ONLINE MONITORING AND PREDICTION OF COMPLEX TIME SERIES EVENTS FROM NONSTATIONARY TIME SERIES DATA

BY SHOUYI WANG

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Wanpracha Art Chaovalitwongse

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2012

ABSTRACT OF THE DISSERTATION

Online Monitoring and Prediction of Complex Time Series Events From Nonstationary Time Series Data

by Shouyi Wang

Dissertation Director: Wanpracha Art Chaovalitwongse

Much of the world's supply of data is in the form of time series. In the last decade, there has been an explosion of interest in time series data mining. Time series prediction has been widely used in engineering, economy, industrial manufacturing, finance, management and many other fields. Many new algorithms have been developed to classify, cluster, segment, index, discover rules, and detect anomalies/novelties in time series. However, traditional time series analysis methods are limited by the requirement of stationarity of the time series and normality and independence of the residuals. Because they attempt to characterize and predict all time series observations, traditional time series analysis methods are unable to identify complex (nonperiodic, nonlinear, irregular, and chaotic) characteristics. As a result, the prediction of multivariate noisy time series (such as physiological signals) is still very challenging due to high noise, non-stationarity, and non-linearity.

The objective of this research is to develop new reliable frameworks for analyzing multivariate noisy time series, and to apply the framework to online monitor noisy time series and predict critical events online. In particular, this research made an extensive study on one important form of multivariate time series: electroencephalography (EEG) data, based on which two new online monitoring and prediction frameworks for multivariate time series were introduced and evaluated. The new online monitoring and

prediction frameworks overcome the limitations of traditional time series analysis techniques, and adapt and innovate data mining concepts to analyzing multivariate time series data. The proposed approaches can be general frameworks to create a set of methods that reveal hidden temporal patterns that are characteristic and predictive of time series events.

In second part of this dissertation provide an overview of the state-of-the-art prediction approaches. In the third part of this dissertation, we perform an extensive data mining study on multivariate EEG data, which indicates that EEG may be predictable for some events. In chapter 4, a reinforcement learning-based online monitoring and prediction framework is introduced and applied to solve the challenging seizure prediction problem from multivariate EEG data. In chapter 5, it first overview of the most popular representation methods for time series data, and then introduce two new robust algorithms for offline and online segmentation of a time series, respectively. Chapter 6 proposes a general online monitoring and prediction framework, which combines temporal feature extraction, feature selection, online pattern identification, and adaptive learning theory to achieve online prediction of complex time series events. Two prediction-rule construction schemes are proposed. In chapter 7, the proposed framework is applied to solve two challenging problems including seizure prediction and 'anxiety' prediction in a simulated driving environment. The significant prediction results demonstrated the superior prediction capability of the proposed framework to predict complex target events from online streams of nonstationary and chaotic time series.

Acknowledgements

I would like to thank Dr. W. Art Chaovalitwongse for the encouragement, support, and direction he has provided during the past four years. His insightful suggestions and enthusiastic endorsement have made the completion of this research possible. I thank Dr. Changxu Wu, for offering valuable insights, supports and experimental platform at all of the key moments in the development of this work. I thank Dr. Myong Jeong, for helping me in the data mining research, and expanding the breadth of my research. I extend special thanks to Dr. Stephen Wong, for helping me a lot in the early stage of seizure prediction research. I also thank Dr. Susan Albin, for being my committee member and providing me insights into her areas of expertise.

I am grateful to Rutgers University for its financial support of this research, and the faculty of the Industrial and Systems Engineering Department for sharing their support and passion for this research, and providing an environment in which I might succeed. I thank all of the members of the COSMOLAB, past and present, for their wonderful discussions and advice along the way.

I thank my father and mother, Yijun Wang and Meili Zhao, for encouraging my efforts, even when it was not clear to me where they would lead. Finally, I would like to thank my wife, Fei Zheng, for her ongoing moral support, during the best and worst of it.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Time series data accounts for an increasingly large fraction of the world's supply of data. Given the ubiquity of time series data, and the exponentially growing sizes of databases, there has been recently been an explosion of interest in time series Data Mining. The major task of the time series data mining are mainly as follows:

- Indexing: Given a query time series, and some similarity/dissimilarity measure, find the most similar time series in a time series database.
- Clustering: Find natural groupings of the time series in database given some similarity/dissimilarity measure.
- Classification: Given an unlabeled time series, assign it to one of two or more predefined classes.
- Prediction: Given a time series, predict the values of the time series in the following time points.
- Anomaly Detection: Given a baseline time series which is assumed to be normal, and find 'abnormal' sections of an unannotated time series which contain anomalies or unexpected occurrences.

Prediction can be viewed as a type of clustering or classification. The difference is that prediction is predicting a future state, rather than a current one. Time series prediction is fundamental to engineering, scientific, and business endeavors. Researchers study systems as they evolve through time, hoping to discern their underlying principles and develop models useful for predicting or controlling them. Some important applications include obtaining forewarning of natural disasters (flooding, hurricane,

snowstorm, etc), epidemics, stock crashes, etc. Many techniques have been proposed for time series prediction.

Although time series has been worked with for more than a century, many of the existing techniques hold little utility for researchers working with massive time series databases. The transitional machine learning and data mining algorithms do not work well on massive time series data mainly due to following reasons:

- Time series data are often massive in volumes. In the medical domain alone, large volumes of data as diverse as electrocardiograms, electroencephalograms, gait analysis and growth development charts are routinely created. Similar remarks apply to industry, entertainment, finance, meteorology and virtually every other field of human endeavors. While classic time series data mining algorithms assume relatively low dimensionality.

- It is often the case that time series data have very high dimensionality, high non-stationary, and large amount of noise, which present a difficult challenge in time series data mining tasks [16]. However, the traditional algorithms are generally static and nonadaptive due to their unique structures. For example, the Box-Jenkins or Autoregressive Integrated Moving Average (ARIMA) method is widely applied to various time series problems. However, the ARIMA method is seriously limited by the requirement of stationarity of the time series and normality and independence of the residuals [14]. Similar to ARIMA, many existing algorithms require the statistical characteristics of a stationary time series remain constant through time, prediction errors must be uncorrelated and normally distributed. As a result, the severe drawback of these approaches are their inability to identify complex non-stationary characteristics of a time series.

The objective of this research is to overcome the limitations of the traditional time series approaches, and to uncover complex and hidden patterns for massive non-stationary time series data. This body of this work draws on the fields of data mining, machine learning, statistics, signal processing, and mathematics. In particular, this dissertation describes a set of adaptive learning methods that reveal complex hidden

patterns in time series data. The adaptive time series data mining frameworks, introduced by this dissertation, are fundamental contributions to the fields of time series analysis and data mining. The adaptive learning frameworks allows the proposed prediction methods are able to successfully characterize and predict complex patterns for nonperiodic, irregular, and chaotic time series. The proposed methods overcome limitations (including stationarity and linearity requirements) of traditional time series analysis techniques by adapting adaptive learning mining concepts for analyzing time series. The new methods are applicable to time series that appear stochastic, but occasionally (though not necessarily periodically) contain distinct, but possibly hidden, patterns that are characteristic of the desired events. For example, the challenging problem of seizure prediction from multivariate EEG data is well suited to the new adaptive prediction frameworks.

The dissertation is divided into seven chapters. Chapter 2 reviews several of the start-of-the-art prediction models and as well as a recent advances in machine learning and data mining.

Chapter 3 studies the possibility of early detection of numerical typing errors from EEG recordings. The objective of this study is to perform various data mining techniques on EEG time series, and evaluate if EEG time series can be employed to predict a future event of the brain.

Chapter 4 investigates the challenging problem of epileptic seizure prediction problem. We introduced an adaptive seizure prediction framework, which combines reinforcement learning, online monitoring and adaptive control theory to advance the flexibility and adaptability of the prediction system. Using EEG recordings from five patients with epilepsy, we have demonstrated that the adaptive learning framework considerably improved the prediction performance of the system.

Chapter 5 firstly overviews the current pattern representations methods of time series. Inspired by the top-down decomposition structure, a new decomposition algorithm is proposed to extract key skeleton points of time series. Two new statistic measures are also introduced. Based on the new measures, the stop criterion of the decomposition algorithm can be determined. The numerical studies show that the proposed skeleton

extraction technique is generally robust and computational efficiency.

Chapter 6 proposes a general online monitoring and prediction framework, which combines temporal feature extraction, feature selection, online pattern identification, and adaptive learning theory to achieve online prediction of complex time series events. Two prediction-rule construction schemes are proposed.

Chapter 7, the proposed framework is applied to solve two challenging problems including seizure prediction and 'anxiety' prediction in a simulated driving environment. The significant prediction results demonstrated the superior prediction capability of the proposed framework to predict complex target events from online streams of nonstationary and chaotic time series.

Chapter 8 summarizes the dissertation and discusses the possible future research directions.

Chapter 2

Start-of-the-Art Prediction Models

Prediction is a very important aspect of any business, and has enormous social, economic, and environmental impacts. Many prediction models have been developed to empower people in decision-making for various application areas. For example, accurate demand forecasts are essential for manufacturers to determine the optimal production rate by making a tradeoff between stock-outs and high inventory levels. Successful climate prediction models have been developed to provide early warnings of adverse climatic conditions, such as hurricanes, storms, or frogs [146]. In business activities, forecasting technologies have become indispensable tools in a wide range of managerial decision-making processes, such as finance, banking, investments, employment, mortgages and loans [5].

Based on historical time series data, forecasts can be made based on either empirical qualitative analysis or mathematical quantitative analysis. Accordingly, forecasting models can be broadly classified as qualitative methods and quantitative methods. The categorization of the current most popular prediction models is shown in Figure 2.1. Qualitative prediction techniques rely primarily on human judgment based on expertise, experience, or intuition. They can be used in a wide range of circumstances where historical data are not available, or circumstances that are changing so rapidly that a mathematical forecasting model based on past data may become irrelevant or questionable. An overview of the qualitative prediction methods, including Delphi method, Jury of Expert Opinion, Scenario Analysis, Sales Force Composite, and Market Survey, can be referred to [158].

Quantitative methods make forecasts based on mathematical models rather than subjective judgment. These methods are the mainstream of forecasting techniques as

a result of the great advances in mathematical modeling and computational power in modern times. Quantitative forecasting models have been utilized across a wide spectrum of business and industry. As shown in Figure 2.1, quantitative methods can be classified as non-causal models and causal models [60]. Non-causal models are also known as time-series models, which make forecasts by extracting systematic patterns (such as trends and seasonality) from historical time series data. Causal models are also known as cause-and-effect models, which investigate how the variable being forecasted is determined by its relevant influential factors. This chapter briefly summarizes the state-of-the-art quantitative prediction methods in terms of basic procedure, underlying assumptions, applications and limits.

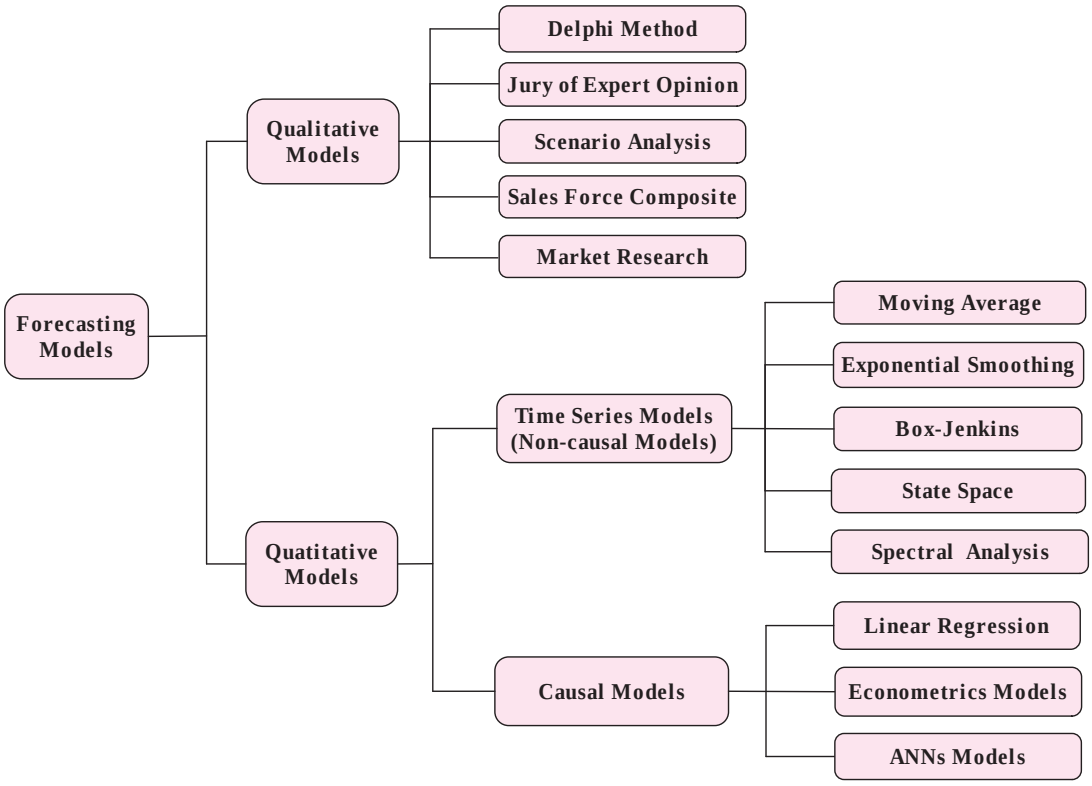


Figure 2.1: Categorization of prediction models.

2.1 Time Series Prediction Methods

Non-causal models predict values of the variable being forecasted based on historical patterns. Thus the basic underlying assumption of these methods is that future patterns are similar to historical patterns. To extract characteristics of time series patterns, four basic properties of data are often analyzed:

- Trend. Given a set of time series data, the term trend refers to a stable tendency of growth or decline exhibited in the data. The trend of a time series can be either linear or nonlinear. Accordingly, linear and nonlinear functions can be utilized to model the trend.

- Seasonality. If a pattern always repeats at a fixed interval, it is called a seasonal pattern. Seasonality is a very common characteristic of time series data. For example, air temperature exhibits a strong yearly seasonal pattern.

- Cycles. Cyclic patterns are similar to seasonal patterns, except that they repeat at varying intervals. For example, it is common to find nonstationary cycles in financial time series data.

- Randomness. Most time series data are assumed to contain both systematic patterns and random noises. The randomness usually makes the pattern difficult to identify. Most time series models include a noise term to take into account the effects of randomness.

Various time series methods have been developed to analyze these properties of time series data. Five of the most popular ones are moving average, exponential smoothing, Box-Jenkins models, state-space models, and spectral methods.

2.1.1 Moving Average (MA)

MA models are simple but popular forecasting methods in time series analysis. A MA model involves taking arithmetic average of N most recent observations, where N is a specified number according to the nature of the data to be forecasted. For example,

if you are forecasting monthly sales, you might use 12-month MA model, which takes the average sales over the past 12 months. A one-step-ahead MA model of N periods is given by

$$F_{t+1} = \frac{1}{N} \sum_{i=t-N+1}^{t} x_i = \frac{x_t + x_{t-1} + \cdots + x_{t-N+1}}{N}, \tag{2.1}$$

where F_{t+1} is the forecast for the $t+1$ period, and x_is, $i = t-N+1, \ldots, t$ are the observations in the past N periods. The mean of N most recent observations is used as the forecast of the next period. The moving averages method is probably the most commonly used technique to smooth out short-term fluctuations and capture characteristics of varying trends in time series data.

2.1.2 Exponential Smoothing

Exponential smoothing assigns exponentially decreasing weights as observations getting older. The most commonly used single exponential smoothing is given by

$$F_0 = x_0, \tag{2.2}$$

$$F_{t+1} = \alpha x_t + (1-\alpha) F_t = F_t + \alpha(x_t - F_t), \tag{2.3}$$

where $\alpha \in [0,1]$ is the smoothing factor, F_{t+1} is the new forecast for next period, x_t is the current observation at period t, and F_t is the last forecast made in period $t-1$. In the above formula, one can substitute $F_t = \alpha x_{t-1} + (1-\alpha) F_{t-1}$, and continue so forth to obtain the infinite expansion of F_t as follows

$$F_t = \sum_{i=0}^{\infty} \alpha(1-\alpha)^i x_{t-i-1} = \sum_{i=0}^{\infty} \alpha_i x_{t-i-1}, \tag{2.4}$$

where $\alpha_i = \alpha(1-\alpha)^i$. From this expression, one can see clearly that the weight α_i decreases exponentially with time. This illustrates why this method is called 'exponential smoothing'. Single exponential smoothing works best only for stationary time series data. Double exponential smoothing has been developed to handle time series

data with linear trends. And triple exponential smoothing has been proposed to deal with both trend and seasonality. A very detailed discussion of exponential smoothing techniques can be found in [49, 50].

2.1.3 Box-Jenkins Methods

Box-Jenkins methods, named after the statisticians George Box and Gwilym Jenkins, who applied autoregressive moving average (ARMA) to make forecasts for time series data [15]. An ARMA model can be generally described by

$$x_t = c + \epsilon_t + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_p x_{t-p}, -b_1 \epsilon_{t-1} - b_2 \epsilon_{t-2} - \ldots + b_q \epsilon_{t-q} \quad (2.5)$$

$$= c + \sum_{i=1}^{p} a_i x_i - \sum_{j=1}^{q} a_j \epsilon_j, \quad (2.6)$$

where x_t is the current observation, $x_{t-1}, \ldots, x_{t-p}$ are the observations in the past p periods, and the $a_1, \ldots, a_p$ are the regression coefficients of the past p observations. The ϵ_t is the current prediction error, the $\epsilon_{t-1}, \epsilon_{t-2}, \ldots, \epsilon_{t-q}$ are the past q prediction errors, and the $b_0, b_1, \ldots, b_q$ are the associated regression coefficients. ARMA model assumes that the time series to be analyzed is stationary. To handle the nonstationarities such as trend and seasonality, Box and Jenkins proposed a differencing version of ARMA model, which is known as ARIMA model. The 'I' stands for 'Integrated', since the estimation process is performed on differenced data, and the time series needs to be integrated before making a forecast. More mathematical details of Box-Jenkins model can be referred to Box et al. [15].

2.1.4 State Space Models

State space models virtually build up a generalized representation of linear time series models in state space form. For example, one can represent an ARIMA model in state space form. Once a state space model is built, it can be conveniently analyzed by Kalman filter and the associated smoother. Kalman filter is a recursive procedure to compute the optimal estimator of the state vector at time t, based on the information

available at time t. The parameters of a state space model are usually estimated by maximum likelihood functions. State space method is a sophisticated form of forecasting models. A detailed discussion of various algorithms of state space models can be found in Harvey [59].

2.1.5 Spectral Analysis

Spectral analysis represents a group of methods which decompose time series data into a few underlying sine and cosine functions of different frequencies. Compared to ARIMA or Exponential Smoothing techniques, for which seasonal period is known as a priori in the analysis, spectrum analysis is suitable to deal with the seasonal series data for which lengths of cyclic patterns or fluctuations are changing rapidly or difficult to estimate. In spectral analysis, some important recurring cycles of different frequencies in the time series can be discovered. Those patterns may be hidden in random noises and are extremely difficult to find out by other methods. The most common spectrum decomposition process is also referred as Fourier analysis, which can be considered as a linear multiple regression process. The dependent variable is the time series to be studied, and the independent variables are sine and cosine functions of all possible frequencies. In general, a spectral decomposition model is give by

$$x_t = c + \sum_{j=1}^{k} (a_k cos(\omega_j t) + b_k sin(\omega_j t)), \quad 0 < \omega_1 < \cdots < \omega_k < \pi \tag{2.7}$$

where $\omega_1 < \cdots < \omega_k$ are k possible wave lengths of the cyclic patterns in the time series, $a_1, a_2, \ldots, a_k$, and $b_1, b_2, \ldots, b_k$ are regression coefficients that represent the degree of corresponding sinusoidal functions are correlated with the data. In other words, a large sine or cosine coefficient indicates a strong periodicity of the respective frequency in the data. There are various techniques available to perform spectral analysis for time series data. A comprehensive discussion of spectral analysis can be found in Koopmans [80].

2.2 Casual Prediction Models

Causal models are also known as cause-and-effect models, which establish a causal relationship between the variable being forecasted and all other related variables. The most well known causal models are called regression models, which build up a rigorous mathematical model of causal relationship based on sound statistical techniques. In a regression model, the variable to be forecasted is called dependent or response variable, and the variables that represent the causal factors of the dependent variable are called independent or explanatory variables.

2.2.1 Regression Models

Regression models are in principle to investigate the relationship between one dependent variable and its relevant independent predictor variables. In general, a standard regression model takes the form as follows

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n, \tag{2.8}$$

where Y is the forecasted value of the dependent variable, b_0 is the intercept, and $b_1, b_2, \ldots, b_n$ are the estimated regression coefficients representing the contribution of the independent predictor variables $X_1, X_2, ..., X_n$, respectively. When linear regression models do not appear to adequately capture the relationships between dependent and predictor variables, nonlinear regression models (such as polynomial regression) can be used. An overview of the popular nonlinear regression models can be found in Seber and Wild [141].

2.2.2 Econometrics Models

In many real problems, the cause-and-effect relationship between dependent and independent variables are not straightforward. The estimated model parameters by the standard regression analysis may become inappropriate due to the highly dynamic relationship between the dependent and independent variables. For example, we wish to

forecast sales of a product which are related to its price. However, the market price in turn is also affected by sales. Another typical example is the supply and demand model. The interaction of supply and demand jointly determines the equilibrium price and quantity of the product in the market. In such cases, it no longer makes sense to separate dependent and independent variables completely. To handle this problem, a set of simultaneous regression models is necessary to describe dynamics of these systems. The simultaneous regression models are called econometrics models in literature, since they are often applied to analyze the relationships between economic variables that should be jointly determined. One can consider that a single regression model is a special case of econometrics models. The rigorous mathematical formulations of econometrics can be found in Pindyck and Rubinfeld [118].

2.2.3 ANN Models

Artificial neural networks (ANNs) represent another important form of causal models, which have shown powerful capabilities of modeling complex relationships between inputs and outputs. An ANN model consists of a network of neurons connected by arcs with assigned weights. Neurons take some form of basic nonlinear functions. Therefore, an ANN model can be equivalently considered as a nonlinear regression model in mathematics. A typical ANN has three layers, an input layer, a hidden layer, and an output layer. As a causal model, the inputs to an ANN are independent or explanatory variables, and the outputs are dependent or response variables being forecasted. There are various algorithms available to train ANNs, such as Perceptron learning rule and backpropagation [134]. Once the structure and weights of an ANN is determined, it can be employed to perform forecasting. ANNs have been increasingly used in forecast modeling in the past decade. They are suitable for complicated problems which are difficult to be mathematically formulated by regression models or econometric models. In many real applications, ANN methods can often achieve good performance if given enough training data. An overview of the applications of ANNs in forecasting can be found in Zhang et al. [164].

2.3 Comparing Forecasting Models

We have discussed the most popular prediction models above. To evaluate prediction models, two aspects of terms are often concerned about: accuracy and bias. Accuracy refers to the distance between the forecasts and actual values. And a forecast is biased if the errors in one direction are significantly larger than those in other directions. In general, the basic objective of all forecast models is to maximize accuracy and minimize bias. A number of criteria have been proposed to compare prediction models, which will be discussed in the following.

2.3.1 Prediction Error Measures

To achieve high prediction accuracy is the primary objective in most forecasting tasks. To evaluate forecasting accuracy, four of the more popular direct error measures are mean squared error (MSE), or its variants such as root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Minimizing these measures is usually the most essential criterion in comparing forecasting models. These measures are most frequently used due to their mathematical convenience. For each of these measures, a smaller value indicates higher prediction accuracy. Given a set of real data $y_i, i = 1, 2, \ldots, n$, each of which has an associated forecast value $\hat{y}_i$, then these measures are defined as follows

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}, \tag{2.10}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2.11}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}.| \tag{2.12}$$

The R-squared (R^2), also known as coefficient of determination, is another most commonly used criterion to evaluate a forecast model. The most general form of the

R^2 is defined as follows

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} = 1 - \frac{SS_{err}}{SS_{tot}}, \tag{2.13}$$

where $\bar{y} = \sum_{i=1}^{n} y_i/n$, it is the mean of the observed data. The second term compares the variance of the forecast errors with the total variance of the data. One minus the second term is the proportion of variability in a data set that can be explained by the forecast model. R^2 is used to measure how well a model approximates real data values. The magnitude of R^2 is usually restricted within 0 and 1. An R^2 close to 1.0 indicates that the model perfectly fits the data, while an R^2 close to 0 means that the model cannot explain the data at all.

