213–222.

Baxi, L. V. and Petrie, R. H. 1987. Pharmacologic effects on labor: Effects of drugs on dystocia, labor and uterine activity. *Clin Obstet Gynecol 30*, 19–32.

Bontempi, G. 2008. Long term time series prediction with multi-input multi-output local learning. In *Proc. of the 2nd European Symposium on Time Series Prediction*. Porvoo, Finland, 145–154.

Boonjing, V. and Songram, P. 2007. Efficient algorithms for mining closed multidimensional sequential patterns. In *Proc. of the 4th International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 2. Haikou, China, 749–753.

Challis, J. R. and Lye, S. J. 1994. Parturition. In *The Physiology of Reproduction (2nd Ed.)*, E. Knobil, J. D. Neill, G. S. Greenwald, C. L. Markert, and D. W. Pfaff, Eds. Raven Press, 985–1031.

Chang, J. H. and Lee, W. S. 2003. Finding recent frequent itemsets adaptively over online data streams. In *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, D.C.

Chang, J. H. and Lee, W. S. 2004. A sliding window method for finding recently frequent itemsets over online data streams. *Journal of Information Science and Engineering 20*, 753–762.

Chang, M.-W. and Lin, C.-J. 2005. Leave-One-Out bounds for support vector regression model selection. *Neural Comput. 17*, 1188–1222.

Cheng, J., Ke, Y., and Ng, W. 2008. A survey on algorithms for mining frequent itemsets over data streams. *Knowledge and Information Systems 16*, 1–27.

Cheung, D. W. and Xiao, Y. 1998. Effect of data skewness in parallel mining of association rules. In *Proc. of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Vol. 1394. Springer Verlag, Melbourne, Australia, 48–60.

Chi, Y., Wang, H., Yu, P., and Muntz, R. 2004. Moment: maintaining closed frequent itemsets over a stream sliding window. In *Proc. of the 4th IEEE International Conference on Data Mining*. Brighton, UK, 59–66.

Chuang, K.-T., Chen, M.-S., and Yang, W.-C. 2005. Progressive sampling for association rules based on sampling error estimation. In *Advances in Knowledge Discovery and Data Mining*, T. Ho, D. Cheung, and H. Liu, Eds. Lecture Notes in Computer Science, vol. 3518. Springer Berlin/Heidelberg, 37–44.

Clement, M. P. and Hendry, D. F. 1998. *Forecasting Economic Times Series*. Cambridge University Press.

Coad, J. and Dunstall, M. 2001. *Anatomy and Physiology for Midwives*. Mosby.

Coyle, D. 2009. Neural network based auto association and time-series prediction for biosignal processing in brain-computer interfaces. *IEEE Computational Intelligence Magazine 4,* 4 (November), 47–59.

Cristi, R. and Tummala, M. 2000. Multirate, multiresolution, recursive kalman filter. *Signal Process 80,* 9, 1945–1958.

Crone, S. F. 2010. Artificial neural network & computational intelligence forecasting competition.

Das, A., Ng, W.-K., and Woon, Y.-K. 2001. Rapid association rule mining. In *Proc. of the 10th International Conference on Information and Knowledge Management*. Atlanta, Georgia, USA, 474–481.

Duquette, R., Shmygol, A., Vaillant, C., Mobasheri, A., Pope, M., Burdyga, T., and Wray, S. 2005. Vimentin-positive, c-KIT-negative interstitial cells in human and rat uterus: a role in pacemaking? *Biology of Reproduction 72*, 276–283.

Evron, S., Glezerman, M., Sadan, O., Boaz, M., and Ezri, T. 2005. Remifentanil: a novel systemic analgesic for labor pain. *Anesthesia & Analgesia 100,* 1 (January), 233–238.

Farooq, T., Guergachi, A., and Krishnan, S. 2007. Chaotic time series prediction using knowledge based green's kernel and least-squares support vector machines. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*. Montreal, Quebec, Canada, 373–378.

Fayyad, U., Piatetsky-shapiro, G., and Smyth, P. 1996. From data mining to knowledge discovery in databases. *AI Magazine 17*, 37–54.

Friedrichs, F. and Igel, C. 2004. Evolutionary tuning of multiple SVM parameters. In *Proc. of the 12th European Symposium on Artificial Neural Networks*. Bruges, Belgium, 519–524.

Garfield, R. E. 1987. Cellular and molecular basis for dystocia. *Clin Obstet Gynecol 30,* 1, 3–18.

Garfield, R. E., Ali, M., Yallampalli, C., and Izumi, H. 1995. Role of gap junctions and nitric oxide in control of myometrial contractility. *Semin Perinatol 19*, 41–51.

Garfield, R. E., Blennerhassett, M. G., and Miller, S. M. 1988. Control of myometrial contractility: role and regulation of gap junctions. *Oxf Rev Reprod Biol 10*, 436–490.

Garfield, R. E. and Maner, W. L. 2007. Physiology and electrical activity of uterine contractions. *Seminarsin Cell & Developmental Biology 18,* 3 (March/April), 289–295.

Giannella, C., Han, J., Pei, J., Yan, X., and Yu, P. S. 2003. Mining frequent patterns in data streams at multiple time granularities. In *Data Mining: Next Generation Challenges and Future Directions*, H. Kargupta, A. Joshi, D. Sivakumar, and Y. Yesha, Eds. MIT/AAAI Press, 191–212.

Glass, P., Hardman, D., Kamiyama, Y., Quill, T., Marton, G., Donn, K., Grosse, C., and Hermann, D. 1993. Preliminary pharmacokinetics and pharmacodynamics of an ultra short acting opioid: remifentanil. *Anesthesia & Analgesia 77,* 5 (November), 1031–1040.

Guo, X., Yang, J., Wu, C., Wang, C., and Liang, Y. 2008. A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. *Neurocomputing 71,* 16-18, 3211–3215.

Han, J., Gong, W., and Yin, Y. 1998. Mining segment-wise periodic patterns in time-related databases. In *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, 214–218.

Han, J., Lu, H., and Feng, L. 1998. Stock movement prediction and n-dimensional inter-transaction association rules. In *Proc. of the 1998 SIGMOD Workshop Research Issues on Data Mining and Knowledge Discovery*. Vol. 12. Seattle, WA, USA, 1–7.

Han, J. and Pei, J. 2000. Mining frequent patterns by pattern-growth: methodology and implications. *SIGKDD Explor. Newsl. 2*, 14–20.

Hathaway, D. H. 2011. Royal greenwich observatory - USAF/NOAA sunspot data.

Herrera, L. J., Pomares, H., Rojas, I., Guilln, A., Prieto, A., and Valenzuela, O. 2007. Recursive prediction for long term time series forecasting using advanced models. *Neurocomputing 70,* 16-18 (May), 2870–2880.

Hill, D. 2008. The use of remifentanil in obstetrics. *Anesthesiology Clinics 26,* 1 (March), 169–182.

Homma, N., Sakai, M., and Takai, Y. 2009. Time series prediction of respiratory motion for lung tumor tracking radiation therapy. In *Proc. of the 10th WSEAS International Conference on Neural networks*. Prague, Czech Republic, 126–131.

Hu, Y., Mak, J., and Luk, K. 2009. Effect of electrocardiographic contamination on surface electromyography assessment of back muscles. *Journal of Electromyography and Kinesiology 19,* 1 (February), 145 – 156.

Huang, K., Chang, C., and Lin, K. 2004. Prowl: an efficient frequent continuity mining algorithm on event sequences. In *Proc. of the 6th International Conference on Data Warehousing and Knowledge Discovery, Vol. 3181 of Lecture Notes in Computer Science*. Aix-en-Provence, France, 351–360.