2.3.2 Information Criterion

A number of model selection criteria have also been developed based on information theory. A well-known criterion is Akaike's information criterion (AIC) developed by Akaike in 1974 [4]. It makes a tradeoff between accuracy and complexity in model construction. In general, this criterion can be defined as:

$$AIC(K) = log(MSE) + \frac{2K}{n}, \tag{2.14}$$

where K is the number of parameters in the model, n is the number of observations in the data. The MSE has been defined above, and it can be explained as the estimated residual variance in this criterion. In the formula of AIC, the first term indicates model accuracy, and the second term indicates model complexity in terms of the number of parameters. Hence AIC not only rewards prediction accuracy, but also gives a penalty to larger number of model parameters. One major benefit of this penalty is to discourage overfitting. For a set of models, the one with the lowest AIC value is considered as the preferred model. This criterion is particular suitable for comparing a set of nested models. For example, compare an AR(m) model with an AR(m+1) for a given set

of data. One drawback of AIC is that it is not consistent, since as the number of observations grows, the probability of selecting the correct model does not approach one.

The Bayesian information criterion (BIC), also known as Schwarz criterion, is another well-known criterion to select a set of parametric models with different choices of explanatory parameters [140]. BIC is actually a variant of AIC in a form of:

$$BIC(K) = log(MSE) + \frac{log(n)K}{n}. \tag{2.15}$$

The BIC also makes a tradeoff between accuracy and complexity of a model. For a set of models, the one with the lowest value of BIC is the one to be preferred. It differs from AIC in that the penalty coefficient of K becomes $log(n)/n$ instead of $2/n$. The BIC generally penalizes free parameters more strongly than does the AIC. In addition, Hannan and Quinn [58] also proposed an alternative to AIC and BIC called Hannan-Quinn criterion (HQC), which is given by

$$HQC(K) = log(MSE) + \frac{2Klnln(n)}{n}. \tag{2.16}$$

Similar to AIC and BIC, the model with the lower value of HQC is preferred. It has been shown that consistency can be obtained by the BIC [140] or HQC [58, 57].

2.3.3 Cross-Validation

Cross-validation is also a commonly used technique to compare different predictive models in practice [52]. Given a set of data, the basic idea of cross-validation is to partition the data set into training and validating subsets, and estimate the predictive accuracy on the validating data by the model obtained from the training data set. The MSE, RMSE, MAE, and MAPE can be used to measure the expected level of fit of a predictive model. If a model fits the training data set very well but does not fit the validation data, it is called overfitting. A good predictive model is supposed

to generate consistent results in both training and validating data sets. To reduce the variability of the model evalutation, multiple trials of cross-validation are usually performed with respect to different partitions to the original data set. The evaluation result of a predictive model is the averaged result over these trials. There are several different approaches to perform multiple steps of cross-validation.

- Repeated Random Partition Validation: this method simply divides the dataset into two subsets randomly each time and repeats the same procedure a number of times. One problem of this method is that the validation subsets may overlap and some observations may never be selected in the validation subsets.

- K-fold cross-validation: the dataset is partitioned randomly into K subsets. Then the cross-validation is repeated K times. Each time one of the subsets is reserved as the validation data, and the remaining $K-1$ subsets are the training data sets. The validation result is the average over K results. This approach guarantees that each observation can be used for validation exactly once.

- Leave-one-out cross-validation: this method is actually a special case of K-fold cross-validation, when the number of observations in each subset is one. In other words, only one observation is reserved for validation and the remaining observations are used for training. The procedure is repeated until each observation has been used once for validation.

2.3.4 Stepwise Model Selection

Stepwise model selection approaches are very useful for automatically selecting a set of nested predictive models, for which there are a large number of potential predictive variables [32]. The selection procedure is generally grounded in some statistical tests and usually takes in the form of partial F-test. Other measures can also be used, such as t-tests, R^2, AIC, and BIC. Since the basic procedures are similar, only the case of partial F-test is discussed here. To compare two nested models with different number

of predictor variables, the partial F-test can be generally formulated as follows:

$$F = \frac{\text{Extra sum of squares/Extra model df}}{\text{SSR of large model/Residual df Large model}}, \tag{2.17}$$

where SSE denotes the sum of squared residual. The key idea of this test is to check if the 'extra' predictors of the large model explain significantly more of the variability compared to the variability that is explained by the predictors that are already in the small model. Based on partial F-test, three approaches are commonly used for model selection:

- Forward selection: starts with the smallest number of possible predictors and adds predictors one by one until a stop criterion is satisfied or the largest model is reached. The current model satisfying the stop criterion is selected. In particular, suppose that the current model has P parameters, and we want to test if one of the model with $P+1$ parameters is more preferred. If for all models of $P+1$ parameters, it satisfies

$$F = \frac{SSE(P+1) - SSE(P)}{SSE(P+1)/n - P - 1} < F_{m,P}, \tag{2.18}$$

  where $F_{m,P}$ is the critical values of F-statistic for a chosen level of significance. Then stop the process and the current model with P parameters is preferred. Otherwise, select one preferred model of $P+1$ parameters, and repeat the test for all models with $P+2$ parameters, and so forth.

- Backward selection: starts with the largest number of possible predictors and removes predictors one by one. At each step, it compares all the smaller model candidates with the old larger model and stops the process if

$$F = \frac{SSE(P-1) - SSE(P)}{SSE(P)/n - P} < F_{m,P}. \tag{2.19}$$

- Stepwise selection: a modified version of forward-selection which allows the elimination of predictors those become statistically insignificant in the model. At each step of the process, the p-values of all predictors are computed. If the largest of these p-values is greater than a critical value, then the corresponding predictor is eliminated. Other steps are all the same with those of forward selection.

2.3.5 Residual Diagnostics

Residuals represent the portion of the validation data not explained by the model. The graphical residual analysis is commonly used in complementary with the quantitative techniques. A typical residual diagnostics includes plots of residuals versus the predicted values, versus other predictors, and versus time, residual autocorrelation plots, residual histogram, and normal probability plots. In general, the residual analysis can be used to test the following:

- Whiteness test: a good predictive model should have the uncorrelated residuals.
- Independence test: a good model should have residuals uncorrelated with past inputs.
- If there are some extreme influential observations. Identifying and deleting outliers from the training dataset may significantly improve the quality of a model.
- If the residuals exhibit systematic patterns and bias. The residuals of a good model should be approximately dispersed around zero evenly. If systematic patterns are found, the most probably reason is that one or several relevant predictors are missing. A forecast is biased if residuals in one direction are significantly larger than those of the other direction.

2.4 Overview of Machine Learning Techniques

With the explosion of computing power in the past decade, machine learning and pattern recognition techniques have become important tools in the analysis of various

biological problems, such as in cancer research [93], cognitive neuroscience [29], and genomics and proteomics [26]. Machine learning best depicts the computational methods that allow a system to evolve behaviors through an automated process of knowledge acquisition from empirical data. Machine learning techniques generally fall into three broad categories: supervised learning, reinforcement learning and unsupervised learning. A supervised learning technique usually first finds a mapping between inputs and outputs of a training dataset, and then makes predictions to the inputs that it has never seen. A large number of supervised learning algorithms have been developed, which can be categorized into several major groups including neural networks, support vector machines, locally weighted learning, decision trees, and Bayesian inference [83]. Reinforcement learning is another learning paradigm in which an agent is able to learn a decision policy by 'trial and error'. A reinforcement learner receives feedback of its actions and makes adjustments to its actions accordingly [154]. Reinforcement learning is a natural framework for building models to accumulate knowledge from previously learned tasks to new tasks with increasing complexity and variability. Reinforcement learning techniques have been applied to many complex learning tasks, such as robot control [33] and traffic network control[132]. Unsupervised learning is inspired by the brain's ability to recognize complex patterns of visual scenes, sounds or odors. It takes root in neuroscience/psychology and is established on information theory and statistics. An unsupervised learner usually performs clustering or associative rule learning to extract the implicit structure of a given dataset. The established clusters, categories or associative networks are then used for decision making, prediction, or efficient communication [31].

Modern devices can produce voluminous datasets fueling a need for effective and user-friendly data mining models. At present, machine learning techniques have found widespread applications in many real world complex problems, which are difficult to deal with using traditional modeling tools. Bhaskar et al. [12] reported that the number of studies in the area of machine learning for solving problems in bioinformatics has rapidly increased since 1999. Some examples include the use of neural networks and SVMs to classify and diagnose cancers using gene expression data [78, 87], data clustering

for the analysis of breast cancer and colon cancer [53], and reinforcement learning to individualize erythropoietin dosages for hemodialysis patients [100].

2.5 Data Mining for Non-stationary Chaotic Time Series Prediction

Data mining is the search for valuable information in large volumes of data. Predictive data mining is a search for very strong patterns in big data that can generalize to accurate future decisions. The most interesting time series presented in this dissertation may be classified as non-stationary deterministic chaotic time series. A working definition of a chaotic time series is one generated by a nonlinear, deterministic process highly sensitive to initial conditions that has a broadband frequency spectrum [1]. Since the chaotic time series is deterministic, it is still predictable. However, the time series patterns are extremely complex, and the prediction horizon is unknown.

Many researchers have applied data mining concepts to finding predictive patterns in time series include Berndt and Clifford [11], Keogh [75, 73, 77]. Berndt and Clifford use a dynamic time warping technique taken from speech recognition. Their approach uses a dynamic programming method for aligning the time series and a predefined set of templates. Keogh represents the templates using piecewise linear segmentations. Local features such as peaks, troughs, and plateaus are defined using a prior distribution on expected deformations from a basic template. A probabilistic method was used for matching the known templates to the time series data.

The adaptive learning framework, introduced in this dissertation, differs fundamentally from these approaches. In particular, most prediction approaches, such those advanced in [75, 11, 73, 77], require a priori knowledge of the types of structures or temporal patterns to be discovered and represents these temporal patterns as a set of templates. The use of predefined templates completely prevents the achievement of the basic data mining goal of discovering useful, novel, and hidden temporal patterns from massive chaotic time series data.

In this dissertation research, we performed a number of study on EEG data, which is an important form of chaotic multivariate time series. In the next chapter, we made

an extensive study on EEG time series, and applied the existing data mining techniques to identify if some predictive time series patterns can be found from EEG data. We demonstrated that EEG signal may be predictable based on this preliminary study. Thus EEG data can be a good candidate to test and evaluate our proposed online monitoring and prediction frameworks discussed in Chapter 4 and Chapter 6.

Chapter 3

An Extensive Study of EEG Time Series for Early Detection of Numerical Typing Errors

This chapter studies the possibility of early detection of numerical typing errors from EEG recordings. The objective of this study is to perform various data mining techniques on EEG time series, and evaluate if EEG time series can be employed to predict a future event of the brain. Three feature extraction techniques were developed to capture temporal, morphological and time-frequency (wavelet) characteristics of EEG data. Two most commonly used data mining techniques, Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) were employed to classify EEG samples associated with correct and erroneous keystrokes. The leave-one-error-pattern-out and the leave-one-subject-out cross validation methods were designed to evaluate the in-subject and cross-subject classification performance, respectively. For in-subject classification, the best testing performance had a sensitivity of 62.20% and a specificity of 51.68%, which were achieved by SVM using morphological features. For cross-subject classification, the best testing performance was achieved by LDA using temporal features, based on which it had a sensitivity of 68.72% and a specificity of 49.45%. In addition, the Receiver Operating Characteristic (ROC) analysis revealed that the averaged values of the area under ROC curves (AUC) of LDA and SVM for in-subject and cross-subject classification were both greater than 0.60 using the EEG 300ms prior to the keystrokes. The classification results of this study indicated that the EEG patterns prior to erroneous keystrokes might be different from those of correct ones. The outcome of this study indicates that the EEG time series may show predictive patterns prior to events, based on which one can develop develop online monitoring and prediction systems based on EEG signals.

3.1 Introduction

Numerous types of electronic devices with alphabetical or numerical keyboards have become very important tools in modern times. An erroneous keystroke can be easily caused by many reasons such as operators' inexperience, fatigue, and carelessness. At the present time, many typing error correction systems have been developed for computer users. For example, current word processing software such as Microsoft Word provides automatic spelling checks as well as automated corrective actions. Some other methods have also been developed to detect and remove errors due to overlapped keystrokes [155]. It is noted that most of the automatic typing error detection systems are designed for text typing; very few studies have focused on detecting numerical typing errors. In fact, numerical typing is as a common task as text typing in practice [138, 139]. In particular, in some crucial tasks, numerical typing errors may result in serious consequences or accidents. For instance, numerical typing errors in medical records may result in inaccurate diagnoses and/or drug administrations. In financial transactions, numerical errors may cause significant losses at the stock exchanges. In aviation control, incorrect numerical inputs may lead to serious air traffic accidents [163].

Human typing involves intricate interactions of concurrent perceptual and cognitive processes [137]. Numerous studies of typing behaviors have been conducted to explore their underlying cognitive mechanisms in the past decade. However, most studies in the literature were in the field of transcription (text) typing. Error correction of numerical data is much more difficult and challenging because there is no pattern database to look-up. The operator cannot visualize and identify if there are errors in the data because there is no contextual information for verification. This is very common in hear-and-type tasks such as telling a phone number to a phone representative, bank account number to a teller, and tracking number of a parcel to a customer service agent. The operators are very susceptible to making errors when they receive auditory inputs while typing.

Although double data entry (DDE) and read-aloud (RA) [72] methods are commonly

used to assure the data quality, these methods are very tedious and inefficient. Also, in reality to avoid typing errors, each data entry may accompany a confirmative action, such as pressing an ENTER key, before which input data can be checked and corrected. However, such a mechanism does not always exist. For example, a selection menu may be coded by numbers and pressing a number already commands the execution. In this kind of situations, afterward detection mechanism is too late to reverse the outcome. Especially in a crucial task, such numerical errors may result in serious or even life threatening consequences. As a result, when afterward checking/confirmation is limited or even impossible, predicting and avoiding numerical typing errors are becoming very critical to assure data quality. Unfortunately, to the best of our knowledge, there are no effective tools available to assist human operators in this task. If numerical typing errors can be detected in advance, the detection can be integrated in an error prevention system for many crucial typing works.

Erroneous keystrokes are possibly caused by an operator's psychophysiological state such as a lack of attention, external distractions and fatigue. In our previous study, we have successfully built a computational model, called the Queuing-Network-Model Human Processor, to establish mathematical representations of cognitive functions of typing behaviors. The results of our study on brain modeling and human factor analysis of erroneous keystrokes suggested that the brain activity before erroneous keystrokes might be different from that of correct ones [161, 94]. The goal of this study is to develop an early detection system of erroneous keystrokes. We employ feature extraction and data mining techniques to perform quantitative analysis of EEG recordings prior to keystrokes. Although there have been numerous EEG studies in various fields, very few studies in the literature have been conducted to investigate the early detection (or prediction) of typing behaviors based on EEG data. The characterization of the underlying EEG patterns before someone is about to make an error is still in a great need of further investigation. The development of an effective method to classify erroneous keystrokes based on their (generating) mechanisms remains a difficult but worth-pursuing task.

The rest of this chapter is organized as follows. In Section 3.2, the research background and previous related work are discussed, including the background of data entry correction methods, error related EEG potentials, and the data mining techniques for quantitative EEG analysis. In Section 3.3, the proposed EEG feature extraction techniques and the employed classification techniques, linear discriminant analysis (LDA) and support vector machine (SVM), are described. Section 3.4 presents the design of human experiments and computational data analysis. The computational results of the classification systems are provided in Section 3.5. Finally, the concluding remarks and future work are given in Section 3.6.

3.2 Background

3.2.1 Data Entry Correction Methods

A number of studies have been performed to develop correction methods for numerical typing errors. Scholtus [138, 139] developed an algorithm for automatic correction of typing errors in numerical data. However, this algorithm can be only applied to some systematic typing errors, such as checking the inconsistencies when there are mathematical relations between the data digits. Kawado et al. [72] compared the efficiency of the two commonly used data verification methods: DDE and RA. In the DDE method, the double data entry was performed by either an identical or a different operator. In the RA method, one operator read the typed data on a printed sheet or computer screen aloud, and another operator compared the data (that were) heard with the data (that were) recorded to confirm whether they were the same. The error detection rate was 59.5% for the RA method, and 69.0% for the DDE method. Their results surprisingly showed that there might still be a large portion of undetected errors even after the two commonly used data verification methods were applied. Their study also indicated that it is very hard to achieve full accuracy for large amount of numerical data input even when data verification methods are used to the data management. As mentioned in Arndt et. al. [8], databases of large projects may contain a great absolute number of mistakes in data collection, and thus have data quality problems. They

investigated the types and frequencies of data errors in 688 forms from seven sites in a multicenter field trial. It was found that 2.4% of the received data had errors even though conscientious efforts had been made in checking and correcting data.

3.2.2 Error-Related Potentials (ErrP)

Typing behaviors involve complex interactions of concurrent perceptual and cognitive processes [137]. The brainwave activity measured by EEG is often an essential and natural way to study the brain activity during typing. The event-related potentials (ERP) in response to a perceptual, cognitive or motor event have been extensively studied in neuroscience and brain computer interfaces [56]. Since the early 1990s, many studies have found that a subject's recognition of response errors is often associated with some specific error-related EEG potentials [38, 108]. More recently, the work of Ferrez and Millán [124] showed that ErrP of a brain-computer-interface (BCI) can be reliably recognized. The pioneering work in ErrP detection provided a prelude for us to explore the underlying mechanisms of erroneous keystrokes. It should be noted that current ErrP studies mostly focused on the EEGs after response errors, and the EEGs prior to errors were much less studied. However, EEGs prior to errors are of great importance to prevent errors from occurring, especially in some crucial typing tasks mentioned in the introduction part. Therefore, this study particularly focused on early error detection using EEGs prior to keystrokes.

3.2.3 Data Mining in EEG: Feature Extraction

Over the past decade, numerous studies have been performed to apply quantitative signal processing methods and time series techniques to analyze characteristics of EEG data. The simplest feature extraction can be obtained by downsampling an EEG signal from its usually high sampling rate (such as 1000Hz) into a low frequency range of particular interest (such as 0-30Hz). The resulting features are supposed to be representative to the temporal characteristics of EEG data in this low frequency band [13]. Another common univariate feature extraction method uses the morphological

characteristics of EEG data, such as curve length [110], zero crossings [130], number of peaks [160], nonlinear energy [3], etc. In addition, grounded in signal processing techniques, some more complex EEG feature extraction techniques have also been developed. Traditional linear methods include frequency and power spectrum analysis and the parametric modeling of EEG time series (e.g., autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models). Though widely used in EEG analysis, these methods actually treat EEG as statistically stationary signals. To deal with nonstationarity in EEG, various methods based on time-frequency analysis have been developed. The most well-known time-frequency technique is called wavelet transform, which is capable of providing a representation of nonstationary EEG signals in both time and frequency domain accurately.

3.2.4 Data Mining in EEG: Classification

Over the past decade, there were increasing interests in using classification techniques to discriminate different brain activities based on EEG recordings. Numerous data mining techniques have been proposed to EEG classification. Those methods include decision trees [120], neural networks [149], association rule induction [37], K-Nearest-Neighbor method [23], and genetic algorithms [109]. There have been many studies suggesting that EEG signals at different mental states or in different mental tasks may be classifiable [7, 6, 10, 107, 145].

The error detection task in this study is in principle a binary classification problem of correct and erroneous EEG samples. LDA and SVM are two popular classification techniques for binary classification tasks. Both of them construct a hyperplane to separate data into two subsets based on optimization theories. Parra et al. [115] used LDA to detect response errors for seven subjects in a forced choice visual discrimination task. Using 64 EEG electrodes and two time windows of 100 ms, they were able to reach an accuracy of 79 % on average. Blankertz et al. [13] adopted Sparse Fisher Discriminant (SFD) to differentiate index finger movement from small finger movement in a self-paced key typing task. Using EEG data 120ms prior to keystrokes, they achieved overall classification accuracies of 96.7% and 93.6% for filtered and non-filtered

EEG data, respectively. SVMs have also been widely applied to a large number of EEG classification problems [47, 162]. Our group successfully applied data mining techniques to classify normal and abnormal (epileptic) brain activities based on EEG recordings of a number of epileptic patients [23, 18, 20]. Garrett et al. [51] applied both LDA and SVM to classify EEG signals in five mental tasks. The averaged classification accuracies of LDA and SVM were 66% and 72%, respectively.

3.3 Methods

3.3.1 Feature Extraction

We employed three feature extraction techniques to capture characteristics of EEG signals. They were temporal, morphological and wavelet features. For an EEG epoch with n channels, we first extracted features from each channel, and then concatenated the features of all the n channels to construct the feature vector of this multichannel EEG epoch. Let $X = \{x_1, x_2, , \cdots, x_m\}$ denote a single-channel EEG with m sampling points, the extraction of the temporal, morphological and wavelet features of X are described as follows.

3.3.1.1 Temporal Features

the temporal features can be obtained by downsampling of EEG signals. The downsampling of EEG data reduces the amount of data that needs to be analyzed while it is still capable of capturing patterns for slow brain activity [13]. Since the most common EEG patterns (e.g., alpha, beta, delta, theta wave patterns) contain frequency elements mainly below 30Hz, we downsampled the EEG data from 1000Hz to 30Hz in this study. In particular, the downsampling was accomplished by calculating means of consecutive, non-overlapping intervals of every 33 points. For example, a 100ms EEG epoch of 1000Hz has 100 points in each channel, then three temporal features are extracted for each channel of EEG through the downsampling process.

3.3.1.2 Morphological Features

seven morphological features were extracted from each channel of EEG. These features were based on the features previously described in Wong et al. [160]. A brief description of the morphological features is given in the following.

- Curve Length: this feature is also known as 'line length' which was first proposed by Olsen et al. [110]. Curve length is the sum of distances between successive points, given by

$$\sum_{i=1}^{m-1} |x_{i+1} - x_i|. \tag{3.1}$$

  Since curve length increases as the signal magnitude or frequency increases, it can be used to measure the amplitude-frequency variations of the EEG signals. It has been used in many EEG studies, such as epileptic seizure detection [35], stimulation responses of the brain [36].

- Standard Deviation: it is among the most widely used measures of signal variability. It indicates how all the points of the signal are clustered around the mean. The standard deviation can be obtained by

$$\sqrt{\frac{\sum_{i=1}^{m}(x_i - \bar{X})^2}{m-1}}, \tag{3.2}$$

  where $\bar{X}$ is the mean of the single-channel EEG X.

- Number of Peaks: the number of peaks per second is a commonly used characteristic to measure the overall frequency of EEG signals. The number of peaks in the single-channel EEG X can be calculated by

$$\frac{1}{2}\sum_{i=1}^{m-2} max\{0, sgn(x_{i+2} - x_{i+1}) - sgn(x_{i+1} - x_i)\}. \tag{3.3}$$

- Root Mean Square (RMS) Amplitude: RMS is one of the most commonly used methods to determine the power changes of the signal [103], especially for complex waveforms, such as EEG signals. The RMS amplitude of the single-channel EEG

X is defined as

$$\sqrt{\frac{\sum_1^m x_i^2}{m}}. \tag{3.4}$$

- Average Nonlinear Energy: nonlinear energy was first proposed by Kaiser [70]. It is a measure of signal energy that is proportional to both signal amplitude and frequency. It has been found that the nonlinear energy is sensitive to spectral changes. Thus it is also useful to capture spectral information of an EEG signal [3]. The average nonlinear energy of the single-channel EEG X is computed as

$$\frac{1}{m-2}\sum_{i=2}^{m-1} x_i^2 - x_{i-1}x_{i+1}. \tag{3.5}$$

- Zero Crossings: the frequency information of EEG signals can also be estimated by the number of times its value crosses the zero axis. Zero-crossing feature extraction has been applied in many signal processing and pattern recognition tasks including EEG signal analysis [130]. The zero crossings of the single-channel EEG X can be mathematically defined as

$$\frac{1}{2}\sum_{i=1}^{m-1} |sgn(x_{i+1}) - sgn(x_i)|. \tag{3.6}$$

- Variance-to-Range Ratio: this feature calculates the ratio of the variance to the magnitude range of the EEG signal. It takes into account both variation and range of EEG magnitudes. The ratio of the single-channel EEG X is given by

$$\frac{\sum_{i=1}^m (x_i - \bar{X})^2}{(m-1)(X_{max} - X_{min})}, \tag{3.7}$$

  where X_{max} and X_{min} are the maximum and minimum value of X, respectively.