Huang, Z. and Shyu, M.-L. 2010. k-NN based LS-SVM framework for long-term time series prediction. In *Proc. of the 2010 IEEE International Conference on Information Reuse and Integration*. Las Vegas, NV, USA, 69–74.

Huang, Z. and Shyu, M.-L. 2012. Long-term time series prediction using k-NN based LS-SVM framework with multi-value integration. In *Recent Trends in Information Reuse and Integration*, T. Özyer, K. Kianmehr, and M. Tan, Eds. Springer Vienna, 191–209.

Huang, Z., Shyu, M.-L., and Tien, J. M. 2012. LS-SVM based framework with heuristic parameter tuning for long-term time series prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* submitted.

Huebner, U., Abraham, N. B., and Weiss, C. O. 1989. Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH_3 laser. *Phys. Rev. A 40,* 11, 6354–6365.

Jain, A., Duin, R., and Mao, J. 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22,* 1 (January), 4–37.

Jang, J.-S. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics 23,* 3, 665–685.

Jiang, N. and Gruenwald, L. 2006. Research issues in data stream association rule mining. *SIGMOD Record 35*, 14–19.

Jiang, T., Wang, S., and Wei, R. 2007. Support vector machine with composite kernels for time series prediction. In *Proc. of the 4th International Symposium on Neural Networks*. Nanjing, China, 350–356.

Jones, A. J. 2004. New tools in non-linear modelling and prediction. *Comput. Manag. Sci*, 109–149.

Jones, R. H. 1966. Exponential smoothing for multivariate time series. *Journal of the Royal Statistical Society. Series B (Methodological) 28,* 1, 241–251.

Kotsiantis, S. and Kanellopoulos, D. 2006. Association rules mining: a recent overview. *GESTS International Transactions on Computer Science and Engineering*.

Kusiak, A., Zheng, H., and Song, Z. 2009. Short-term prediction of wind farm power: a data mining approach. *IEEE Transactions on Energy Conversion 24,* 1, 125–136.

La Rosa, P., Nehorai, A., Eswaran, H., Lowery, C., and Preissl, H. 2008. Detection of uterine MMG contractions ssing a multiple change point estimator and the k-means cluster algorithm. *IEEE Transactions on Biomedical Engineering 55,* 2 (February), 453–467.

Lee, D. and Lee, W. 2005. Finding maximal frequent itemsets over online data streams adaptively. In *Proc. of the 5th IEEE International Conference on Data Mining*. 266–273.

Lendasse, A., Ji, Y., Reyhani, N., and Verleysen, M. 2005. LS-SVM hyperparameter selection with a nonparametric noise estimator. In *Proc. of the 15th International Conference on Artificial Neural Networks*. Warsaw, Poland, 625–630.

Lin, K., Lin, Q., Zhou, C., and Yao, J. 2007. Time series prediction based on linear regression and SVR. *Proc. of the 3rd International Conference on Natural Computation 1*, 688–691.

Linden, G., Smith, B., and York, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing 7,* 1 (January/February), 76–80.

Liu, P. and Yao, J. 2009. Application of least square support vector machine based on particle swarm optimization to chaotic time series prediction. *Proc. of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems 4*, 458–462.

Lotric, U. 2004. Wavelet based denoising integrated into multilayered perceptron. *Neurocomputing 62*, 179–196.

Maralloo, M., Koushki, A., Lucas, C., and Kalhor, A. 2009. Long term electrical load forecasting via a neurofuzzy model. In *Proc. of the 14th International CSI Computer Conference*. Tehran, Iran, 35–40.

Marshall, J. M. 1959. Effects of estrogen and progesterone on single uterine muscle fibers in the rat. *American Journal of Physiology 197*, 935–942.

Marshall, J. M. 1962. Regulation of activity in uterine smooth muscle. *Physiological Reviews 42*, 213–227.