3.3.1.3 Time-Frequency Features

Wavelet transform (WT) was employed to analyze time-frequency characteristics of the EEG signals. The basic idea of wavelet analysis is to express a signal as a linear

combination of a particular set of functions obtained by shifting and dilating one single function called mother wavelet. The WT of the signal $X(t)$ is defined as

$$C(a,b) = \int_R X(t) \frac{1}{\sqrt{a}} \Psi(\frac{t-b}{a}) dt \tag{3.8}$$

where Ψ is the mother wavelet, $C(a,b)$ are the WT coefficients of the signal $X(t)$, a is the scale parameter, and b is the shifting parameter. Continuous wavelet transform (CWT) has $a \in R^+$ and $b \in R$; and discrete wavelet transform (DWT) has $a = 2^j$ and $b = k2^j$ for all $(j,k) \in Z$ given the decomposition level of j. Analyzing the signal by CWT at every possible scale a and shifting b requires substantially more computations than the DWT. As a result, the DWT with dyadic scaling and shifting is often employed in many studies to decompose EEG signals into different frequency sub-bands [133]. The coefficients of DWT decomposition provide a non-redundant and highly efficient representation of a signal in both time and frequency domain. At each level of decomposition, DWT works as filters to divide the signal into two bands called approximations and details signals. The approximations (A) are the low frequency components of the signal, and the details (D) are the high-frequency components. For more detailed mathematical formulations of wavelet transform can be referred to Addison [2].

Among different wavelet families, Daubechies wavelets are well known for its orthogonality property and efficient filter implementation, and the db4 is frequently used in EEG analysis [151]. In this study, we applied the typical db4 to decompose EEG signals into eight levels. Table 3.1 shows the frequency bands of different levels of DWT decomposition. Since the frequency band of EEG signals is often considered to be less than 30 Hz, we employed the coefficients of the level A7, D7, D6, D5, which roughly correspond to the commonly recognized delta, theta, alpha, and beta brainwaves, respectively. The other four levels of signals were considered as the high-frequency background noises, and thus were eliminated in the wavelet feature vector. Moreover, to further decrease the feature dimensionality for classification, the statistics of the DWT coefficients were extracted. They are mean, standard deviation, maximum and minimum of the wavelet coefficients of the four used levels. By doing so, each channel of EEG can be represented

by a $4 \times 4 = 16$ dimensional feature vector, and an EEG epoch of n channels can be represented by a $16n$-dimensional feature vector.

Table 3.1: Frequency ranges and the corresponding brainwave bands of the eight levels of signals by discrete wavelet decomposition.

Decomposed Signal	Frequency Range (Hz)	Approximate Band
D1	250-500	-
D2	125-250	-
D3	62.5-125	-
D4	31.3-62.5	-
D5	15.7-31.3	Beta
D6	7.9-15.7	Alpha
D7	4.0-7.9	Theta
A7	0-4.0	Delta

3.3.2 Classification Methods

Let Y denote the $n \times k$ dimensional feature vector for a multi-channel EEG epoch, where n is the number of channels and k is the number of features of each single channel of EEG. In this study, $n = 36$ and $k = 3, 7, 16$ for temporal, morphological and wavelet features, respectively. Let l denote the class label of the EEG epoch, for which $l = 1$ denotes a correct EEG sample, and $l = -1$ means an erroneous EEG sample. Given $p + q$ training samples (Y_i, l_i), $i = 1, ..., p + q$, the dataset of p correct EEG epochs is denoted by $D_1 = \{(Y_1, l_1), (Y_2, l_2), ..., (Y_p, l_p)\}$, and the dataset of q erroneous epochs is denoted by $D_2 = \{(Y_{p+1}, l_{p+1}), (Y_{p+2}, l_{p+2}), ..., (Y_{p+q}, l_{p+q})\}$. The difference between them is that the optimal decision boundary is determined based on different optimization theories, which will be briefly discussed in the following.

3.3.2.1 Fisher's Linear Discriminant Analysis

Fisher's LDA aims to find an optimal projection by minimizing the intraclass variance and maximizing the distance between the two classes simultaneously [46]. Mathematically, LDA tries to find an optimal direction $\omega^* \in R^{n \times k}$ as a solution of the following

optimization problem:

$$\omega^* = argmax_\omega \frac{\omega^T S_b \omega}{\omega^T S_{\omega^*} \omega}, \tag{3.9}$$

where ω is the direction of the hyperplane that is used to separate the two data sets. S_b and S_ω are the interclass and intraclass covariance matrix, respectively. They are defined as follows

$$S_b = (m_1 - m_2)^T (m_1 - m_2), \tag{3.10}$$

$$S_\omega = \sum_{i \in 1,2} \sum_{i \in D_i} (Y_i - m_i)^T (Y_i - m_i), \tag{3.11}$$

where m_1 and m_2 are the means of the feature vectors Y in the two data sets D_1 and D_2, respectively. They can be calculated by

$$m_1 = \frac{1}{p} \sum_{Y \in D_1} Y = \frac{1}{p} \sum_{i=1}^{p} Y_i, \tag{3.12}$$

$$m_2 = \frac{1}{q} \sum_{Y \in D_2} Y = \frac{1}{p} \sum_{i=p+q}^{p+1} Y_i. \tag{3.13}$$

When S_ω is not singular, the above optimization problem can be solved by applying the eigen-decomposition to the matrix $S_\omega^{-1} S_b$. The eigenvector corresponding to the largest eigenvalue forms the optimal direction w^* by

$$\omega^* = S_\omega^{-1} (m_1 - m_2). \tag{3.14}$$

When S_ω is singular, an identity matrix with a small scalar multiple can be used to tackle this problem [102]. The optimal w^* then becomes

$$\omega^* = (S_\omega + \lambda I)^{-1} (m_1 - m_2). \tag{3.15}$$

Once ω^* is obtained, the optimal decision boundary of LDA can be represented by

$$\omega^{*T} Y + b = 0, \tag{3.16}$$

where b is the bias term. There is no general rule to determine the bias term, a most commonly used bias term is $b = -\omega^{*T}(m_1 + m_2)/2$. The class of an EEG epoch Y depends on which side of the hyperplane its feature vector is on. In particular, for a new EEG epoch represented by a feature vector Y_{new}, then the prediction rule is as follows

$$\begin{cases} \omega^{*T} Y_{new} + b > 0, & l_{new} = 1 \text{ (an erroneous keystroke)}, \\ \omega^{*T} Y_{new} + b < 0, & l_{new} = -1 \text{ (a correct keystroke)}. \end{cases}$$

3.3.2.2 Support Vector Machine

SVMs are another group of binary classification tools, which have been successfully applied in many EEG classification problems [13, 84, 129, 71, 51]. The fundamental problem of SVM is to build an optimal decision boundary to separate two categories of data. In the data sets of EEG epochs D_1 and D_2, each EEG epoch is represented by a $n \times k$ dimensional feature vector. One can actually find infinitely many hyperplanes in $R^{n \times k}$ to separate the two data groups. Based on statistics learning theory (STL), a SVM selects a hyperplane which maximizes its distance from the closest point from the samples. This distance is referred to as *margin*. The standard SVM formulation that maximizes the *margin* and minimizes the training error is as follows:

$$\min_{\omega,\xi,b} \{ \tfrac{1}{2} \|\omega\|^2 + C \textstyle\sum_{i=1}^{p+q} \xi_i : D(Y^T \omega + be) \geq e - \xi_i \}, \tag{3.17}$$

where ω is the weight vector, and the slack variables ξ is introduced to measure the degree of misclassification during training. The penalty cost C is used to control the tradeoff between a large *margin* and a small prediction error penalty. Each column of Y is an observation Y_i, D is a diagonal matrix with class-label elements $D_i i$ equal to 1 if Y_i belongs to one class, or -1 otherwise. The vector e has all its elements equal to

one. The first term of the objective function in 3.17 is due to maximize the *margin* of separation $2/\|w\|$, and the second term measures how much emphasis is given to the minimization of the training error.

Since the standard SVM classifiers usually require a large amount of computation time for training, the Proximal SVM (PSVM) algorithm was introduced Mangasarian and Wild [99] as a fast alternative to the standard SVM formulation. The formulation for the linear PSVM is as follows:

$$\min_{\omega,\xi,b}\{\tfrac{1}{2}(\|\omega\|^2 + b^2) + \tfrac{1}{2}C{\xi_i}^T\xi_i : D(Y^T\omega + be) = e - \xi_i\}, \tag{3.18}$$

where the traditional SVM inequality constraint is replaced by an equality constraint. This modification changes the nature of the support hyperplanes ($\omega^T Y + b = \pm 1$). Instead of bounding planes, the hyperplanes of PSVM can be thought of as 'proximal' planes, around which the points of each class are clustered and which are pushed as far apart as possible by the term ($\|\omega\|^2 + b^2$) in the above objective function. It has been shown that PSVM has comparable classification performance to that of standard SVM classifiers, but can be an order of magnitude faster [99]. Therefore, we employed PSVM in this study.

3.4 Typing Experiment

3.4.1 Experimental Design

The experimental task was a typical hear-and-type task which emulated daily work done by bank tellers or representatives in customer services. A computer program read out 30 random numbers of nine digits in a trial and the subjects were told to type out those numbers. The numbers were not linguistically grouped, i.e. every digit was read out separately without chunking two or three digits (e.g., read 123 as "one two three" instead of "one twenty three" or "one hundred twenty three"). In addition, there was a small pause (300 ms) in-between every 3 digits. The numbers were read out this way because based on an observation and interview by the author in a pilot study,

this was the most natural way to read out numbers without any specific format known beforehand. The interval between two digits is 750ms on average. A short pause of 2.5 seconds existed after each nine-digit number, during which the subjects would be reminded of pressing the enter key.

Nine subjects were recruited from the student body of University at Buffalo. All subjects were native speakers of English without any hearing disability. Before the experiment, each subject was pre-tested on his/her typing skill to assure his/her familiarity with typing. The subjects were allowed to adjust the volume, posture and other settings of typing environment to his/her preference. Each subject was given two practice trials prior to formal experimental trials. If a subject did not show any inability in the hear-and-type task, she or he was then allowed to continue eight trials of hear-and-type tasks. During each trial, the EEG data of each subject were recorded. A five-minute break was given to subjects after four trials so that their EEG would not be influenced by long exposure to a relatively boring task.

The descriptive statistics of the typing performance of the nine subjects are summarized in Table 3.2. No significant difference was found in terms of age, accuracy or typing speed between male and female subjects. Hence, male and female subjects can be regarded as a homogeneous group. The percentage of erroneous keystrokes was ranged from 0.42% on subject 4 to 3.59% on subject 7. The latency between auditory stimuli and keystrokes was 728ms on average.

Table 3.2: The typing performance of each subject.

Subject	# Keystrokes	# Erroneous Keystrokes	Percentage of Erroneous Keystrokes
1	2122	36	1.70%
2	2113	51	2.41%
3	2419	69	2.85%
4	2401	10	0.42%
5	2422	60	2.48%
6	2117	76	3.59%
7	2420	45	1.88%
8	2405	63	2.62%
9	2134	54	2.53%

3.4.2 EEG Acquisition and Preprocessing

During the experiment, EEG data were collected with an EEG cap containing 40 Ag/AgCl electrodes according to the international 10-20 system. There are four electrodes that were used for measuring eye movements to remove muscular artifacts. The rest 36 electrodes were mounted on the scalp and thus used for analyses in this chapter. The placement of the 36 scalp electrodes is shown in Figure 3.1. The signals were amplified by NuAmps Express system (Neuroscan Inc, USA) and sampled at 1000Hz. The typed number as well as the timing of each keystroke was recorded simultaneously by the system. After comparing with the reference number, each keystroke was labeled as either 'correct' or 'erroneous' by using '1' and '-1', respectively. The raw EEG data were first processed by a 0.1-30Hz band-pass filter [119]. Then the EEG epochs were extracted from the filtered data based on the keystroke events recorded during typing. The length of each EEG epoch was set to 500 ms before a keystroke according to the minimal interval between two successive keystrokes. The flowchart of the EEG acquisition and the epoch sampling is shown in the upper part of Figure 3.2.

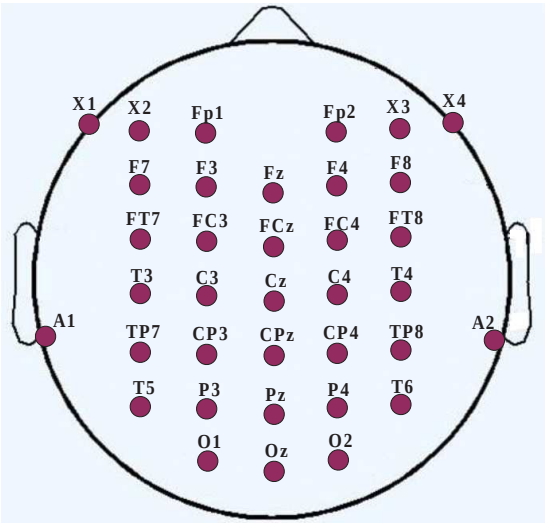


Figure 3.1: The allocations of the 36 scalp electrodes.

3.4.3 Classification Procedure

As shown in Figure 3.2, each 500ms EEG epoch was divided into five non-overlapping sub-epochs with equal length of 100ms. The size of sub-epochs was chosen empirically with the goal of obtaining salient information of the brain activity prior to keystrokes. The temporal, morphological and wavelet features of each 100ms EEG epoch were extracted. The feature vector of a multichannel EEG epoch was constructed by concatenating the feature vectors of all the channels. For example, if we want to classify the 100ms EEG epochs based on temporal features, then each epoch was represented by a $3 \times 36 = 108$ dimensional feature vector. Similarly, the morphological feature dimension for an EEG epoch is $7 \times 36 = 252$, and the wavelet feature vector has a dimension of $16 \times 36 = 576$.

3.4.4 Evaluation Metric of a Single Prediction

Sensitivity and specificity are commonly used performance measures of binary classification tests. For example, people are tested for a disease in a clinic study. Sensitivity is defined as the proportion of actual positives which are correctly identified as positive, and specificity is the proportion of negatives which are correctly identified as negative. In this study, we labeled the erroneous EEG samples as positive and the correct EEG samples as negative. Then we use sensitivity to measure the percentage of erroneous EEG samples that are correctly identified as positive, and specificity to measure the percentage of correct EEG samples that are correctly identified as negative. For each testing EEG sample, the classification result can be always categorized into one of the following four subsets:

- True positive (TP): if an erroneous EEG epoch is classified as positive.
- False positive (FP): if an correct EEG epoch is classified as positive.
- True negative (TN): if an correct EEG epoch is classified as negative.
- False negative (FN): if an erroneous EEG epoch is classified as negative.

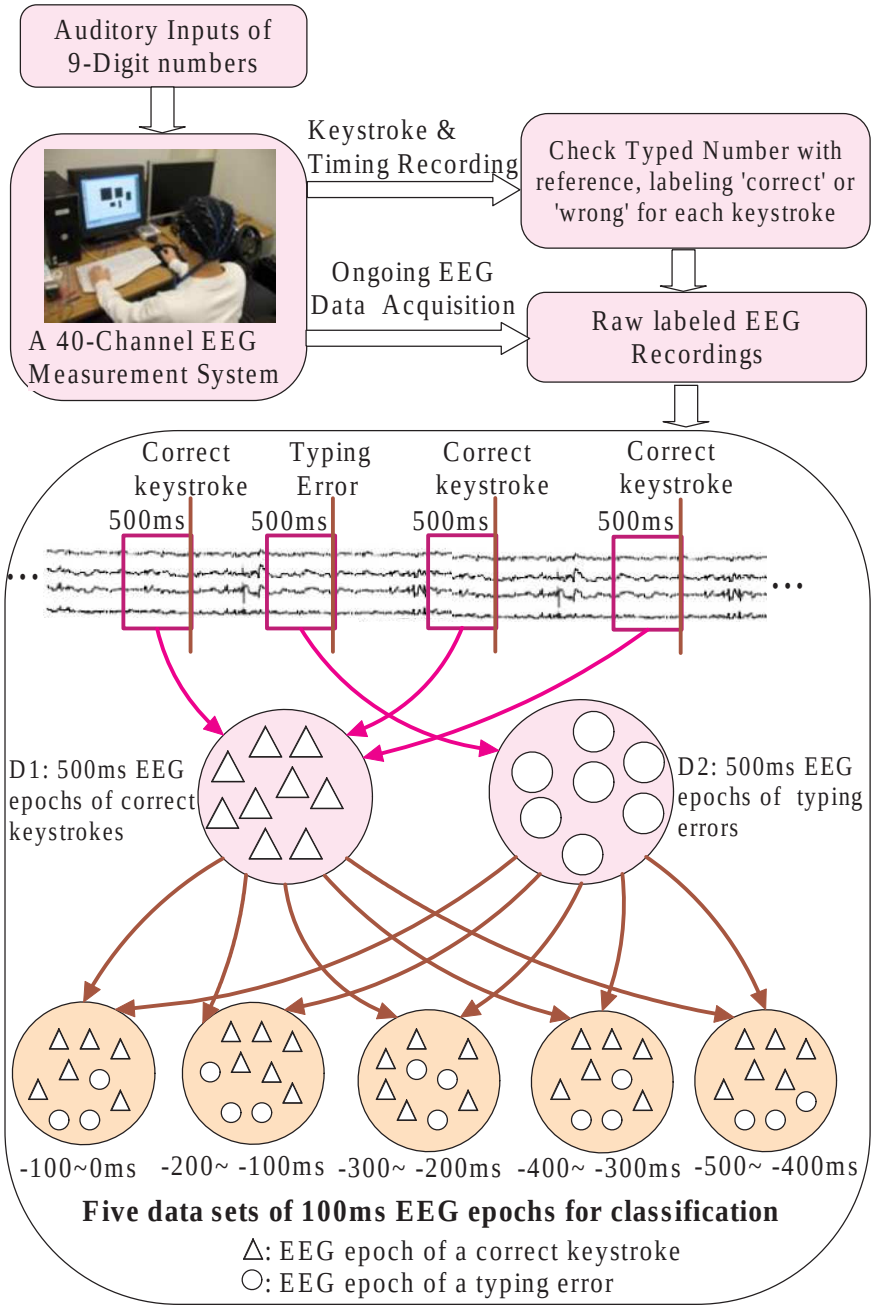


Figure 3.2: Flowchart of the typing experiment as well as the EEG acquisition and epoch sampling procedure. The 500ms EEG epochs were first extracted from the raw EEG data, and then they were divided into five 100ms sub-epochs corresponding to the five non-overlapping time intervals prior to keystrokes.

Then sensitivity and specificity can be calculated as follows:

$$sensitivity = \frac{TP}{TP+FN}, \tag{3.19}$$

$$specificity = \frac{TN}{FP+TN}. \tag{3.20}$$

3.4.5 Training and Evaluation

A standard classification problem generally follows a two-step procedure which consists of training and testing phases. During the training phase, a classifier is trained to achieve the optimal separation for the training data set. Then in the testing phase, the trained classifier is used to discriminate new samples with unknown class information. The leave-one-out cross validation is an attractive method of model evaluation, it is capable of providing almost unbiased estimate of the generalization ability of a classifier [150]. In this study, we trained and tested the classifiers under two frameworks, namely in-subject and cross-subject error detection. Correspondingly, two leave-one-out cross validation methods with perturbed duplications of erroneous samples were designed to achieve an unbiased estimate of the classification performance. The two methods are described in the following.

- In-subject Error Detection: training and testing on each subject individually. We employed a leave-one-error-pattern-out cross validation method for each subject. Let n_c and n_e denote the number of correct and erroneous keystrokes of a subject. Each time, we picked one erroneous EEG sample and $\lceil n_c/n_e \rceil$ correct samples out, and trained the classifier by the rest samples. To eliminate the unbalanced problem during training, we employed an oversampling method with perturbed replications of erroneous samples. Let n_c^t and n_e^t denote the number of correct and erroneous samples in the training data set ($n_c^t \gg n_e^t$). Then the feature vector of each erroneous sample was replicated $\lceil n_c^t/n_e^t \rceil$ times. For each replication, a synthetic erroneous feature vector was generated by adding a random perturbation to the original erroneous feature vector. In particular, let Y_j be a feature vector of an erroneous EEG sample, then a synthetic erroneous feature vector Y_j' can be created by

$$Y_j' = Y_j + \alpha \times \frac{\bar{Y}_e - Y_j}{\tau_e}, \tag{3.21}$$

  where $\bar{Y}_e$ and τ_e are $1 \times 36k$ vectors, which contain the means and standard deviations of the $36k$ features for all the erroneous samples in the training data set, α is a random number uniformly generated in [-1, 1]. The trained classifiers

were tested on the left-out samples. Repeat the procedure for all the erroneous samples of a subject. The averaged prediction result was used to indicate the classification effectiveness based on the current data set.

- Cross-subject Error Detection: In this framework, we speculate that the erroneous EEG patterns of different subjects may share some common characteristics due to a high level of uncertainty or anxiety prior to making typing errors. There have been a number of recent BCI studies focusing on subject-independent ERP classification. The studies showed that the EEG potentials of different subjects may exhibit similar waveform characteristics in performing the same mental task [39, 98]. Stemmed from this consideration, we designed a leave-one-subject-out method to train and evaluate the classifiers. Each time we picked one subject out, and trained the classifiers by the EEG samples from the rest eight subjects. The oversampling method with perturbed replications of erroneous samples was also used to form a balanced training data set. The EEG samples of the left-out subject were considered as unknown samples to test the trained classifiers. Repeating this procedure for all the subjects, the averaged prediction accuracy can be used to indicate the effectiveness of the trained classification models.

3.4.6 Receiver Operating Characteristic Analysis

ROC analysis is another popular method to evaluate the performance of a prediction model. A ROC curve is a plot of sensitivity versus false alarm rate (1-specificity) as the discriminant threshold of a classifier varied throughout its possible ranges. The ROC curve for a perfect prediction model is the line connecting [0, 0] to [0, 1] and [0, 1] to [1, 1]. And the diagonal line connecting [0, 0] to [1, 1] is the ROC curve corresponding to a random model. Generally, a ROC curve lies between these two extreme lines. The *area under the ROC curve* (AUC) is often used as a important metric to evaluate a prediction model. The AUC is an overall summary of prediction accuracy across the spectrum of its decision-making values. AUC values are usually between 0.5 and 1. The AUC of a perfect predictor is 1 while a purely random chance model has an AUC

of 0.5 on average. The higher the AUC value is to one, the better the prediction power a predictor has. A typical generation procedure of the ROC curve for a classifier is demonstrated in Figure 3.3. One may also find that the value of AUC may also be a classificability index of the two data sets without knowing their exact distributions based on the current classification framework.

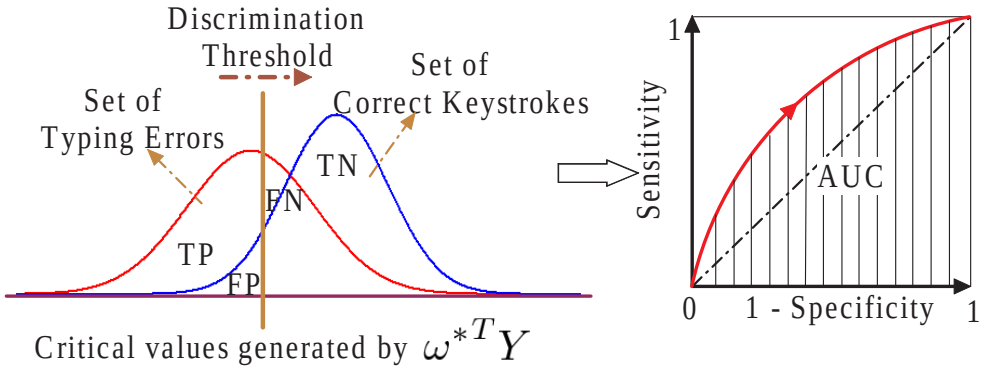


Figure 3.3: A demonstration of ROC as the discrimination threshold of a classifier (LDA or PSVM) is varied through the whole range of its possible values. The value of AUC indicates the overall performance of a classifier. It may also indicate the classificability of the two data sets without knowing the distributions of the two data sets based on the current classification framework.

3.5 Results

3.5.1 In-Subject Sensitivity and Specificity Analysis

Table 3.3 summarize the in-subject training and testing sensitivity and specificity of LDA and PSVM based on the leave-one-error-pattern-out cross validation methodology using the three choices of EEG features. The best training performance was achieved by the wavelet features for both LDA and PSVM. Using wavelet features, the training sensitivity and specificity were above 90% for both LDA and PSVM at all the five time intervals. It had an averaged training sensitivity of above 90% and an averaged specificity of above 80% when using morphological features. The temporal features had the worst training performance, which had an averaged training sensitivity of above 80% and an averaged specificity of above 70%. As for the testing performance, a noteworthy

observation is that the best testing results of LDA and PSVM were both achieved at the time interval of -100ms $\sim$ 0ms. In particular, the best testing performance of LDA was achieved at a sensitivity of 62.77% and a specificity of 51.03% when using morphological features, while the best testing performance of PSVM had a sensitivity of 62.20% and a specificity of 51.68% when using morphological features. In a contrast experiment, we also tested a randomized detection model with prior probability of error rate (RDPP). For a subject with an error rate of p, the RDPP classified each EEG sample as erroneous with a probability of p, and as correct with a probability of $1 - p$. The testing results of the RDPP are shown in the last row of Table 3.3. It was noted that only about 2% of the erroneous keystrokes can be detected on average by the RDPP, while both LDA and PSVM detected more than 60% of the erroneous keystrokes at the time interval of -100ms $\sim$ 0ms. Our trained classification models considerably increased the error detection rate.

In addition, the averaged testing sensitivities of LDA and PSVM over the nine subjects and the three choices of features for the five time intervals are shown in Figure 3.4. Interestingly, the error detection accuracies tended to increase as the time interval became closer to the timing of keystrokes, especially at the last three 100ms time intervals. This observation may indicate that the closer the analyzed EEGs to the keystrokes, the more prominent brainwave patterns can be captured to discriminate an upcoming erroneous keystroke from correct ones. This result nicely matches with our physiological intuition and the previous study of Blankertz et. al. in [13], which also reported an increased classification accuracies in detecting upcoming finger movements (keystrokes) based on EEG recordings prior to the keystrokes. They also claimed that the most salient information of brain may be gained within 230ms before the finger movements based on their experiments. However, this hypothesis still needs further investigation in future work.

Table 3.3: In-Subject training and testing results of LDA, PSVM and a Random model based on the leave-one-error-pattern-out cross validation methodology (Results were all averaged over the nine subjects).

	Classifier	Feature	-500ms		-400ms		-300ms		-200ms		-100ms	
			sen.	spe.	sen.	spe.	sen.	spe.	sen.	spe.	sen.	spe.
Training Results	LDA	Temporal	87.11%	76.33%	87.64%	76.76%	88.26%	76.84%	85.07%	76.66%	89.63%	79.19%
		Morph.	95.08%	84.64%	95.62%	85.20%	95.28%	85.20%	96.41%	85.21%	95.71%	86.16%
		Wavelet	99.59%	93.67%	99.78%	94.43%	99.59%	93.28%	99.34%	93.64%	99.23%	93.71%
	PSVM	Temporal	87.16%	75.74%	87.33%	76.37%	87.03%	76.70%	86.90%	76.77%	87.40%	78.08%
		Morph.	93.73%	82.71%	95.57%	84.01%	93.22%	83.36%	96.08%	83.88%	95.89%	85.05%
		Wavelet	99.28%	92.42%	99.74%	92.64%	99.52%	92.60%	99.63%	93.45%	99.66%	93.68%
Testing Results	LDA	Temporal	56.03%	50.02%	57.47%	49.86%	54.88%	49.26%	61.28%	49.68%	61.74%	49.83%
		Morph.	54.48%	50.35%	55.93%	51.13%	58.81%	50.73%	55.17%	51.23%	**62.77**%	**51.03**%
		Wavelet	53.10%	49.72%	55.91%	50.41%	52.03%	50.85%	54.74%	51.97%	56.95%	49.70%
	PSVM	Temporal	55.62%	50.16%	57.83%	49.58%	55.89%	49.44%	59.88%	49.61%	63.15%	49.37%
		Morph.	52.87%	50.48%	55.82%	50.75%	60.13%	51.05%	58.92%	50.91%	**62.20**%	**51.68**%
		Wavelet	50.98%	50.49%	57.00%	50.28%	51.37%	50.63%	55.30%	51.64%	57.00%	50.52%
	RDPP	-	2.35%	97.73%	2.18%	97.74%	2.33%	97.72%	2.21%	97.72%	2.32%	97.73%

Table 3.4: Cross-Subject Training and testing results of LDA, PSVM and a Random Model based on the leave-one-subject-out cross validation methodology (Results were all averaged over the nine subjects).

	Classifier	Feature	-500ms		-400ms		-300ms		-200ms		-100ms	
			sen.	spe.	sen.	spe.	sen.	spe.	sen.	spe.	sen.	spe.
Training Results	LDA	Temporal	51.67%	79.52%	52.57%	85.15%	52.64%	86.00%	54.89%	71.75%	55.04%	72.96%
		Morph.	68.09%	64.75%	65.59%	63.25%	69.06%	66.18%	68.04%	62.01%	70.00%	63.89%
		Wavelet	62.47%	69.66%	66.03%	68.38%	63.64%	64.73%	66.10%	68.34%	67.81%	71.06%
	PSVM	Temporal	46.69%	93.06%	47.48%	91.99%	48.83%	89.00%	52.09%	79.39%	56.06%	73.15%
		Morph.	66.46%	66.20%	66.40%	62.63%	67.44%	64.80%	66.04%	63.57%	69.25%	64.69%
		Wavelet	60.94%	74.19%	64.97%	72.66%	61.52%	69.18%	62.74%	69.87%	67.29%	71.98%
Testing Results	LDA	Temporal	63.39%	48.50%	63.98%	49.52%	59.75%	50.43%	64.17%	48.67%	**68.72**%	**49.45**%
		Morph.	55.74%	54.41%	55.53%	56.29%	60.39%	53.84%	60.84%	53.12%	60.88%	53.66%
		Wavelet	58.73%	49.97%	57.21%	50.88%	62.37%	48.83%	61.51%	50.21%	63.63%	51.84%
	PSVM	Temporal	54.85%	55.43%	54.65%	58.40%	55.08%	56.55%	61.21%	53.46%	**66.63**%	**51.30**%
		Morph.	54.40%	55.68%	54.25%	56.20%	60.35%	53.51%	56.87%	56.41%	60.61%	54.46%
		Wavelet	58.23%	50.97%	57.49%	52.79%	62.23%	48.93%	61.52%	51.38%	62.09%	52.86%
	RDPP	-	2.31%	97.74%	2.25%	97.74%	2.26%	97.74%	2.29%	97.74%	2.23%	97.74%

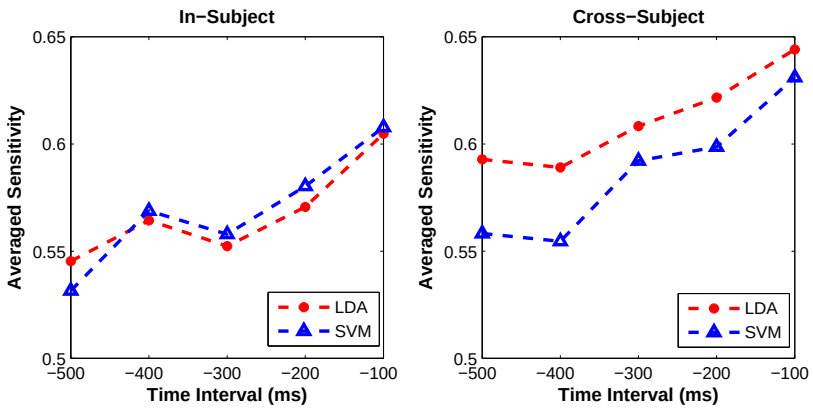


Figure 3.4: The averaged testing sensitivity of LDA and PSVM over the nine subjects and the three choices of features for the five time intervals. In both in-subject and cross-subject experiments, there is an increasing trend of error detection accuracy as the time interval moves closer to the timing of keystrokes.

3.5.2 Cross-Subject Sensitivity and Specificity Analysis

Table 3.4 summarize the cross-subject training and testing performance based on the leave-one-subject-out cross validation methodology. It is noted that the cross-subject training performance was worse than in-subject training performance. The best training performance of LDA has a sensitivity of 52.64% and a specificity of 86.00%, and that of PSVM was achieved at a sensitivity of 46.69% and a specificity of 93.06%. As for testing performance, it is interesting to observe that the cross-subject testing performance was comparable to in-subject testing performance. Also it is worth mentioning that the best testing performance was achieved at the time interval of -100ms $\sim$ 0ms for both LDA and PSVM. In particular, the best testing performance of LDA has a sensitivity of 68.72% and a specificity of 49.45%, and PSVM has a sensitivity of 66.63% and a specificity of 51.30% at best. These results indicate that the erroneous EEG at the time interval of -100ms $\sim$ 0ms may exhibit more prominent patterns than the other four time intervals, which lead to increased classification accuracies. More importantly, the classifiability of erroneous and correct EEG samples across the subjects confirmed our hypothesis that different subjects may exhibit some similar EEG patterns prior to erroneous actions. Otherwise, the leave-one-subject-out method would produce an overall accuracy no better than a chance level. The subject-independent erroneous EEG

potentials may be associated with a high level of uncertainty or anxiety prior to wrong response actions. Such uncertainty/anxiety related EEG potentials may have much in common for human beings.

3.5.3 Receiver Operating Characteristic Analysis

The ROC analysis is an important method to further investigate the classificability of the erroneous and correct EEG samples. Table 3.5 and Table 3.6 present the in-subject and cross-subject AUC values of the nine subjects based their best choices of features. The corresponding in-subject and cross-subject ROC curves are shown in Figure 3.5 and Figure 3.6, respectively. From the ROC plots, one can observe that both in-subject and cross-subject ROC curves of the nine subjects are apparently deviated from the 45-degree diagonal line which represents a random chance level, especially the last three time intervals. These ROC curves suggested that the distribution of erroneous EEG patterns might be different from that of correct ones.

In addition, AUC is a convenient indicator of the discrimination between the two distributions of erroneous and correct EEG samples. As for in-subject experiments, the best AUC value of LDA was 0.76 achieved at subject 4 using temporal features at -200ms $\sim$ -100ms. The best AUC value of PSVM was 0.80 achieved also at subject 4 using temporal features at -200ms $\sim$ -100ms. The best averaged AUC values were 0.63 and 0.64 for LDA and PSVM, respectively. They were both achieved at the time interval of -100ms $\sim$ 0ms. In the cross-subject experiments, the best AUC values of LDA and PSVM were both 0.78 achieved at subject 4 using temporal features at the time interval of-100ms $\sim$ 0ms. The best averaged cross-subject AUC value of LDA was 0.62 achieved at -200ms $\sim$ -100ms, and the best averaged AUC value of PSVM was also 0.62 achieved at both time intervals of -200ms $\sim$ -100ms and -100ms $\sim$ 0ms. It is noted that the averaged in-subject and cross-subject AUC values were all above 0.60 at the last three time intervals of -300ms $\sim$ 200ms, -200ms $\sim$ 100ms, and -100ms $\sim$ 0ms. Also, we notice that the classification accuracy on subject 4 was generally higher than that of other nine subjects. When excluding subject 4, we still can get an averaged AUC values of around

0.60 at the last three time intervals. These results further confirmed our hypothesis that the most salient information of the brain activity associated with erroneous keystrokes may be gained within 300ms prior to keystrokes. The AUC values indicated that the distributions of erroneous and correct EEG patterns might be different. As a result, erroneous keystrokes might be predictable based on EEG recordings.

Table 3.5: In-Subject AUC Values of LDA and PSVM based on the best choice of features.

	sub.	-500ms		-400ms		-300ms		-200ms		-100ms	
		AUC	Feat.	AUC	Feat.	AUC	Feat.	AUC	Feat.	AUC	Feat.
LDA	1	0.56	Morp.	0.53	Temp.	0.62	Temp.	0.63	Wave.	0.68	Temp.
	2	0.59	Wave.	0.62	Morp.	0.55	Morp.	0.53	Morp.	0.58	Temp.
	3	0.59	Wave.	0.58	Temp.	0.59	Morp.	0.53	Morp.	0.64	Morp.
	4	0.6	Temp.	0.61	Morp.	0.75	Morp.	**0.76**	Temp.	0.75	Temp.
	5	0.61	Temp.	0.59	Wave.	0.58	Temp.	0.57	Temp.	0.61	Wave.
	6	0.68	Morp.	0.65	Temp.	0.66	Morp.	0.63	Morp.	0.64	Morp.
	7	0.61	Temp.	0.57	Morp.	0.62	Temp.	0.65	Morp.	0.61	Morp.
	8	0.57	Temp.	0.53	Temp.	0.6	Temp.	0.58	Wave.	0.54	Temp.
	9	0.59	Morp.	0.58	Morp.	0.62	Morp.	0.56	Morp.	0.62	Temp.
	ave.	**0.60**	-	**0.58**	-	**0.62**	-	**0.60**	-	**0.63**	-
PSVM	1	0.61	Morp.	0.53	Temp.	0.67	Temp.	0.65	Wave.	0.65	Temp.
	2	0.59	Wave.	0.67	Temp.	0.55	Temp.	0.55	Temp.	0.66	Morp.
	3	0.59	Wave.	0.58	Temp.	0.61	Morp.	0.59	Morp.	0.64	Morp.
	4	0.59	Temp.	0.59	Wave.	0.74	Morp.	**0.80**	Temp.	0.72	Temp.
	5	0.6	Temp.	0.57	Wave.	0.56	Temp.	0.54	Morp.	0.62	Temp.
	6	0.65	Morp.	0.63	Temp.	0.64	Morp.	0.67	Morp.	0.65	Morp.
	7	0.6	Temp.	0.54	Morp.	0.6	Temp.	0.63	Morp.	0.62	Wave.
	8	0.57	Temp.	0.54	Morp.	0.59	Temp.	0.61	Wave.	0.57	Wave.
	9	0.59	Morp.	0.6	Morp.	0.55	Temp.	0.57	Wave.	0.67	Morp.
	ave.	**0.60**	-	**0.58**	-	**0.61**	-	**0.62**	-	**0.64**	-

3.6 Conclusion

In this Chapter, we applied the start-of-the-art data mining techniques to investigate EEG time series patterns during numerical typing. The temporal, morphological, and wavelet-based time-frequency features were extracted. Popular data mining tools LDA and PSVM were employed in this binary classification task. Since the number of erroneous EEG samples of each subject was too few to train the classifiers, we designed the in-subject leave-one-pattern-error-out and the cross-subject leave-one-subject-out cross validation methodology to achieve an unbiased estimate of classification performance. The experimental results of this study were promising. The averaged in-subject and

Table 3.6: Cross-Subject AUC Values of LDA and PSVM based on the best choice of features.

	subject	-500ms		-400ms		-300ms		-200ms		-100ms	
		AUC	Feat.	AUC	Feat.	AUC	Feat.	AUC	Feat.	AUC	Feat.
LDA	1	0.56	Temp.	0.57	Temp.	0.57	Temp.	0.68	Wave.	0.55	Temp.
	2	0.53	Morp.	0.58	Temp.	0.56	Morp.	0.59	Morp.	0.57	Temp.
	3	0.6	Morp.	0.58	Wave.	0.58	Wave.	0.57	Temp.	0.59	Temp.
	4	0.67	Temp.	0.6	Temp.	0.73	Wave.	0.76	Temp.	**0.78**	Temp.
	5	0.6	Temp.	0.56	Morp.	0.57	Temp.	0.6	Temp.	0.64	Temp.
	6	0.64	Morp.	0.64	Morp.	0.64	Morp.	0.66	Morp.	0.63	Morp.
	7	0.59	Morp.	0.59	Temp.	0.62	Morp.	0.56	Wave.	0.61	Morp.
	8	0.55	Temp.	0.56	Temp.	0.57	Temp.	0.6	Temp.	0.58	Wave.
	9	0.57	Morp.	0.6	Morp.	0.61	Morp.	0.59	Morp.	0.57	Temp.
	ave.	**0.59**	-	**0.59**	-	**0.61**	-	**0.62**	-	**0.61**	-
PSVM	1	0.56	Wave.	0.57	Morp.	0.57	Temp.	0.68	Wave.	0.57	Wave.
	2	0.53	Morp.	0.59	Wave.	0.56	Morp.	0.59	Morp.	0.57	Temp.
	3	0.6	Morp.	0.58	Temp.	0.58	Wave.	0.57	Temp.	0.59	Temp.
	4	0.67	Temp.	0.64	Temp.	0.73	Wave.	0.75	Temp.	**0.78**	Temp.
	5	0.6	Temp.	0.56	Morp.	0.57	Temp.	0.6	Temp.	0.64	Temp.
	6	0.64	Morp.	0.64	Morp.	0.64	Morp.	0.66	Morp.	0.63	Morp.
	7	0.57	Morp.	0.6	Wave.	0.61	Morp.	0.56	Morp.	0.61	Morp.
	8	0.55	Temp.	0.56	Temp.	0.59	Temp.	0.6	Temp.	0.58	Wave.
	9	0.57	Morp.	0.6	Morp.	0.58	Morp.	0.6	Morp.	0.57	Temp.
	ave.	**0.59**	-	**0.59**	-	**0.60**	-	**0.62**	-	**0.62**	-

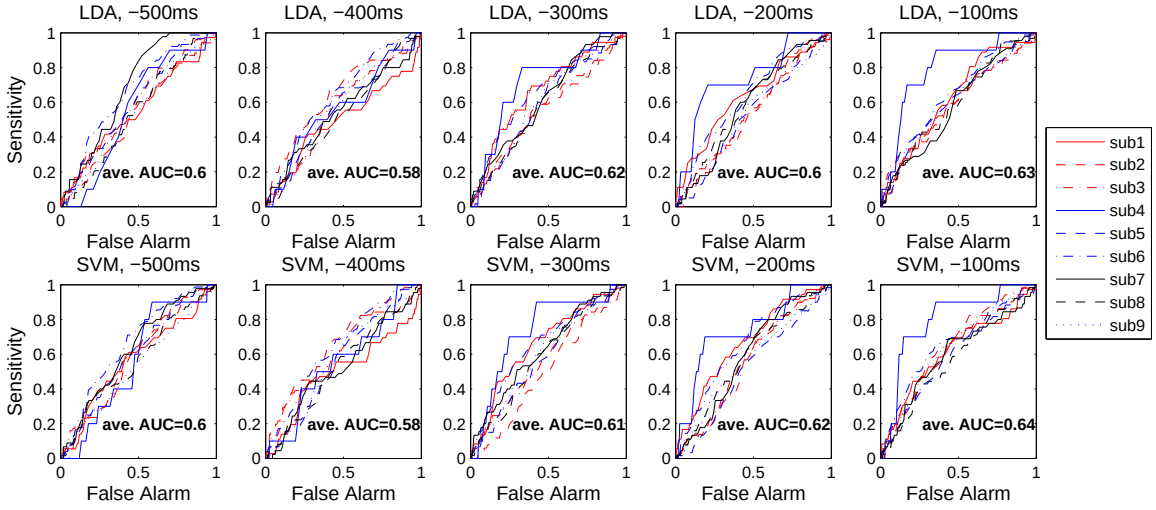


Figure 3.5: The in-subject ROC curves of the nine subjects at each time interval for LDA and PSVM based on their best choice of features. The averaged AUC value over the nine subjects is denoted in the bottom part of each subplot.

cross-subject AUC values were both above 0.60 at the last three time intervals of -300ms $\sim$ 200ms, -200ms $\sim$ 100ms, and -100ms $\sim$ 0ms. These results indicated that the distribution of erroneous EEG patterns may be considerably different from that of correct

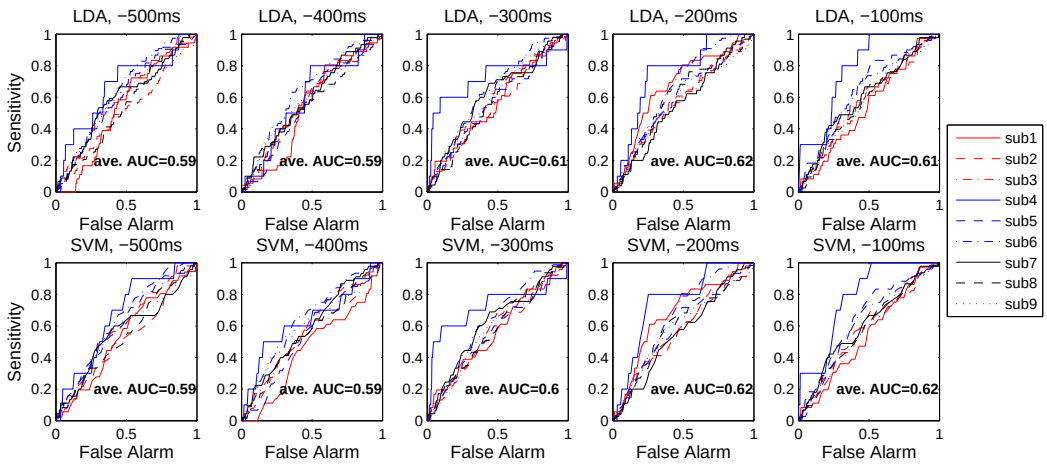


Figure 3.6: The cross-subject ROC curves of the nine subjects of LDA and PSVM at each time interval based on their best choice of features. The averaged AUC value over the nine subjects is denoted in the bottom part of each subplot.

ones, especially at the last 300ms prior to keystrokes. The results are very encouraging considering that the classification problem of this study is extremely challenging due to the highly imbalanced data structure, and that we only used a very simple and straightforward classification framework. This study confirmed our hypothesis that it is possible to predict a future event (such as an erroneous keystroke) based on EEG recordings.

All our experiments were performed based on nine subjects. The number of subjects is limited due to difficulties in recruiting subjects and complex experimental settings. Although this study based on the limited data pool might not represent a generalized result for all people, the concept of automated EEG-based online monitoring and prediction system seems to be conceivable. In the next chapter, we will explain an reinforcement learning based online prediction framework based on EEG time series data. The whole framework is explained in detail by solving the challenging seizure prediction problem.

Chapter 4

Reinforcement Learning-Based Online Monitoring and Prediction Approach: an Application to Seizure Prediction

Epilepsy is one of the most common neurological disorders, affecting approximately 1% of the world's population. The sudden and spontaneous occurrence of epileptic seizures imposes a significant burden on patients with epilepsy. Being able to predict impending seizures could greatly improve the life of patients with epilepsy. One prominent challenge in seizure prediction is the high intra- and inter- individual variability of epileptic seizures and their episodic events. In this study, we propose a novel autonomous adaptive learning approach for online seizure prediction based on analysis of electroencephalogram (EEG) recordings. For each individual patient, after the first seizure in the EEG recording, we construct baseline patterns of normal and pre-seizure EEG samples. Then our approach monitors continuous EEG recordings using sliding windows, and classify each of the EEG windows as normal or pre-seizure by comparing it with the baseline samples using a K-nearest-neighbor (KNN) method. We proposed a gradient-based reinforcement learning algorithm to update the normal and preseizure baseline patterns online based on the feedback of each prediction (true or false). The proposed approach was evaluated by an EEG dataset of 10 patients with epilepsy. For each one of the 10 patients, the adaptive algorithm was trained to find the best parameter setting using the EEG recordings containing the first half of seizure occurrences. With the best setting, our approach was tested *prospectively* on the EEG recordings containing the second half of seizure occurrences. The testing prediction performance using the prediction horizon of 150 minutes yields the sensitivity of 73%

and the specificity of 67% on average over 10 patients. We also compared the performance of our approach with that of a non-update prediction scheme and two native prediction schemes (periodic and Poisson). We performed a receiver operating characteristic (ROC) analysis on each prediction scheme, and the area under the ROC curve (AUC) was used to compare the prediction power of different prediction schemes. The statistical validation of the results demonstrated that the proposed adaptive learning approach outperformed the the non-update and the two naive prediction schemes with the p-values smaller than 0.001. The results of this study are considered as an ample evidence of the need of adaptive baseline/model update in online monitoring problems.