Mathworks.com. 2008. Interpolation.

Menezes, Jr., J. M. P. and Barreto, G. A. 2008. Long-term time series prediction with the NARX network: an empirical evaluation. *Neurocomput 71,* 16-18, 3335–3343.

Meng, K., Dong, Z., and Wong, K. 2009. Self-adaptive radial basis function neural network for short-term electricity price forecasting. *Generation, Transmission Distribution, IET 3,* 4, 325–335.

Nguyen, H. H. and Chan, C. W. 2004. Multiple neural networks for a long term time series forecast. *Neural Computing & Applications 13*, 90–98.

OB-Tools. 2011. Electrical uterine monitor.

O'Haver, T. 2011. Peak finding and measurement.

Ormandi, R. 2008. Variance minimization least squares support vector machines for time series analysis. In *Proc. of the 8th IEEE International Conference on Data Mining*. Pisa, Italy, 965–970.

Ozden, B., Ramaswamy, S., and Silberschatz, A. 1998. Cyclic association rules. In *Proc. of the 14th International Conference on Data Engineering*. Orlando, FL, USA, 412–421.

Palomar Pomerado Health. 2011. External and internal heart rate monitoring of the fetus.

Parthasarathy, S., Zaki, M. J., Ogihara, M., and Li, W. 2001. Parallel data mining for association rules on shared memory systems. *Knowledge and Information Systems 3*, 1–29.

Pelckmans, K., Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Lukas, L., Moor, B. D., and Vandewalle, J. 2003. LS-SVMlab toolbox user's guide. *ESAT-SCD-SISTA Technical Report*, 1–106.

Piatetsky-Shapiro, G. 1991. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. J. Frawley, Eds. AAAI/MIT Press, Cambridge, MA.

Puma-Villanueva, W. J., dos Santos, E., and Von Zuben, F. 2007. Long-term time series prediction using wrappers for variable selection and clustering for data partition. In *Proc. of the 2007 International Joint Conference on Neural Networks*. Orlando, Florida, 3068–3073.

Rabotti, C., Mischi, M., Oei, S., and Bergmans, J. 2010. Noninvasive estimation of the electrohysterographic action-potential conduction velocity. *IEEE Transactions on Biomedical Engineering 57,* 9 (September), 2178–2187.

Renaud, O., Starck, J.-L., and Murtagh, F. 2005. Wavelet-based combined signal filtering and prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35,* 6, 1241–1251.

Rubio, G., Pomares, H., Rojas, I., and Herrera, L. J. 2011. A heuristic method for parameter selection in LS-SVM: Application to time series prediction. *International Journal of Forecasting 27,* 3, 725–739.

Sapankevych, N. and Sankar, R. 2009. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine 4,* 2, 24–38.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th International Conference on World Wide Web*. New York, NY, USA, 285–295.

Saunders, T. A. and Glass, P. S. 2002. A trial of labor for remifentanil. *Anesthesia & Analgesia 94,* 4 (April), 771–773.

Schuster, A. and Wolff, R. 2004. Communication-efficient distributed mining of association rules. *Data Minning and Knowledge Discovery 8*, 171–196.

Sfetsos, A. and Siriopoulos, C. 2004. Time series forecasting with a hybrid clustering scheme and pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 34,* 3, 399–405.

Shyu, M.-L., Huang, Z., and Luo, H. 2009. Efficient mining and detection of sequential intrusion patterns for network intrusion detection systems. In *Machine Learning in Cyber Trust, Part 2*, J. Tsai and P. Yu, Eds. Springer, 133–154.

Soltani, S., Boichu, D., Simard, P., and Canu, S. 2000. The long-term memory prediction by multiscale decomposition. *Signal Processing 80,* 10, 2195–2205.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. 2007. Methodology for long-term prediction of time series. *Neurocomputing 70,* 16-18, 2861–2869.