4.1 Introduction

Epilepsy is one of the most common neurological disorders, affecting approximately 1% of the world's population [34]. Epileptic seizures generally occur without warning, and the shift between a normal brain state and seizure onset is often considered an abrupt, unpredictable phenomenon. The unpredictability of seizures represents a significant source of morbidity in patients with epilepsy. Patients with epilepsy frequently suffer from seizure-related injuries due to loss of motor control, loss of consciousness or delayed reactivity during seizures [117]. Current technology has yet to reach a point where epileptic patients can be warned by an automated system prior to seizure onsets. The ability to predict the occurrence of impending seizures could significantly improve the life quality of epileptic patients.

One crucial question in seizure prediction is whether an identifiable, specific, pre-seizure state exists. Over the recent years, there has been accumulating evidence indicating that a transitional pre-seizure state does exist prior to seizure onsets [64, 89, 126, 106, 90, 19, 22]. The majority of the quantitative evidence supporting the existence of a pre-seizure state is derived from EEG analyses. For example, Iasemidis et al. [64] noted premonitory pre-seizure changes based on the analysis of dynamical entrainment. Lehnertz and Elger [89] showed that the correlation dimension decreases prior to seizures. Le van Quyen et al. [126] reported a reduction in the dynamical similarity index before

seizure occurrence. Mormann et al. [106] observed that there was a relative decrease of signal power in the delta band of the EEG up to hours prior to seizure onsets. They also demonstrated statistically significant discrimination between pre-seizure and normal brain states. In our previous study, Chaovalitwongse et al. [21] investigated the EEG characteristics of pre-seizure transition and found that the probability of detecting pre-seizure transition was as high as 83% using the optimized critical EEG channels. In later studies, a network-based approach was built to study the evolution of epileptic seizures. The evolutional structural changes of the brain network hours prior to seizure onsets indicated that the seizures may slowly develop by an evolutional epileptogenic process [22, 24].

The current seizure prediction algorithms generally employ some EEG features as precursors of imminent epileptic seizures. If the extracted EEG features cross an optimized threshold, a warning is issue for a patient. Examples of published features include dynamical entrainment [67, 54], correlation dimension [88], dynamic similarity index [126], accumulated energy [96], phase synchronization [105], wavelet and median filtering [111]. Recently, Feldwisch-Drentrup et al. [40] investigated the possibility of combining different seizure prediction algorithms and different EEG features to improve prediction accuracy. Using Boolean operations, they showed the different prediction methods with different EEG features can be combined and can generate significant better performance than each individual method. In particular, they found that sensitivity can be markedly improved by combining dynamic similarity index [126] and phase synchronization [105], given a fixed maximum FPR.

A significant challenge of seizure prediction is the high inter- and intra-individual variability of epileptic seizures with a variable degree of success [66]. Although many nonadaptive methods have achieved promising results, this variability makes it difficult to develop a universal robust predictor to accurately predict seizures for a wide range of patients with different seizures. This variability also highlights the emerging need for an automated adaptive approach for epileptic seizure prediction. A number of adaptive seizure prediction algorithms have been proposed to account for the high variability of epileptic seizures [66, 67, 135, 128, 25]. Iasemidis et al. [66, 67] and Sackellares et al.

[135] developed optimization-based prediction algorithms which, based on dynamical synchronization in the human epileptic brain, adaptively selects a group of critical EEG electrodes to predict impending seizures. More recently, Iasemidis's group published similar results, with high sensitivity (85.9%) and specificity (0.18 false positive rate (FPR) per hour), and long warning times prior to seizures (67.6 minutes on average), on prospective seizure prediction in rodents with chronic epilepsy [54]. Rajdev et al. [128] also proposed an adaptive prediction algorithm based on a Wiener implementation of autoregressive (AR) modeling. A warning was issued if the prediction errors over a moving window exceeded a threshold. The threshold was continuously updated online, and it was optimized to maximize sensitivity and latency, while minimizing FPR. This algorithm achieved an averaged sensitivity of 92% on four rats with 70 seizures. This study also compared the proposed algorithm with the state-of-the-art seizure prediction algorithms [67, 88, 126, 96, 27, 105, 111]. In particular, we are interested to compare the two most recent adaptive algorithms in Rajdev et al. [128] and Iasemidis et al. [54]. It is noted that the FPR in [128] was 4.8/hour, which is much higher than that in [67] (0.18/hour). And also the averaged warning time in [128] is only 6.7 seconds, which is much shorter than that in [67] (67.6 minutes).

The current a few adaptive seizure prediction approaches are generally based on an adaptively-optimized set of EEG channels [66, 67, 135] or an adaptive threshold [128]. In principle, these approaches employed the prediction settings optimized by one or several recently occurred seizures to predict the next seizure. Due to the high intra-individual variability of epileptic seizures, the characteristics of the EEG patterns of the next seizure may become quite different from those of its preceding ones. The current adaptive approaches actually do not make full use of the whole monitored EEG recordings, and thus have problems to deal with the challenging problems of high intra-individual variability of seizures in prediction. Therefore, it is extremely desirable to enable a prediction system to accumulate more and more knowledge of predictive EEG patterns over time instead of only holding 'short memories'.

To tackle this problem, we propose a novel adaptive learning approach for prospective online seizure prediction. For each individual patient, after the first seizure in the

EEG recording, we construct baseline patterns of normal and pre-seizure EEG samples. Our approach monitors continuous EEG recordings using sliding windows, and classify each of the EEG windows as normal or pre-seizure by comparing it with the baseline samples using a K-nearest-neighbor (KNN) method. We proposed a gradient-based reinforcement learning algorithm to update the normal and preseizure baseline patterns online based on the feedback of each prediction (true or false). This study is among the first to investigate adaptive learning algorithms to solve the challenging online monitoring and prediction problem of seizure prediction [66, 135, 55, 128]. It is noted that the seizure prediction approach has to work with a seizure detection algorithm to provide prediction feedbacks for baseline updating. Since there have been a number of automated seizure detection algorithms embedded in clinical EEG systems, our proposed prediction approach can be readily integrated to the current EEG systems. The proposed adaptive learning approach eliminates the need for a complicated threshold-tuning process, and makes it possible to achieve a personalized seizure prediction in real clinical applications. Since seizure detection is beyond the scope of this research, we assume that all seizures can be detected perfectly in this study.

This chapter is organized as follows. In section 4.2, the background and previous related work are discussed. The data collection, feature extraction, the adaptive seizure prediction approach, and the evaluation metrics of prediction performance are presented in section 4.3. The experimental results are provided and discussed in Section 4.4, and we conclude the chapter in Section 4.5.

4.2 Background and Related Works

4.2.1 Overview of Machine Learning Techniques

With the explosion of computing power in the past decade, machine learning and pattern recognition techniques have become important tools in the analysis of various biological problems, such as cancer research [93], cognitive neuroscience [29], and genomics and proteomics [26]. Machine learning best depicts the computational methods that allow a system to evolve behaviors through an automated process of knowledge

acquisition from empirical data. Machine learning techniques generally fall into three broad categories: supervised learning, reinforcement learning and unsupervised learning. A supervised learning technique usually first finds a mapping between inputs and outputs of a training dataset, and then makes predictions for inputs that it has never seen. A large number of supervised learning algorithms have been developed that can be categorized into several major groups, including neural networks, support vector machines, locally weighted learning, decision trees, and Bayesian inference [83]. Reinforcement learning is another learning paradigm in which an agent is able to learn a decision policy by 'trial and error'. A reinforcement learner receives feedback of its actions and makes adjustments to its actions accordingly [154]. Reinforcement learning is a natural framework for building models to accumulate knowledge from previously learned tasks to new tasks with increasing complexity and variability. Reinforcement learning techniques have been applied to many complex learning tasks, such as robot control [33] and traffic network control [132]. Unsupervised learning is inspired by the brain's ability to recognize complex patterns of visual scenes, sounds or odors. It takes root in neuroscience/psychology and is established on the basis of information theory and statistics. An unsupervised learner usually performs clustering or associative rule learning to extract the implicit structure of a given dataset. The established clusters, categories or associative networks are then used for decision making, prediction, or efficient communication [31].

4.2.2 EEG Analysis for Epileptic Seizures

Most seizure prediction methods are based on quantitative analysis of the EEG, and can be broadly categorized into univariate and multivariate analysis, respectively.

Univariate analyses focus on the features of each single channel of EEG. Based on the morphological characteristics of EEG, Lange et al. [85] reported that there were consistent changes in EEG spike activity prior to seizures. With the help of advanced signal processing methods, more complex univariate EEG feature extraction techniques have been developed for seizure prediction. Litt et al. [96] introduced signal energy variations to seizure prediction, and reported EEG changes hours before seizure onsets.

Autoregressive (AR) and autoregressive moving average (ARMA) models have also been utilized for seizure prediction. Characteristic changes of AR/ARMA coefficients before seizure onsets were reported in [136, 25]. Nonlinear measures based on chaos theory have drawn considerable attention in EEG studies of brain activity. The two well-known nonlinear chaotic measures that have been applied in seizure prediction are the Lyapunov exponent and correlation dimension. Iasemidis et al. [66] monitored the evolution of Lyapunov exponents extracted from EEG data. They designed an adaptive prediction scheme that attempted to select the most informative channels to predict an impending seizure with optimization techniques. Channel selection was adjusted after every seizure since it was assumed that the pre-seizure dynamics may change from seizure to seizure over time. Lehnertz et al. [88] investigated the feasibility of seizure prediction based on transitions of correlation dimension, a feature that is considered as an index of neuronal complexity.

Multivariate analyses take more than one channel of EEG into account simultaneously rather than only looking at each channel individually. The most influential multivariate analysis methods in seizure prediction are phase synchronization and dynamical entrainment. Le Van Quyen et al. [127] used phase synchronization to distinguish pre-seizure features from normal state. They compared the normal synchronization patterns taken from 3-10 hours before seizures with the pre-seizure patterns taken from 30 minutes before seizures. The variables that achieved best discriminating performance were chosen for each individual patient. Mormann et al. [105] designed a seizure prediction scheme based on their finding that the degree of synchronization may decrease up to hours prior to seizure onsets. Iasemidis et al. [67] explored the effectiveness of a method called dynamical entrainment, which estimated the difference of the largest Lyapunov exponents from any two observed time series of EEG. A progressive convergence of the dynamical entrainment was considered as sign of transition from normal to pre-seizure states.

4.2.3 Related Work in Seizure Prediction and Challenges

In the 1970s, accumulating evidence from clinical practice suggested that epileptic seizures might be predicable. Viglione and Walsh started a project to investigate the predictability of seizures based on EEG data [157]. Iasemidis et al. pioneering work started in the 1980s [68, 69, 65]. Since then, many studies have been carried out aiming to predict epileptic seizures.

Most current seizure prediction methods involve two steps. First, univariate or multivariate EEG features are extracted from a sliding window. Then each EEG epoch in the moving window is classified as either pre-seizure or normal based on an optimized threshold level. Whenever a windowed EEG epoch is classified as pre-seizure, a warning alarm is triggered indicating that an impending seizure may occur within a pre-defined prediction horizon. Although some methods have shown promising results for selected patients, the reliability and repeatability of the results have been questioned when tested on other EEG datasets. Many of the earlier optimistic findings were irreproducible or achieved poor performance in extended EEG datasets [9]. This is not surprising since the optimal threshold obtained from a limited number of patients may not be generalizable. Manually tuning a threshold level for each individual patient is a subjective procedure and would pose a significant burden on physicians and patients. The inability to apply these techniques to a wide spectrum of epileptic patients with a variety of types of epileptic seizures may represent the greatest limitation of current seizure prediction methods.

Given our accumulated knowledge regarding seizure prediction, we conjecture that a promising approach may be the one that processes adaptive learning ability and is capable of achieving personalized seizure prediction autonomously. The flowchart of a prospective adaptive seizure prediction system is illustrated in Figure 4.1. In this study, we attempted to construct an adaptive prediction system using machine learning algorithms. We developed a novel adaptive learning approach, which combines reinforcement learning, online monitoring, and feedback control theory into an online seizure prediction system. The proposed adaptive seizure prediction approach can be

readily integrated to any clinical EEG system. With the attractive adaptive learning ability, the proposed approach is capable of achieving a personalized seizure prediction through baseline-updating as it monitors more and more EEG recordings from a patient.

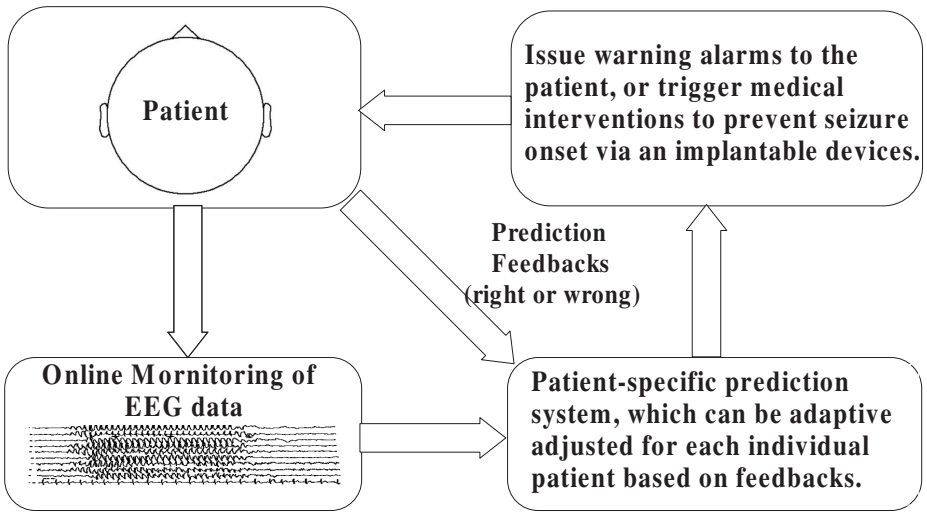


Figure 4.1: A prospective adaptive seizure prediction system, which can be adjusted to each individual patient automatically based on feedbacks.

4.3 Materials and Methods

4.3.1 Data Collection

In this study, we used a dataset containing long-term continuous intracranial EEG recordings from 10 epileptic patients with temporal lobe epilepsy. The placement of the EEG electrodes is shown in Figure 4.2, which is a modified image of the inferior transverse view of the brain from Potter [121]. The EEG recordings consist of 26 standard channels. Recording durations ranged from 3 to 13 days. Expert epileptologists annotated the EEG recordings to determine the number of seizures, their onset, and their offset points. The characteristics of the 10 patients and the EEG data statistics are outlined in Table 4.1.

Table 4.1: Characteristics of the analyzed patients and EEG data

Patient	Gender /Age	Number of Seizures	EEG Length (hour)	Average Inter-seizure Interval (hour)	Seizure Type
1	F/45	7	85.18	12.17	CP, SC
2	M/60	7	280.86	40.12	CP, GTC, SC
3	F/41	24	212.28	8.85	CP
4	M/19	17	315.23	18.54	CP, SC
5	M/33	17	286.76	16.87	CP, SC
6	M/38	9	74.60	8.29	CP, SC
7	M/44	23	146.15	6.35	CP, SC
8	M/29	19	142.32	7.49	CP, SC
9	F/37	20	276.65	13.83	CP, SC
10	M/37	12	231.61	19.30	CP, GTC
Total		155	2051.63		

Seizure types: CP, complex partial; SC, subclinical; GTC, generalized tonic/clonic.

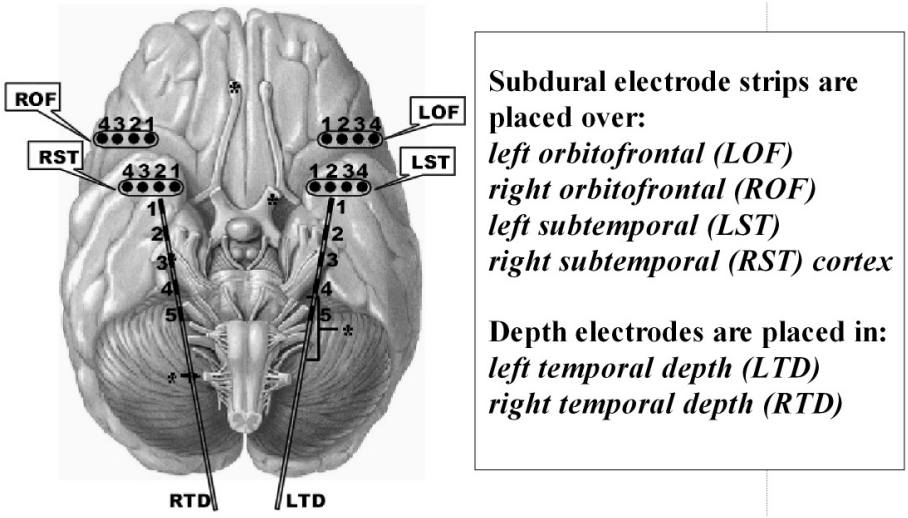


Figure 4.2: The interior transverse view of the brain and the placement of the 26 EEG electrodes.

4.3.2 Data Preprocessing & Feature Extraction

Since EEG signals are highly nonstationary and seemingly chaotic, there has been an increasing interest in analyzing EEG signals in the context of chaos theory [131]. Several commonly used chaotic measures in many recent studies include largest Lyapunov exponent [66], correlation dimension [147], Hurst exponent [28] and entropy [125]. Among these EEG measures, the Lyapunov exponent has been shown to be useful in characterizing a chaotic system [156]. Lyapunov exponents measure the degree of sensitivity to initial conditions for a dynamical system. For an n-dimensional dynamical system,

there will be n corresponding Lyapunov exponents that measure the exponential rate of divergence of the different trajectories in the phase space. If an exponent is positive, it indicates that the corresponding orbits locally defined by that exponent diverge exponentially. The magnitude of the exponents indicates the degree of divergence. The largest Lyapunov exponent in a chaotic system is usually more reliable and reproducible than the estimation of all the exponents [156], and is an important indicator to characterize a chaotic system. In our previous studies, we used an estimation algorithm called the short-term largest Lyapunov exponent (STL_{max}) to quantify EEG dynamics [66]. We employed this measure in the current study. A detailed calculation of STL_{max} as well as parameter selection and variation of STL_{max} has been explained by Iasemidis in [63].

4.3.3 Adaptive Seizure Prediction Approach

The schematic structure of the proposed adaptive seizure prediction system is illustrated in Figure 4.3. A sliding window was applied to monitor continuous multichannel EEG data. The window size is 10 minutes with 50% overlap between two successive windows. Two baselines of normal and pre-seizure states were constructed and initialized by the beginning part of the EEG recordings for each patient. The two baselines were used to classify the monitored EEG epochs of the sliding moving window using a K-nearest-neighbor (KNN) method. All the baseline samples and windowed EEG epochs were represented in terms of the multichannel time profile of STL_{max} values. The two baselines were updated by a reinforcement learning algorithm based on feedbacks of prediction actions (true or false). The adaptive seizure prediction system is discussed in detail in the following.

4.3.3.1 Baseline Construction & Initialization

To start our prediction system, we first initialize the pre-seizure and normal baseline samples. The selection of baseline samples depends on the presumed time length of pre-seizure period, which is often considered the prediction horizon in the seizure prediction

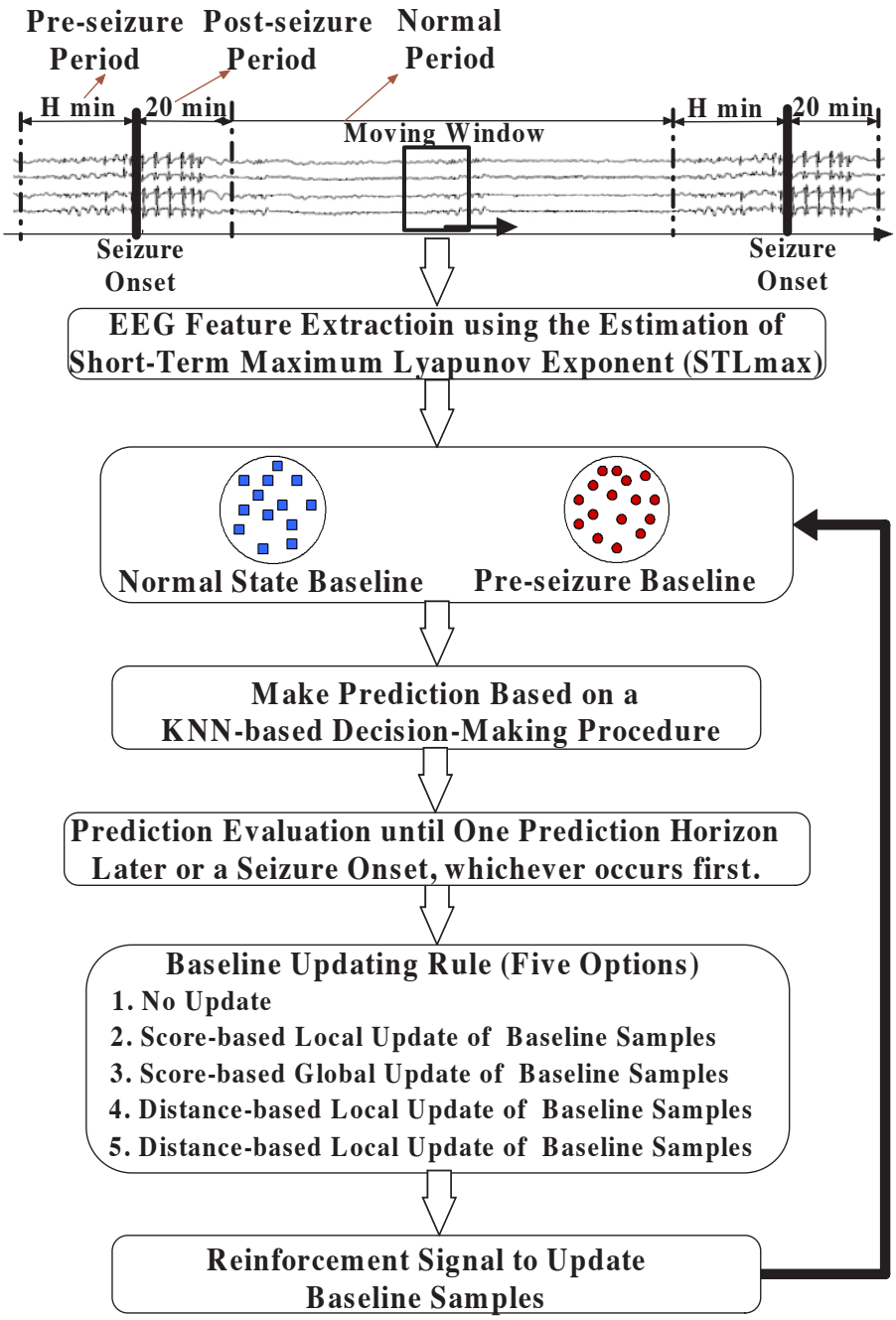


Figure 4.3: Schematic structure of the adaptive prediction system.

literature. The pre-seizure duration has been reported to be between a few minutes and several hours prior to seizure onset, and remains an open question in epilepsy research. In this study, we tried three prediction horizons (30, 90, and 150 minutes). For convenience, we denote the length of the prediction horizon as H minutes, then the EEG recordings can be divided into the following three periods:

- Pre-seizure period: 0-H minutes preceding a seizure onset.
- Post-seizure period: 0-20 minutes after a seizure onset.
- Normal period: between pre- and post-seizure periods.

The initial samples of the two baselines were randomly chosen from the normal and pre-seizure period preceding the first seizure onset. The length of the baseline samples is equal to that of the moving window. Since there are no guidelines available to determine the number of samples in each baseline, we tentatively stored a fixed number of 50 samples in each baseline.

4.3.3.2 KNN Prediction Procedure

With baselines for normal and pre-seizure states, it is intuitive to classify a windowed EEG epoch based on its degree of similarity to the two baselines. For this purpose, KNN is a reasonable choice because it classifies a new unlabeled sample by comparing the sample with all the samples in the two baseline sets. For each EEG epoch in the moving window, the KNN method finds its K nearest (best matching) samples in each baseline, and compares the its averaged distances to the two groups of K-nearest neighbors. The epoch is classified to a baseline that is 'closer' to it. The KNN prediction procedure is described in the following.

KNN methods use similarity measures to quantify the closeness between a moving-window EEG and baseline samples. We employed three frequently-used time-series similarity measures. If we denote two time-series of STL_{max} as X and Y with equal length of n, then the three types of distances are briefly described as follows.

- Euclidean distance (EU): measures the degree of similarity in terms of amplitude of the data. The EU between X and Y is defined as $ED_{xy} = \sqrt{\sum_{p=1}^{n}(x_p - y_p)^2}$.

- T-statistical distance (TS): a statistical distance measure between two time series derived from the t-test. It is frequently used to determine if the mean values of

two time series differ from each other in a significant way under the assumptions that the paired differences are independent and identically normally distributed. The TS between X and Y is calculated by $TS_{xy} = \sum_{p=1}^{n} |x_p - y_p| / \sqrt{n}\tau_{|X-Y|}$, where $\tau_{|X-Y|}$ is the sample standard deviation of the absolute difference between the time series X and Y.

- Dynamic time warping (DTW): DTW measures similarity based on the best possible alignment or the minimum mapping distance between two time series. The two time series are 'warped' in the time domain to find the optimal pattern matching between them. DTW is particularly suited to matching time series patterns independent of time variations. A detailed calculation of DTW can be found in [142].

Once a similarity measure is chosen, we can obtain the distance between a baseline sample and an EEG epoch in the moving window. For a multichannel EEG epoch, the window-sample distance is calculated as follows:

$$d_{pre,i} = \sum_{j=1}^{M} distance(S^j_{pre,i}, S^j_{mw}) \tag{4.1}$$

$$d_{int,i} = \sum_{j=1}^{M} distance(S^j_{int,i}, S^j_{mw}) \tag{4.2}$$

where M=26 is the number of EEG channels. $S^j_{pre,i}$ and $S^j_{int,i}$ is the jth channel of EEG time series in the ith pre-seizure and normal baseline sample, respectively; $S^j_{mv,i}$ is the jth channel of EEG in the windowed EEG epoch; $d_{pre,i}$ and $d_{int,i}$ are the distances between the windowed EEG and the ith sample in the pre-seizure and normal baseline, respectively. The term *distance* in the above formula represents a time series distance measure, which denotes EU, TS, or DTW in this chapter.

We used four choices of K. They were three, seven, half, and all of the baseline samples, respectively. Once K is fixed, the weighted summation of K smallest window-sample distances in a baseline was considered as the distance between the windowed EEG epoch and that baseline. Therefore, we call the two distances as window-normal

distance D_{int}^{K} and window-preseizure distance D_{pre}^{K}, respectively. For each windowed EEG epoch, its distances to the two baselines can be calculated by $D_{pre}^{K} = \sum_{k=1}^{K} \alpha_k d_{pre,k}$ and $D_{int}^{K} = \sum_{k=1}^{K} \beta_k d_{int,k}$. The α_k and β_k are the weights of the kth pre-seizure and normal baseline, respectively. The $d_{pre,k}$ and $d_{int,k}$ are the distances between the windowed EEG epoch and its kth nearest neighbor in the pre-seizure and normal baseline, respectively. Once the two baseline-window distances are obtained, the prediction decision can be made by:

$$predictor = \begin{cases} 1, & \text{if } D_{pre}^{K}/D_{int}^{K} \leq R^{*} \text{ (issue an alarm)} \\ 0, & \text{otherwise (no warning);} \end{cases}$$

where the threshold R^{*} can be used to control the sensitivity of the prediction system. In this study, we employed $R^{*} = 0.99$ to make the prediction less sensitive to noises which would lead to many false predictions.

4.3.3.3 Evaluation of a Prediction Result

Baseline updating depends on prediction evaluation feedback. We define the evaluation metrics of each prediction outcome by the following. If the predefined prediction horizon is H minutes, then we can categorize each prediction outcome into one of the following four subsets:

- True positive (TP): if $predictor = 1$ and a seizure occurs within H minutes after the prediction.
- False positive (FP): if $predictor = 1$ and no seizure occurs within H minutes after the prediction.
- True negative (TN): if $predictor = 0$ and no seizure occurs within H minutes after the prediction.
- False negative (FN): if $predictor = 0$ and a seizure occurs within H minutes after the prediction.

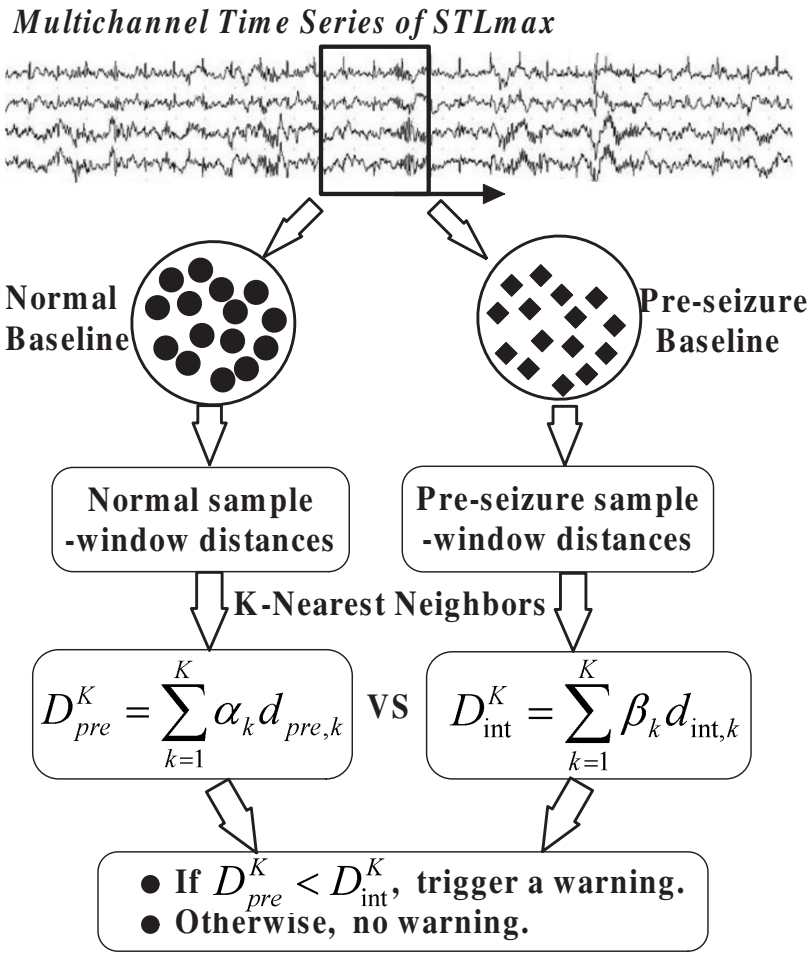


Figure 4.4: Schematic structure of the KNN-based prediction rule.

		Prediction Outcome	
		pre-seizure	*normal*
Actual	*pre-seizure*	*TP*	*FN*
	normal	*FP*	*TN*

Figure 4.5: The categorization of prediction outcomes. Each prediction outcome can always be classified into one of the four subsets (TP, FP, TN, and FN).

4.3.3.4 Baseline Updating Mechanism

The flowchart of the baseline update framework from delayed prediction feedback is shown in Figure 4.7. In medical practice, a physician mentally compares the EEG patterns from an individual with the patterns from a database of many other patients

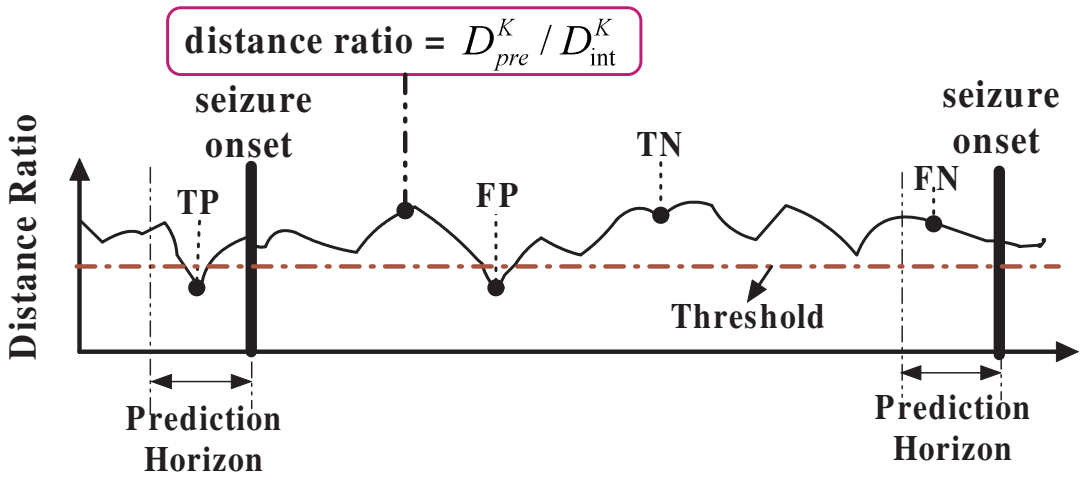


Figure 4.6: A demonstration of the evaluation metrics: TP, FP, TN, and FN.

and healthy people. The search of the best matching patterns can be global within the whole database, and can also be local within a sub-group of the database. We designed both local and global update rules inspired by this consideration. In particular, we designed four update rules including score-based local update (SL), score-based global update (SG), distance-based local update (DL), and distance-based global update (DG).

Score-Based Update: In this prediction scheme, we assume that different baseline samples have different power in decision making. We assigned a score to each baseline sample to indicate its 'importance'. The basic idea of score updating is to reinforce the scores of the 'good' baseline samples when correct predictions are made, and decrease the scores of 'bad' baseline samples when false predictions are made. The score of a baseline sample is determined by its window-sample distances. For example, if a windowed EEG epoch is mis-classified as pre-seizure via the KNN evaluation, then the pre-seizure baseline samples that are closest to, and the normal baseline samples that are furthest from, this windowed epoch will see their scores penalized according to their window-sample distances. The closest pre-seizure baseline sample and the furthest normal baseline sample receive the highest penalties. The mathematical formulations of the score updating rules are stated in the following.

At the beginning, the initial scores of the baseline sample are all equal, and are

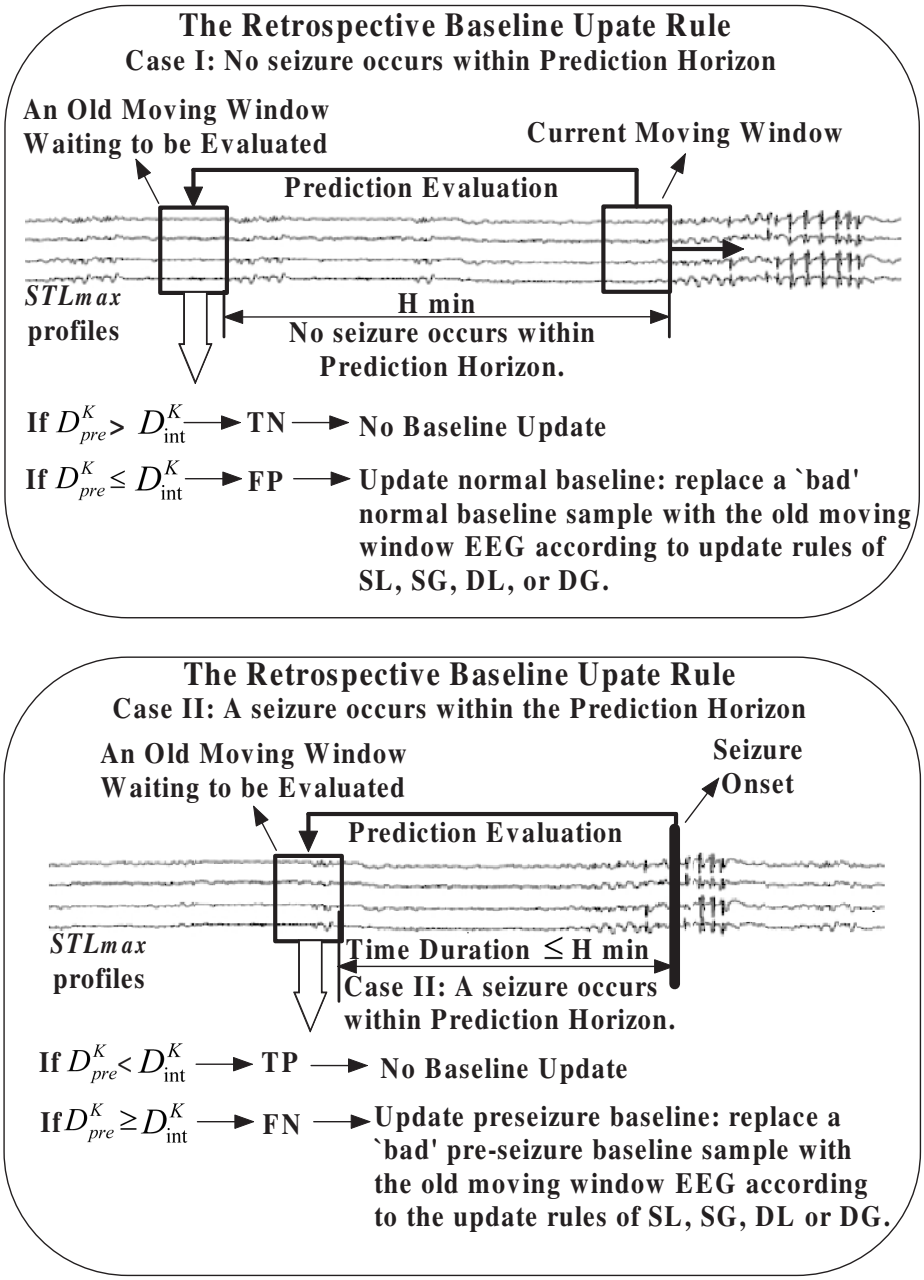


Figure 4.7: Flowchart of the retrospective baseline-updating framework.

given by:

$$\alpha_i = \beta_i = \frac{1}{N},\ i = 1, \ldots, N, \tag{4.3}$$

where α_i and β_i are the scores of the ith sample in the pre-seizure and normal baseline, respectively. $N = 50$ is the number of samples in each baseline. Let $r \in (0, 1)$ denote the learning rate to control the update size for the scores, then the score update rule is

represented as follows:

- For feedback of TP or FN (the windowed EEG is in pre-seizure period), the scores are updated by:

$$\alpha_i = \alpha_i(1 - \frac{d_{pre,i} - \overline{d}_{pre}}{\overline{d}_{pre}}) \times r, \tag{4.4}$$

$$\beta_i = \beta_i(1 + \frac{d_{int,i} - \overline{d}_{int}}{\overline{d}_{int}}) \times r, \tag{4.5}$$

- For feedback of FP or TN (the windowed EEG is in normal period), the scores are updated by:

$$\alpha_i = \alpha_i(1 + \frac{d_{pre,i} - \overline{d}_{pre}}{\overline{d}_{pre}}) \times r, \tag{4.6}$$

$$\beta_i = \beta_i(1 - \frac{d_{int,i} - \overline{d}_{int}}{\overline{d}_{int}}) \times r, \tag{4.7}$$

where $\forall i = 1, 2, \ldots, N$, $\overline{d}_{pre} = \sum_{i=1}^{N} d_{pre,i}/N$, and $\overline{d}_{int} = \sum_{i=1}^{N} d_{int,i}/N$.

For a windowed EEG epoch, the system makes a prediction by the KNN method. The feedback of this prediction is available until either of the following occurs: 1) the prediction horizon passes, or 2) a seizure occurs. Once the feedback of this prediction is given, the score-based retrospective baseline update rules are as follows:

- For case of FP: replace the lowest-scored sample in the normal K-nearest neighbors with the moving-window EEG epoch.

- For case of FN: replace the lowest-scored sample in the pre-seizure K-nearest neighbors with the moving-window EEG epoch..

- For cases of TP and TN: keep the current baseline samples unchanged.

When K equals to N, the above update is a global update rule that replaces the global lowest-scored baseline sample. When K is smaller than N, it is a local update rule which only considers the local K-nearest neighbors of a windowed EEG epoch. The

score-based local and global update rules are denoted as 'SL' and 'SG', respectively, in the remaining part of this chapter.

Distance-based Update: The distance between two EEG epochs indicates the degree of similarity. Intuitively, a shorter distance means a better match, and a larger distance indicates a worse match. For a windowed EEG epoch, the goodness of a baseline sample depends on its window-sample distances. For example, suppose a normal state windowed EEG epoch, via KNN evaluation, is falsely classified as pre-seizure. We consider the furthest normal baseline sample as the 'bad' baseline sample, which may be the primary cause of the false prediction, and we replace it with the windowed EEG epoch. In summary, for a windowed EEG epoch, the retrospective distance-based baseline update rules are as follows:

- For feedback of FP: replace the furthest sample in its K-nearest neighbors of the normal baseline with the corresponding windowed EEG epoch.
- For feedback of FN: replace the furthest sample in its K-nearest neighbors of the pre-seizure baseline with the corresponding windowed EEG epoch.
- For feedback of TP or TN: keep the current baseline samples unchanged.

Similar to 'SL' and 'SG', the distance-based update can also be local and global depending on the value of K. The distance-based local and global update rules are denoted as 'DL' and 'DG', respectively.

4.3.4 Evaluation of Prediction Performance

To evaluate a prediction model, the most commonly used performance measures are specificity and sensitivity. In seizure prediction studies, sensitivity is usually defined as the number of correctly predicted seizures divided by the total number of seizures. A seizure is considered to be correctly predicted if there is at least one warning within its preceding prediction horizon. In this study, we also employed this definition of sensitivity, denoted as sen_{blk}. To estimate the prediction specificity, most studies calculated a false prediction rate, which is defined by the number of false predictions per hour (or

unit time). However, false prediction rate does not provide enough information to infer the effect of prediction horizon on the prediction performance. For example, a patient has to wait until the end of prediction horizon to determine if a warning is false. Given the same false prediction rate, an algorithm with a 3-hour prediction horizon will give a patient much longer false awaiting time than the one with a 10-minute prediction horizon. To overcome this bias, Mormann et al. [104] suggested that a prediction specificity can be estimated by quantifying the portion of time during the normal period that is not considered to be false awaiting time. We herein employed this specificity measure, denoted as spe_{blk}. A demonstration of the sen_{blk} and spe_{blk} quantification is shown in Figure 4.8. In turn, we also define the overall prediction performance (OPP) as an average of sen_{blk} and spe_{blk}, i.e., $OPP = (sen_{blk} + spe_{blk})/2$. The OPP values can range from [0.0, 1.0]. An accurate prediction model should have an OPP close to 1, and a random model should have an OPP around 0.5. The closer the OPP value to one, the better the prediction performance.

Receiver Operating Characteristic (ROC) Analysis:

In any prediction algorithm, one can always make a trade-off between sensitivity and specificity, such as increasing sensitivity at the expense of a lower specificity. A common way to compare different prediction models is to construct a ROC curve that plots sensitivity versus (1-specificity) whereas the decision boundary of the prediction model is varied throughout its range. The *area under the ROC curve* (AUC) is commonly used to access the overall prediction power of a prediction model. AUC values are usually between 0.5 and 1. A perfect prediction model has an AUC value of 1 while a random chance model has an AUC of around 0.5.

4.4 Results

4.4.1 Computational Settings

The proposed prediction approach was tested on EEG recordings of 10 patients with epilepsy using three prediction horizons, four baseline-update rules, four settings of

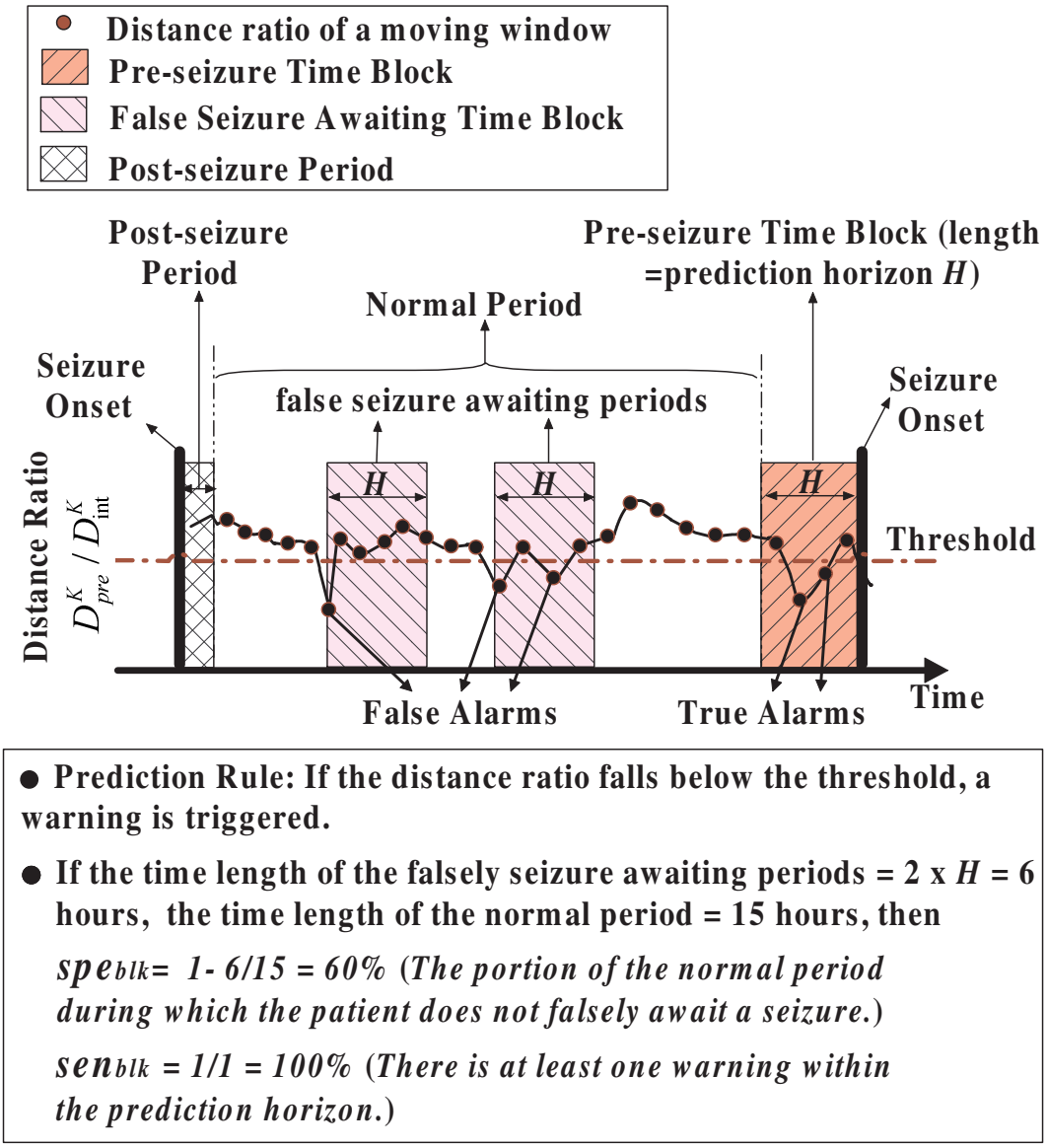


Figure 4.8: A demonstration of the prediction procedure based on the distance ratio D^K_{pre}/D^K_{int}. The definition of sensitivity (sen_{blk}), specificity (spe_{blk}), false alarms, and false seizure awaiting periods are also illustrated.

KNN, and three types of similarity measures. The summary of the parameter settings of the prediction system is shown in Table 4.2.

4.4.2 Random Predication Models

There has been no definite conclusion whether prospective algorithms can predict seizures based on EEG analysis. Before applying it to any clinical application, it is

Table 4.2: Summary of the settings of the prediction system.