Srikant, R. and Agrawal, R. 1995. Mining generalized association rules. In *Proc. of the 21st International Conference on Very Large Data Bases*. San Francisco, CA, USA, 407–419.

Srikant, R. and Agrawal, R. 1996. Mining quantitative association rules in large relational tables. In *Proc. of the 1996 ACM SIGMOD International Conference on Management Data*. Montreal, Quebec, Canada, 1–12.

Stefánsson, A., Koncar, N., and Jones, A. J. 1997. A note on the gamma test. *Neural Computing & Applications 5*, 131–133.

Su, X. and Khoshgoftaar, T. M. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence 2009*, 1–19.

Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D., and Vandewalle, J. 2002. *Least Squares Support Vector Machines*. World Scientific, Farrer Road, Singapore.

Taieb, S. B., Bontempi, G., Sorjamaa, A., and Lendasse, A. 2009. Long-term prediction of time series by combining direct and MIMO strategies. *Proc. of the International Joint Conference on Neural Networks*, 3054–3061.

Terrien, J., Marque, C., and Germain, G. 2008. Ridge extraction from the time-frequency representation (TFR) of signals based on an image processing approach: application to the analysis of uterine electromyogram AR TFR. *IEEE Transactions on Biomedical Engineering 55,* 5 (May), 1496–1503.

Tung, A., Lu, H., Han, J., and Feng, L. 2003. Efficient mining of intertransaction association rules. *IEEE Transactions on Knowledge and Data Engineering 15,* 1, 43–56.

Varadan, V., Leung, H., and Bosse, E. 2006. Dynamical model reconstruction and accurate prediction of power-pool time series. *IEEE Transactions on Instrumentation and Measurement 55,* 1 (February), 327–336.

Volmanen, P., Akural, E. I., Raudaskoski, T., and Alahuhta, S. 2002. Remifentanil in obstetric analgesia: a dose-finding study. *Anesthesia & Analgesia 94,* 4 (April), 913–917.

Wang, Y., Hou, Z., and Zhou, X. 2006. An incremental and hash-based algorithm for mining frequent episodes. In *Proc. of the International Conference on Computational Intelligence and Security*. Vol. 1. Guangzhou, China, 832–835.

Wei, H. L. and Billings, S. A. 2006. Long term prediction of non-linear time series using multiresolution wavelet models. *International Journal of Control 79,* 6, 569–580.

Weigend, A. and Gershenfeld, N. 1994. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley Publishing Company.

Witten, I. H. and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Wojciechowski, M. and Zakrzewicz, M. 2002. Dataset filtering techniques in constraint-based frequent pattern mining. *Lecture Notes in Computer Science 2447*, 77–83.

Wolfs, G. and Rottinghuis, H. 1970. Electrical and mechanical activity of the human uterus during labour. *Archives of Gynecology and Obstetrics 208*, 373–385.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., and Steinberg, D. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems 14*, 1–37.

Xu, R. and Wunsch, D., I. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks 16,* 3 (May), 645–678.

Yegnanarayana, B. 2004. *Artificial Neural Networks*. Prentice-Hall of India Pvt.Ltd.

Zhang, G. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing 50*, 159–175.

Zhang, S., Huang, Z., Zhang, J., and Zhu, X. 2008. Mining follow-up correlation patterns from time-related databases. *Knowledge and Information Systems 14,* 1, 81–100.

Zhang, S., Zhang, J., Zhu, X., and Huang, Z. 2006. Identifying follow-correlation itemset-pairs. In *Proc. of the 6th IEEE International Conference on Data Mining*. 765–774.

Zhou, Z., Xu, Z., and Wu, W. B. 2010. Long-term prediction intervals of time series. *IEEE Transactions on Information Theory 56,* 3 (March), 1436–1446.

Zou, H. and Yang, Y. 2004. Combining time series models for forecasting. *International Journal of Forecasting 20,* 1, 69–84.