Parameter Setting	Values or Choices
Moving Window	10 minutes length with 50% overlap each step
Prediction Horizon	30 minutes, 90minutes, 150minutes
Similarity Measure	EU, TS, DTW
The value of K	3, 7, half, all
Update Scheme	1. Non-update (No update to the initial baselines) 2. SL (score-based local update) 3. SG (score-based global update) 4. DL (distance-based local update) 5. DG (distance-based global update)

necessary to evaluate if the designed prediction model is indeed able to perform better than a chance model. Therefore, we compared the performance of the proposed adaptive prediction model with two random prediction schemes: periodic prediction scheme and Poisson prediction scheme. The periodic prediction scheme gives warnings at a fixed time interval T. The Poisson prediction scheme issues a warning according to an exponential distributed random time interval with a fixed mean λ. We performed the periodic prediction scheme and the Poisson prediction scheme for each patient. The values of λ and T were determined according to the average length of inter-seizure intervals for each patient as shown in Table 4.1. For example, for patient 1, the averaged inter-seizure interval is 12.17 hours, we set $\lambda = T = 12.17$ hours. This is the best value setting of T and λ the one could obtain.

4.4.3 Prediction Performance of sen_{blk} and spe_{blk}

For each patient, the EEG recordings were divided into training and testing dataset. The training dataset is the EEG recordings that contain the first half of seizure occurrences. It is used to train our approach to find the best parameter setting. The testing dataset is the EEG recordings that contain the second half of seizure occurrences. It is used to test our prediction approach ***prospectively*** using the best parameter setting found from the training data. The best parameter setting is defined as one with the highest OPP value. In addition, to find the most appropriate trade-off between sensitivity and specificity, we also added a constraint that the sen_{blk} must be greater than 0.6, and the spe_{blk} must be greater than 0.4. If none of the settings meet this

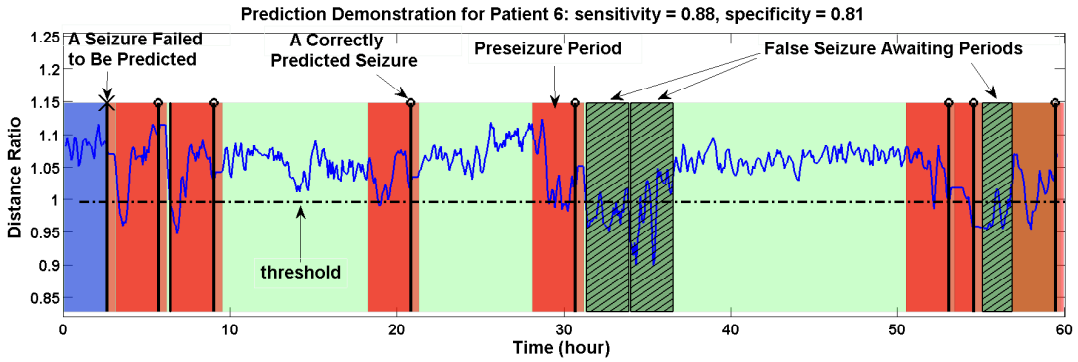


Figure 4.9: An example of the prediction outcomes of the adaptive prediction system for patient 6 using the prediction horizon of 150 minutes. Other experimental settings are SG, K =all, and DTW. The vertical black lines are the recorded seizures in this patient, and the dashed horizontal line is the threshold of distance ratio. A warning is issued if the distance ratio falls below the threshold.

constraint, we simply selected the one with the highest OPP value.

Table 4.3 summarizes the performance characteristics of the adaptive learning prediction scheme in the training and testing dataset. To determine the importance and effectiveness of the proposed baseline-update rule, we also summarizes the performance characteristics of the non-update prediction scheme in Table 4.3. The non-update prediction scheme employed the same initial baselines as the adaptive ones for each patient, and kept the baseline unchanged throughout the prediction process. Table 4.3 clearly shows that the training and testing OPP values of the adaptive learning approach are considerably higher than those of non-update prediction scheme in all the three prediction horizons. To compare with random predictions, the prediction results of the periodic and Poisson prediction schemes are also shown in Table 4.3. The adaptive learning approach performed much better than the two random prediction schemes in terms of the overall OPP values.

The adaptive prediction approach achieved the best overall performance using the prediction horizon of 150 minutes. An example of the prediction outcomes of the adaptive prediction system is also shown in Figure 4.9. In general, the averaged testing OPP over the 10 patients of the adaptive prediction approach is 0.70, which is 14%, 25%, and 27% higher than that of non-update prediction scheme, the Poisson prediction scheme, and the periodic prediction scheme, respectively. Starting from the initial (less

Table 4.3: The training and testing performance characteristics of the adaptive prediction approach and the non-update prediction scheme. The performance characteristics of the two random prediction schemes (periodic and Poisson) are also reported using $T = \lambda =$ averaged length of inter-seizure intervals for each patient.

		Adaptive Scheme					Non-Update Scheme					Possion Predictor		Periodic Predictor	
		Training			Testing		Training			Testing					
Horizon	Patient	Setting	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	Setting	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}
30minutes	1	SG-3-DTW	1.00	0.77	0.33	0.55	3-EU	0.67	0.82	0.33	0.48	0.00	0.53	0.00	0.53
	2	SG-all-TS	0.67	0.87	1.00	0.62	all-DTW	0.67	0.65	1.00	0.24	0.00	0.35	0.00	0.35
	3	DL-half-DTW	0.54	0.79	0.50	0.69	half-TS	0.69	0.43	0.20	0.75	0.17	0.52	0.17	0.52
	4	SG-3-TS	0.38	0.95	0.00	0.97	3-TS	0.13	0.93	0.14	0.91	0.00	0.72	0.07	0.72
	5	SG-3-DTW	0.63	0.60	0.25	0.74	3-DTW	0.88	0.25	1.00	0.31	0.00	0.39	0.00	0.39
	6	DG-half-DTW	1.00	0.73	0.75	0.86	all-EU	1.00	0.48	0.50	0.41	0.13	0.34	0.00	0.34
	7	SL-7-DTW	0.70	0.71	0.44	0.73	all-TS	0.70	0.63	0.56	0.55	0.05	0.52	0.00	0.52
	8	SL-7-EU	0.80	0.91	0.00	0.88	half-EU	0.20	0.90	0.00	1.00	0.06	0.77	0.12	0.77
	9	SG-3-DTW	0.22	0.97	0.30	0.94	3-TS	0.67	0.46	0.80	0.36	0.05	0.95	0.05	0.95
	10	SL-half-DTW	0.80	0.62	0.00	1.00	7-DTW	1.00	0.43	1.00	0.62	0.00	0.70	0.09	0.70
		Ave.	**0.62**	**0.79**	**0.31**	**0.80**		**0.62**	**0.59**	**0.54**	**0.50**	**0.06**	**0.96**	**0.06**	**0.96**
		PA	**0.71**		**0.56**			**0.61**		**0.52**		**0.51**		**0.51**	
90minutes	1	DG-3-DTW	1.00	0.71	1.00	0.30	3-DTW	1.00	0.60	1.00	0.37	0.00	0.47	0.33	0.47
	2	DG-3-TS	0.67	0.46	1.00	0.75	3-EU	0.00	1.00	0.67	0.57	0.00	0.17	0.00	0.17
	3	SG-7-EU	0.77	0.67	0.60	0.35	7-EU	0.85	0.47	0.30	0.73	0.30	0.15	0.39	0.15
	4	SL-7-TS	0.63	0.69	0.71	0.71	half-EU	0.75	0.45	0.43	0.32	0.07	0.39	0.07	0.39
	5	SL-7-EU	0.63	0.96	0.00	0.86	3-DTW	1.00	0.12	1.00	0.12	0.00	0.15	0.00	0.15
	6	DG-3-TS	0.50	0.89	0.75	0.76	3-DTW	1.00	0.23	0.50	0.29	0.25	0.00	0.25	0.00
	7	DL-3-DTW	0.80	0.40	0.89	0.34	3-DTW	1.00	0.14	1.00	0.27	0.21	0.22	0.21	0.22
	8	DG-7-EU	0.90	0.96	0.71	0.64	3-DTW	0.40	0.82	0.00	0.92	0.18	0.57	0.18	0.57
	9	DL-7-TS	0.78	0.57	0.20	0.85	all-DTW	0.67	0.45	0.90	0.36	0.11	0.79	0.16	0.79
	10	SL-7-DTW	0.80	0.43	0.67	0.71	half-EU	1.00	0.28	1.00	0.74	0.18	0.24	0.18	0.24
		Ave.	**0.75**	**0.68**	**0.58**	**0.71**		**0.78**	**0.38**	**0.67**	**0.43**	**0.15**	**0.88**	**0.19**	**0.88**
		PA	**0.72**		**0.65**			**0.58**		**0.55**		**0.52**		**0.54**	
150minutes	1	DL-half-DTW	1.00	0.81	1.00	0.40	all-TS	1.00	0.54	1.00	0.40	0.33	0.56	0.33	0.56
	2	DG-7-EU	0.67	0.53	0.67	0.84	3-TS	0.00	1.00	0.67	0.87	0.00	0.10	0.00	0.10
	3	DL-half-3	0.92	0.74	0.80	0.45	7-DTW	0.85	0.74	0.60	0.64	0.70	0.25	0.65	0.25
	4	DL-3-TS	0.63	0.66	0.43	0.82	3-DTW	1.00	0.18	0.71	0.18	0.13	0.25	0.13	0.25
	5	DG-7-EU	0.63	0.86	0.63	0.59	3-TS	1.00	0.15	1.00	0.12	0.13	0.09	0.00	0.09
	6	SG-all-DTW	0.75	1.00	1.00	0.75	7-DTW	0.75	0.72	1.00	0.84	0.13	0.17	0.13	0.17
	7	DG-3-EU	0.60	0.65	0.89	0.58	7-TS	0.60	0.47	0.89	0.25	0.37	0.19	0.37	0.19
	8	DL-3-EU	0.90	0.92	0.57	0.65	all-DTW	0.50	0.64	0.00	0.93	0.24	0.47	0.24	0.47
	9	DL-3-DTW	0.78	0.54	0.60	0.56	3-DTW	1.00	0.22	1.00	0.13	0.32	0.49	0.26	0.49
	10	DL-7-DTW	0.60	0.55	1.00	0.41	3-DTW	1.00	0.14	1.00	0.25	0.09	0.25	0.27	0.25
		Ave.	**0.75**	**0.69**	**0.73**	**0.67**		**0.79**	**0.28**	**0.78**	**0.45**	**0.29**	**0.82**	**0.28**	**0.82**
		PA	**0.72**		**0.70**			**0.54**		**0.62**		**0.56**		**0.55**	

representative) baseline samples, the adaptive system increased the prediction performance considerably by baseline-updating for each individual patient. The experimental results confirmed our goal that it is possible to achieve personalized prediction through adaptive learning approaches. In addition, one can observe an increasing trend of the averaged OPP values for both adaptive and non-update prediction schemes when the prediction horizon increases from 30 minutes to 150 minutes. This may indicate that the prediction horizon of 150 minutes is a better estimate of the real length of pre-seizure periods. The length of prediction horizon is very crucial since a better estimate of pre-seizure periods will give better reinforcement feedbacks to the adaptive learning system, and thus will lead to a better prediction performance.

4.4.4 Receiver Operating Characteristic Analysis

The effectiveness of the proposed four adaptive prediction schemes was also evaluated by the ROC analysis. Table 4.4 summarizes the AUC values of the four adaptive schemes (SL, SD, DL, and DG), the non-update scheme, and the two random schemes (periodic and the Poisson). The four adaptive schemes and the non-update scheme employed the best parameter settings obtained from the training data of each patient. For each prediction scheme with a selected setting, the sensitivity and specificity of the entire EEG recordings of a patient were used to generate ROC curves. The parameter used to generate ROC curves is the threshold of the distance ratio R^*, which was tuned from 0.1 to 10 to make a broad spectrum of tradeoff between sensitivity and specificity. For the periodic and Poisson schemes, the sensitivity and specificity tradeoff is controlled by the parameters T and λ, respectively. The ROC curves were obtained by tuning T and λ from 0.1 to 20 hours. We performed 300 Monte Carlo simulations for both random schemes, a set of λ and T were randomly, uniformly selected from [0.1, 20] hours at each experiment. The averaged AUC values over 300 experiments are reported in Table 4.4.

One can clearly observe that the four adaptive schemes generally have higher AUC values than the non-update and the two random schemes. When using the prediction

Table 4.4: AUC Comparison of the four adaptive prediction schemes with the non-update and the two random prediction schemes. The four adaptive prediction schemes and the non-update prediction scheme employed the best parameter settings using the training data set. Their ROC curves were obtained by tuning the threshold of distance ratio R^* from 0.1 to 10 to make a broad spectrum of tradeoff between sensitivity and specificity. For the periodic and Poisson prediction schemes, the ROC curves were obtained by tuning λ and T from 0.1 to 20 hours for each patient. We performed 300 Monte Carlo simulations for both random prediction schemes, a set of λ and T were randomly, uniformly selected from [0.1, 20] hours at each experiment. The averaged AUC values over the 300 experiments are reported in this table.

		SL		SG		DL		DG		None		Poisson	Periodic
Horizon	Patient	Setting	AUC	Setting	AUC	Setting	AUC	Setting	AUC	Setting	AUC	AUC	AUC
30 minutes	1	all-DTW	0.73	3-DTW	0.8	3-DTW	0.73	all-TS	0.77	3-EU	0.8	0.52	0.51
	2	all-TS	0.73	all-TS	0.73	half-DTW	0.71	half-DTW	0.79	all-DTW	0.35	0.51	0.51
	3	half-TS	0.66	3-TS	0.59	half-DTW	0.64	3-EU	0.7	half-TS	0.52	0.56	0.54
	4	3-TS	0.69	7-DTW	0.61	7-TS	0.61	7-EU	0.64	3-TS	0.59	0.51	0.50
	5	7-DTW	0.57	3-DTW	0.57	all-EU	0.59	3-DTW	0.54	3-DTW	0.64	0.52	0.51
	6	3-DTW	0.62	half-DTW	0.74	half-DTW	0.68	half-DTW	0.85	all-EU	0.53	0.49	0.52
	7	7-DTW	0.7	half-DTW	0.68	half-DTW	0.66	7-EU	0.67	all-TS	0.65	0.53	0.51
	8	7-EU	0.78	7-TS	0.75	3-TS	0.81	half-EU	0.82	half-EU	0.61	0.49	0.50
	9	all-DTW	0.69	3-DTW	0.73	half-TS	0.73	3-EU	0.63	3-TS	0.6	0.51	0.51
	10	half-DTW	0.67	3-DTW	0.63	half-EU	0.7	3-EU	0.74	7-DTW	0.81	0.50	0.52
	Ave.		**0.68**		**0.68**		**0.69**		**0.72**		**0.61**	**0.51**	**0.51**
90 minutes	1	half-TS	0.76	half-EU	0.79	half-TS	0.76	3-DTW	0.84	3-DTW	0.75	0.51	0.54
	2	half-DTW	0.37	3-TS	0.66	7-TS	0.52	3-TS	0.7	half-DTW	0.33	0.49	0.48
	3	half-DTW	0.6	7-EU	0.66	3-TS	0.66	3-TS	0.64	7-EU	0.61	0.62	0.60
	4	7-TS	0.7	3-TS	0.59	3-DTW	0.62	7-DTW	0.53	half-EU	0.59	0.52	0.52
	5	7-EU	0.69	3-EU	0.63	half-EU	0.67	3-DTW	0.56	3-DTW	0.55	0.52	0.52
	6	half-DTW	0.67	all-TS	0.58	all-TS	0.6	3-TS	0.75	3-DTW	0.65	0.54	0.57
	7	7-EU	0.61	3-DTW	0.47	3-DTW	0.6	half-DTW	0.57	3-DTW	0.5	0.57	0.55
	8	3-EU	0.59	3-EU	0.64	half-TS	0.78	7-EU	0.89	3-DTW	0.49	0.51	0.50
	9	half-DTW	0.57	7-DTW	0.67	7-TS	0.57	7-TS	0.65	all-DTW	0.6	0.50	0.53
	10	7-DTW	0.64	all-EU	0.78	7-DTW	0.6	7-DTW	0.58	half-EU	0.52	0.52	0.52
	Ave.		**0.62**		**0.65**		**0.64**		**0.67**		**0.56**	**0.53**	**0.53**
150 minutes	1	half-TS	0.76	7-EU	0.9	half-DTW	0.93	half-TS	0.77	all-TS	0.79	0.56	0.55
	2	3-EU	0.77	7-EU	0.59	3-TS	0.66	7-EU	0.77	all-DTW	0.3	0.46	0.45
	3	half-EU	0.62	7-DTW	0.73	half-DTW	0.73	all-DTW	0.74	7-DTW	0.79	0.62	0.62
	4	7-TS	0.65	7-EU	0.5	3-TS	0.64	3-EU	0.65	3-DTW	0.54	0.54	0.54
	5	7-EU	0.62	3-TS	0.62	half-EU	0.69	7-EU	0.71	3-TS	0.56	0.50	0.51
	6	all-DTW	0.75	all-DTW	0.75	7-DTW	0.76	half-DTW	0.75	3-DTW	0.84	0.54	0.51
	7	7-DTW	0.47	7-DTW	0.49	half-DTW	0.67	3-EU	0.65	7-TS	0.5	0.60	0.59
	8	7-TS	0.86	3-EU	0.82	3-EU	0.84	3-EU	0.83	all-DTW	0.49	0.50	0.50
	9	7-TS	0.68	3-DTW	0.62	3-DTW	0.61	3-DTW	0.6	3-DTW	0.6	0.54	0.54
	10	3-DTW	0.53	3-EU	0.82	7-DTW	0.69	3-EU	0.62	3-DTW	0.52	0.54	0.54
	Ave.		**0.67**		**0.68**		**0.72**		**0.71**		**0.59**	**0.54**	**0.54**

Table 4.5: Averaged AUC values over the 10 patients for all settings of each prediction scheme. The boxplot of these averaged AUC values for all the prediction schemes are shown in Figure 4.10.

		Non-Update				SL				SG				DL				DG			
	setting	all	3	7	half	all	3	7	half	all	3	7	half	all	3	7	half	all	3	7	half
30 minutes	EU	0.58	0.61	0.63	0.64	0.65	0.61	0.65	0.64	0.65	0.66	0.67	0.64	0.67	0.63	0.64	0.66	0.67	0.70	0.67	0.64
	TS	0.57	0.59	0.59	0.59	0.61	0.62	0.61	0.63	0.61	0.62	0.62	0.62	0.64	0.64	0.59	0.66	0.64	0.65	0.63	0.64
	DTW	0.58	0.60	0.61	0.64	0.66	0.61	0.63	0.63	0.66	0.66	0.67	0.68	0.67	0.61	0.63	0.68	0.67	0.65	0.65	0.69
	Ave.	**0.58**	**0.60**	**0.61**	**0.62**	**0.64**	**0.62**	**0.63**	**0.63**	**0.64**	**0.65**	**0.65**	**0.65**	**0.66**	**0.63**	**0.62**	**0.67**	**0.66**	**0.67**	**0.65**	**0.66**
90 minutes	EU	0.55	0.56	0.55	0.55	0.60	0.52	0.58	0.62	0.60	0.58	0.59	0.63	0.60	0.62	0.59	0.63	0.60	0.65	0.63	0.60
	TS	0.56	0.57	0.57	0.55	0.53	0.57	0.60	0.57	0.53	0.63	0.61	0.55	0.59	0.58	0.58	0.60	0.59	0.62	0.64	0.61
	DTW	0.55	0.55	0.55	0.54	0.56	0.61	0.61	0.58	0.56	0.61	0.60	0.56	0.61	0.63	0.63	0.64	0.61	0.64	0.64	0.61
	Ave.	**0.55**	**0.56**	**0.56**	**0.55**	**0.56**	**0.57**	**0.59**	**0.59**	**0.56**	**0.61**	**0.60**	**0.58**	**0.60**	**0.61**	**0.60**	**0.62**	**0.60**	**0.63**	**0.63**	**0.61**
150 minutes	EU	0.55	0.55	0.55	0.57	0.54	0.61	0.58	0.62	0.54	0.66	0.61	0.61	0.58	0.65	0.61	0.63	0.58	0.68	0.65	0.64
	TS	0.55	0.58	0.55	0.55	0.51	0.62	0.66	0.58	0.51	0.61	0.58	0.54	0.59	0.64	0.59	0.63	0.59	0.64	0.65	0.61
	DTW	0.57	0.58	0.58	0.57	0.52	0.65	0.63	0.55	0.52	0.66	0.59	0.56	0.60	0.65	0.67	0.66	0.60	0.67	0.62	0.61
	Ave.	**0.56**	**0.57**	**0.56**	**0.56**	**0.52**	**0.63**	**0.62**	**0.58**	**0.52**	**0.64**	**0.59**	**0.57**	**0.59**	**0.64**	**0.62**	**0.64**	**0.59**	**0.66**	**0.64**	**0.62**

horizons of 150 minutes, the averaged AUC values of the four adaptive schemes (SL, SG, DL, and DG) are 0.67, 0.68, 0.72, 0.71, respectively. The averaged AUC values of SL, SG, DL, and DG are 14%, 15%, 22%, and 20% higher than the averaged AUC value of the non-update scheme. This indicates that all the proposed four adaptive prediction schemes increased the overall prediction performance of the system through adaptive baseline-updating. When compared to the random schemes, the averaged AUC values of SL, SG, DL, and DG are 24%, 26%, 33%, and 31% higher than the averaged AUC values of the Periodic and Poisson scheme (both are 0.54). The significant higher AUC values strongly indicate that the adaptive prediction schemes has a much higher prediction power than random predictions. Similar results can also be obtained when using the prediction horizons of 30 minutes and 90 minutes.

To make a solid statistical comparison, it is also interesting to investigate the performance of the four adaptive schemes as well as the non-update scheme on all the parameter settings over the 10 patients. For each scheme (adaptive and non-update), there are 36 settings including four choices of K, three choices of distance measures, and three choices of prediction horizons. Figure 4.10 shows the boxplots of the averaged AUC values over 10 patients for the entire 36 settings of each scheme. The AUC values of the two random schemes obtained from 300 Monte Carlo simulations are shown in Figure 4.10 for comparison. The boxplot clearly shows that the AUC values of the four proposed adaptive prediction schemes have significantly different distributions with those of the non-update and random schemes. We used the AUC values of the non-update scheme as the baseline group, and performed paired t-test for the AUC values of the four adaptive schemes and the two random schemes. As shown in the Figure 4.10, the p-value of each paired t-test is smaller than 0.001. This outcome indicates that the four adaptive prediction schemes all performed significantly better than the non-update scheme. While the non-update scheme performed significantly better than the two random schemes. This is not unexpected, since the initial baseline samples employed by the non-update scheme already contained some useful information of the preseizure and normal EEG patterns. It thus worked better than random predictions.

When we compare among the four proposed adaptive schemes, we found that the

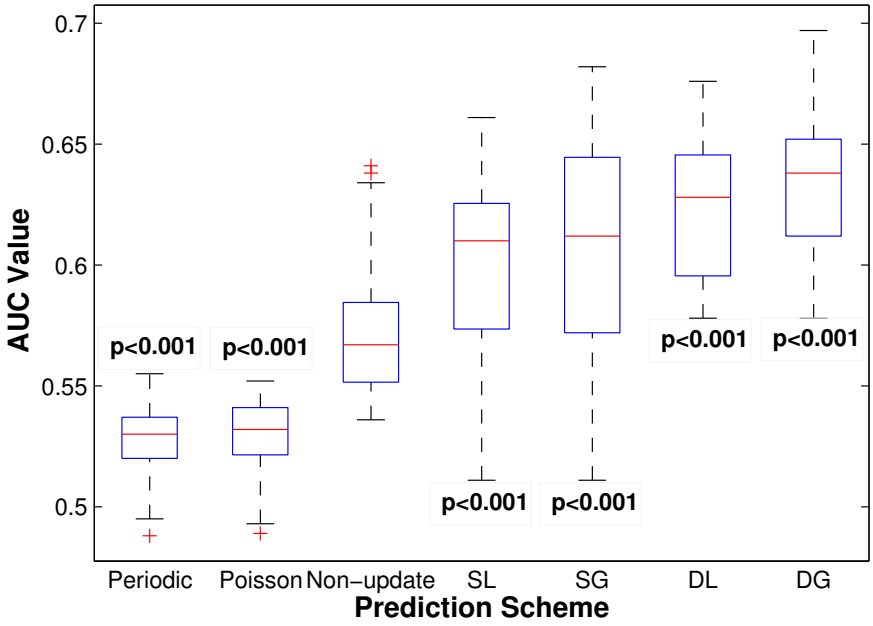


Figure 4.10: Box-plot of the AUC values of the four adaptive schemes, the non-update scheme, and the two random schemes. The AUC values of the adaptive and non-update schemes are the averaged AUC values over 10 patients for all possible parameter settings (=36) of each scheme. The AUC values of the two random schemes are obtained from 300 Monte Carlo simulations, in each of which a set of values of *lambda* and T are randomly and uniformly varied from 0.1 to 20 hours. Each box shows the median, interquartile range, minimum and maximum of the AUC values of each prediction scheme. Using AUC values of the nondicated-update scheme as the baseline group, the p-values of the paired t-tests for the AUC values of other prediction scheme are indicated in the plot. The four adaptive schemes performed significantly better than the non-update scheme with all p-values smaller than 0.001. While the non-update scheme performed significantly better than the two random schemes with both p-values smaller than 0.001.

two distance-based update schemes (DL and DG) performed better than the two score-based update schemes with p-values smaller than 0.001. This outcome implies that the distance-based update rule did a better job in the online baseline-updating than the score-based rule. In addition, the AUC values of the two score-based update schemes SL and SG are comparable with a p-value of 0.15; and DG worked a little better than DL with a p-value of 0.02.

4.4.5 Comparisons to Other Seizure Prediction Methods

There have been many studies focusing on epileptic seizure prediction. However, only a few of them were designed prospectively for online seizure prediction. Since most studies employed different EEG datasets and the ambiguous performance measure of false prediction rate, it is actually very hard to compare the real prediction performances between these algorithms. With recognition of this problem, the seizure prediction researchers began to report more universal and unambiguous performance measures, such as the portion of time a patient is not in the false awaiting state suggested by [104]. Two recent studies have reported this information and thus are convenient to be compared with our approach. Sackellares et al. evaluated an adaptive seizure prediction approach on 10 patients. Given a prediction horizon of 150 minutes and a sensitivity of 80%, the portion of false awaiting time is 37% (corresponding to our specificity of 63%) on average over the 10 patients. Snyder et al. [148] performed a prospective seizure prediction on four patients using a prediction horizon of 120 minutes. The averaged sensitivity is 82.3% and the portion of false awaiting time is 30.5% (corresponding to our specificity of 69.5%). The OPP values of the two studies are 0.72 and 0.76, respectively. For our approach, if we choose the prediction horizon of 150 minutes and select the best prediction performance based on the entire EEG recordings of each patient, the resulting sensitivity is 77% and specificity is 73% on average over the 10 patients. The OPP value of our adaptive learning approach is 0.75. Given comparable prediction performance to the two state-of-the-art studies, our adaptive learning approach is actually more prominent since it does not require a sophisticated parameter/threshold optimization procedure and is capable of improving prediction performance autonomously in the online monitoring and prediction process. The prediction performance of the adaptive learning approach has more potential to be further improved when more EEG recordings are available. In addition, only requiring the first seizure of each patient for initialization, the proposed approach is more convenient to be embedded into the existing EEG systems and achieve a personalized prediction using adaptive learning.

4.5 Conclusions and Discussion

This study investigated the challenging problem of epileptic seizure prediction. We introduced an adaptive learning approach, which combine reinforcement learning, online monitoring and adaptive control theory to achieve a personalized seizure prediction. Using EEG recordings from 10 patients with epilepsy, we demonstrated that the adaptive learning algorithm was effective in increasing prediction performance of the system through adaptive baseline-updating. The best prediction performance was achieved using the prediction horizon of 150 minutes, in which the averaged sensitivity was 73% and the averaged specificity was 67%. The ROC analysis demonstrated that the adaptive prediction schemes indeed performed much better than the non-update scheme and the two chance models.

The experimental outcomes of this study are very encouraging given that seizure prediction techniques are still in their early stages. There has been no definite conclusion that the current prospective prediction algorithms are indeed able to perform better a random prediction [104]. This study confirmed the hypothesis that it is possible to prospectively predict impending seizures based on the proposed adaptive learning algorithm. An autonomous learning framework like the one proposed here was shown capable of self-adjusting the baseline samples for each individual patient without a tedious parameter tuning process. With this attractive online learning ability, the proposed adaptive learning prediction system is expected to be able to further improve the prediction performance when more EEG recordings are available for each patient.

The proposed adaptive learning approach is a pilot framework that can be potentially applied to a wide range of patients with epilepsy, and achieve a personalized seizure perdition for each individual patient through adaptive learning. In practice, a prospective seizure prediction system must have both high sensitivity and specificity for clinical use. If such a seizure-warning device is to become a reality, we envision that adaptive learning techniques will definitely play an important role in handling the great variety of brain-wave patterns among different patients.

Chapter 5

Robust and Efficient Approaches for Offline and Online Time Series Segmentation

A time series data set usually has large data size, high dimensionality, and incrementally updates over time. One of the fundamental problems in time series data mining is how to represent time series data efficiently in a robust and fast way. In the last decade, there has been an explosion of research interest on time series representations to manipulate large volumes of raw time series data. Piecewise linear approximation (PLA) is one of the most frequently representations. There havebeen various PLA approaches developed. However, most of them employ some data-dependent thresholds, which require a careful tuning process to fit different time series data. Thus the resulting approximation performances highly rely on a user's knowledge of the data. In addition, the approaches using heuristic data-dependent thresholds are not robust to noises and outliers, which are inevitable in most real time series data. In this chapter, we first propose a new data-independent threshold strategy developed from statistics theory. Based on the new threshold strategy, we develop a two-stage offline segmentation algorithm that gets rid of a tedious parameter tuning process for various time series data. Finally, we extend the offline algorithm into an efficient online time series segmentation algorithm using a set of incremental closed-form formulas. The proposed offline and online time series segmentation methods have been tested on a variety of real-world time series. The online method achieved a superior overall performance over two popular online approaches in terms of approximation accuracy, compression rate, and computational efficiency.

5.1 Introduction

Time series is an important form of data with ubiquitous applications in science, engineering, manufacturing, finance, and many other fields. A time series is a sequence of data points collected chronologically. Examples of time series include daily closing prices of mutual funds and stocks, monthly sales totals, and a patient's electrocardiogram.

With the great advances in data collection and storage technologies, huge amounts of time series are generated every day. For example, data providers such as Bloomberg, Reuters and Thompson Financial offer large streams of data taken in real time from international electronic trading systems. In particular, there are nearly 200,000 listed options in the US equity and index options markets. At every second, the prices of the underlying equities change, these options are re-priced with over 400,000 updates per second and still growing [61]. The data are collected over days, months, and years, and thus generate massive amounts of data tuples that exceeds human capabilities by far. Similar problems also exist in diverse domains, which produce huge amounts of time series over time.

To efficiently manipulate massive time series data, it is necessary to represent raw time series data in a high-level representation with low dimensionality. The time series characteristics such as the trends, shapes and patterns can be compressed into a compact high-level representation. Thus, an appropriate choice of time series representation is of great importance in most time series knowledge discovery problems.

Traditionally, autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models are widely used in time series data mining. These models employ multivariate regression and represent a time series by the regression coefficients. However, the biggest problem of these models is that they are established on the stationary assumption, which requires that the analyzed time series have time-invariant statistics and that the prediction errors are white noises. Due to this restriction, they are inappropriate to analyze a large portion of time series that are non-stationary.

On the other hand, many time series segmentation methods have been proposed to process various time series data. The basic idea is to partition a time series into segments, and then approximates each time series segment by a math function, such as linear, polynomial functions. Although a time series segment can be approximated by polynomials of any degree, the PLA-based approaches are still the most often used representation in the literature according to Keogh et al. [74]. A PLA approach is to represent a time series by a series of line segments by connecting a set of key turning points. The PLA framework is popular because it is the most intuitive way to represent important time series temporal patterns, such as up and down trends and at what rates (the slopes).

Although many PLA algorithms have been proposed, most of them highly rely on some data-dependent threshold strategies, which have to be manually adjusted to fit for different time series. The resulting segmentation methods are not robust to time series noises and outliers, and cannot generate reliable results for highly non-stationary time series with time-varying statistics.

In this chapter, we aim to tackle this problem with effective solutions. In particular, we made three contributions to achieve a robust and efficient segmentation for various stationary and non-stationary time series data. The three contributions are summarized as follows:

- We introduce a data-independent threshold strategy, which defines the approximation accuracy requirement directly. The proposed new threshold strategy employs a relative statistic measure defined on $[-\infty, 1]$. If the measure is close to 1, it means a high accuracy requirement for local linear approximation; a lower value indicates a bigger error tolerance and thus leads to a coarser segmentation. The proposed threshold strategy is a scaled universal measure, which controls the degree of approximation accuracy directly and data-independent.

- We develop a new two-stage offline time series segmentation approach using the new data-independent threshold strategy. The proposed offline approach has two key steps. In the first step, it adopts a top-down decomposition structure, and

partitions a time series into non-overlapping intervals by the key turning points of the time series. The time series sequence within each interval approximately follows a linear trend (satisfies the linear approximation accuracy requirement). In the second step, we perform linear regression on each interval of time series, and fine-tune the approximation mode by the regression lines. The proposed offline time series segmentation approach is capable of achieving an accurate approximation for various time series without a tedious threshold turning process. For example, one can use the same parameter setting (threshold value) to a financial time series and an electrocardiogram time series given the same degree of approximation accuracy requirement.

- We extend the two-stage offline segmentation approach into an online algorithm, which integrates the offline segmentation algorithm into an adaptive sliding window approach. Most importantly, we formulate the online monitoring and segmentation decision process of time series data into incremental closed-form formulas. Instead of manipulate massive historical time series data, the online algorithm only needs to manipulate three online incremental variables and two approximation parameters to decide whether to perform a segmentation within the current window. With the closed-form formulations, the complexity of online processing an incoming data point is only $O(1)$. This impressive property of our online algorithm makes it possible to process massive time series streams. It achieved superior overall performance over two popular online approaches in our numerical experiments on various real-world time series.

The rest of the chapter is organized as follows. Section 1 briefly introduces the important related work on time series segmentation techniques. In Section 2, we propose a novel two-stage offline time series segmentation algorithm using data-independent threshold strategy. In Section 3, we present the new online time series segmentation framework. In Section 4, extensive experiments on various real-world time series data are performed to evaluate the proposed offline and online time series segmentation algorithms. Finally, we conclude this chapter in Section 5.

5.2 Related Work on Time Series Representation Approaches

5.2.1 Pattern Representations Methods of Time Series Data

Given various different time series representations, each has its merits and limits due to its intrinsic approximation principles. We briefly summarize the most popular time series representation approaches in the following.

5.2.1.1 Discrete Fourier Transform (DFT)

The basic idea of Fourier decomposition is that any signal, no matter how complex, can be represented by the super position of a finite number of sine/cosine waves, where each wave is represented by a single complex number known as a Fourier coefficient. A time series represented in this way is said to be in the frequency domain. A time series signal of length n can be decomposed into $n/2$ sine/cosine waves that can rebuilt the original signal. Since many of the Fourier coefficients have very low amplitude and thus contribute little to reconstructed signal. These low amplitude coefficients can be discarded without much loss of information thereby largely reduce the dimentionality of the ordinal time series data.

5.2.1.2 Discrete Wavelet Transform (DWT)

Wavelets are mathematical functions that represent a time series data in terms of the sum and difference of a template pattern, which is called mother wavelet. In this sense, they are similar to DFT, the only difference is that it replaces the sine/cosine waves by a user selected other wavelet. However, one important difference is that wavelets are localized in time, i.e. some of the wavelet coefficients represent small, local subsections of the data being studied. This is in contrast to Fourier coefficients that always represent the global contribution to the whole time series data. This property of wavelet analysis is very useful for multiresolution analysis of time series data. The first few coefficients contain an overall, coarse approximation of the time series; addition coefficients can be imagined as finer approximation to local areas. There has been an

explosion of interest in using wavelets for data compression, filtering, analysis, and other areas where Fourier methods have previously been used. For example, Chan and Fu [17] produced a breakthrough for time series indexing based on a simple, but powerful type of wavelet known as the Haar Wavelet.

5.2.1.3 Singular Value Decomposition (SVD)

SVD is a global transformation method, which is the optimal linear transform that minimizes reconstruction error. The entire time series data is examined and is then rotated such that the first axis has the maximum possible variance, the second axis has the maximum possible variance orthogonal to the first, the third axis has the maximum possible variance orthogonal to the first two, etc. Then one can only select the first few time series with largest variances for analysis, and the remaining can be discarded.

5.2.2 Symbolic Aggregate Approximation (SAX)

The symbolic representation SAX for time series was introduced by Lin et al. in [95]. This approach has been shown to be able to preserve meaningful information from the original time series data and produce competitive results for classifying and clustering time series.

The basic idea of SAX is to convert the data into a discrete format, with a small alphabet size. In this case, every part of the representation contributes about the same amount of information about the shape of the time series. To convert a time series into symbols, it is first normalized, and two steps of discretization will be performed.

- First, a time series T of length n is divided into w equal-sized segments; the values in each segment are then approximated and replaced by a single coefficient, which is their average of the data points in each segment.

- Second, determine the breakpoints that divide the distribution space into a equiprobable regions; each region has a representing symbol such as A, B, C, etc. By this way, the ordinal time series can be represented a series of symbols in a reduced

dimension, such as a string 'ABBBDCCCFFAC', and each symbol is equiprobable in probability.

The SAX method can roughly preserve the general shape of the time series with large dimensionality reduction. Another advantage of this kind of methods is that, the symbolic representation allows the use of algorithms that are not well defined for real-valued data, including suffix trees, hashing, Markov models etc.

5.2.2.1 Piecewise Segmentation

Segmentation is often used dimensionality reduction algorithm for time series data. Although the segments created could be polynomials of an arbitrary degree, the most common representation of the segments is Piecewise Linear Approximation (PLA). The idea of using PLA to approximate time series dates back to 1970s by Pavlidis and Horowitz [116]. Intuitively, a time series of length n can be represented by K straight lines. Since K is typically much smaller than n, this representation can often largely reduce the dimensionality and makes the computation of the time series data more efficient. There are many algorithms available for segmenting time series, most of them can be grouped into one of the following three categories.

- Sliding-Windows (SW): A segment is grown until it exceeds some error bound. The process repeats with the next data point not included in the newly approximated segment.
- Top-Down (TD): The time series is recursively partitioned until some stopping criteria is met.
- Bottom-Up (BU): Starting from the finest possible approximation, segments are merged until some stopping criteria are met.

An open question of PLA is how to best choose K, the 'optimal' number of linear segments used to represent a particular time series. This problem involves a trade-off between accuracy and compactness, and there is no general solution.

5.2.2.2 An Evaluation of Time Series Representation Methods

Given various different time series representations, it is natural to ask which is best and what are the limitations of each approach. We evaluate the most popular time series representations in the following.

Discrete Fourier Transform (DFT):

- key properties: it compares a time series signal with sine and cosine signals at different frequencies, the obtained fourier coefficients are used to characterize the time series.
- limitations: only extract frequency information for the whole time series, and ignore the local time series fluctuations.

Discrete Wavelet Transform (DWT):

- key properties: it compares a time series with a mother wavelet at different locations and distortion scales, the obtained wavelet coefficients are used to characterize the time series.
- limitations: considers local fluctuations; however, usually generate many coefficients, thus not good for dimensionality reduction. Often statistics of wavelet coefficients are used as features, but many useful 'time series' information is missing by doing that. Moreover, it is not easy to choose a appropriate mother wavelet for various time series.

Singular Value Decomposition (SVD):

- key properties: it linear transforms the ordinal time series into a new space according to variance of each axis.
- limitations: the transformed time series data are still massive in volume, it has to employ other techniques to further reduce the dimensionality.

Symbolic Aggregate Approximation (SAX):

- key properties: it discretizes the amplitudes space into a number of bins, and name them as A,B,C,D,etc. (for example). Then each time point can be categorized into A, B, C, or D, and the whole time series is represented by a symbolic vector. By doing this, many symbolic data mining techniques can be used to extract important patterns.

- limitations: Useful information may be missing due to coarse discretization. Also it is not intuitive for human beings.

ARMA/ARIMA Models:

- key properties: The model consists of two parts, an autoregressive (AR) part and a moving average (MA) part. It is often referred to as ARMA(p,q), which models the regression relationship between the current value X_t with previous p values $(X_{t-p}, ..., X_{t-1})$ and q prediction errors $(\epsilon_{t-q}, ..., \epsilon_{t-1})$. The orginal time series can be represented by the regression coefficients.

- limitations: ARMA/ARIMA identification employ multivariate regression and is often difficult and time consuming. The coefficients of ARMA/ARIMA have no structural interpretation, thus they may be difficult to explain to others. Moreover, these models assumes the analyzed time series is stationary and the prediction errors are white noisy, they are not suitable to nons-stationary chaotic time series.

Piecewise Linear Approximation (PLA):

- key properties: it decomposes a time series into piecewise linear segments. The PLA approaches are very useful to deal with noisy time series data, and identify local trends efficiently. The results of PLA are intuitive and very easy to understand and interpret.

- limitations: This filed has not been well developed, and the existing techniques are mostly ad-hoc and heuristic. Some are not computational efficient and some

are not robust. There is no established benchmark approaches and guidelines in this area. Here, we propose a novel and robust approach to extract piecewise linear segments, in particular skeleton points, for time series data.

5.2.3 Time Series Segmentation Background

A comprehensive review on time series segmentation techniques can be found in Keogh et al. [74] and Fu [44]. Though there are many segmentation algorithms available for time series, most of them are constructed by one of the following three frameworks:

- Sliding-Window: : a single line fits the data points in a sliding-window. If the cost of the line approximation is less than a threshold value, the next point joins the window, and the approximation cost is recalculated. The segment in the window is grown until the cost exceeds a threshold value, and then a new sliding window is open to approximate the next segment. Some typical sliding window algorithms can be found in [101, 82, 114, 97, 45, 30].

- Top-Down: start from the whole time series, and decompose the time series into two subsequences according to a break-down criterion. The fitting error of each subsequence is calculated and compared with some stopping criterion (an error threshold). A subsequence that does not meet the stopping criterion is decomposed further into two parts. The procedure continues recursively until all the subsequences meet the stopping criteria. Some typical top-down segmentation algorithms can be found in [144, 113, 86, 91].

- Bottom-Up: a time series is partitioned into the finest possible approximation first. The lines connecting each pair of adjacent data points constitute the finest basic segments. Then the fitting error of merging each pair of adjacent segments is calculated. The two segments with the least fitting error are merged into a larger line segment. The procedure continues until the merging cost of each pair of adjacent segments exceeds an error threshold. Some important bottom-up algorithms can be found in [75, 77, 62, 112].

5.2.4 Challenges in Time Series Segmentation

No matter what framework it constructed, a segmentation algorithm is inevitably to use some threshold strategy to make a trade-off between approximation accuracy and dimensionality reduction. An ideal segmentation is expected to approximate a time series with as low fitting error as possible using as few segments as possible. As summarized in Keogh et al. [74], a segmentation problem can be formulated into one of the following ways.

- minimize the approximation error given a desired number of segments (denoted as K^*) or a desired compression ratio (denoted as R^*).
- minimize the number of segments given an error threshold (denoted as E^*_{seg}) for each segment.
- minimize the number of segments given an error threshold (denoted as E^*_{tot}) for the whole approximation.

In the literature, many choices of K^*, E^*_{seg}, and E^*_{tot} are defined in different studies and different applications. For example, Pratt and Flink [123] and Fink et al. [43] controlled the desired compression ratio by a parameter r^*. For a time series sequence $x_i, \ldots, x_j$, a segmentation breaking point x_k was be selected if it satisfies $x_k/x_i \geq r^*$ and $x_k/x_j \geq r^*$. Deng et al. [30] fixed the number of segments to decompose time series data. At each step, a segmentation breaking point is selected if it is the furthest point to the two ending points of a sub-sequence. The decomposition continues until the number of segments reaches the desired threshold. Feng et al. [41] defined a heuristic threshold value h to identify the segmentation breaking points, denoted by peaks or valleys. If a point is a peak (valley) if its distances to the two adjacent valleys (peaks) higher than the threshold h. Liu et al. [97] proposed a feasible space criterion to determine the segmentation breaking point. The feasible space of a time series data point is controlled by a parameter δ, which defines the vertical neighborhood space of the data point. Since the parameter δ is highly depend on the scale of data amplitude, thus the parameter has to be manfully re-adjusted to achieve best performance for different time

series. Li et al. [92] defined a threshold ϵ to control the segmentation level in a top-down structure. At each step, if the maximum vertical distance between a time series sequence and its approximation line is greater than ϵ, the time series is divided into two parts by the data point which has the maximum vertical distance to the segment line. Keogh et al. [74] applied the maximum allowed approximation error E^*_{max} in a bottom-up structure. The segmentation procedure continues until the approximation error falls below the threshold E^*_{max}. Palpanas et al. [112] employed two types of threshold strategies (relative and absolute) in an online segmentation approach. The relative threshold strategy employed a time-weighted error, which was determined by some user-defined amnesic functions. The basic idea is to decrease the weights of older data points and try to approximate the most recent time series accurately. The absolute threshold employed the maximum allowable error for the overall approximation. To deal with time series with different amplitude scales, Liu et al. [97] employed a 'maximum error percentage' (MEP) criterion, which defines the threshold percentage of 'max error' to the range of time series values. However, the segmentation performances are still very sensitive to the value of MEP. They still had to choose appropriate MEP values carefully to fit different data sets in their paper.

It is noticed that most of the current segmentation approaches highly rely on some data-specific threshold strategies, such as 'maximum overall error' and 'maximum absolute error'. Some relative threshold strategies, such as 'maximum error percentage' and 'time-weighted error', are also not easy to manipulate, especially for highly non-stationary time series. An appropriate threshold of segmentation highly relies on a user's knowledge of the data, which is often not possible in presence of massive real-world time series streams with time-varying statistics. For example, a 'maximum absolute error' of 1 may work well for a time series ranges within [-100, 100]. It may not acceptable for another time series that ranges within [-1, 1].

From the literature study, we find that most of the current segmentation approaches are incapable of controlling the approximation accuracy and the compression rate efficiently and directly. The approximation accuracy is generally controlled indirectly

through some data-specific threshold strategy, which often requires a tedious threshold tuning procedure to process various time series data in practice. In addition, we notice that the current threshold strategies are not statistically robust due to their intrinsic heuristic structures. Thus, the resulting segmentation approaches are generally not robust to time series noises, outliers, and time-varying properties. It is still an open question to determine 'optimal' levels of these segmentation thresholds for various time series data. This limitation seriously hampers the potentials of the time series segmentation techniques in practical applications.

In addition, we also noticed that the vast majority of the time series segmentation approaches are offline algorithms, only a few of them have been extended for online segmentation of time series streams with a low (linear) computational complexity [97, 112, 45].

5.2.5 New Segmentation Approaches Are Demanding

The existing bottleneck problems in time series segmentation motivate us to develop new segmentation approaches that have the following desirable properties:

- employ a data-independent measure, which is not affected by the scales of a time series, and can be adjusted easily for various time series in real-world applications.
- make a direct and convenient trade-off between approximation accuracy and dimensionality reduction in the segmentation process.
- generate robust performances in presence of time series noises and outliers.
- achieve low computational complexity in the segmentation process.
- can be formulated into an efficient online algorithm, achieve online monitoring and processing of massive time series streams with low computational complexity.

In this chapter, we proposed a novel two-stage robust segmentation approach that achieves all the desirable properties above. In particular, a new data-independent threshold measure is proposed in a top-down decomposition framework. A two-stage

offline approach is developed for a robust and accurate time series segmentation using the data-independent threshold strategy. Finally, the offline approach is extended into an efficient online algorithm. The proposed new data-independent threshold strategy, the offline and online time series segmentation approaches are presented in the following three sections.

5.3 A Two-Stage Approach for Time Series Segmentation Using A Data-Independent Threshold Strategy

In this section, we propose a two-stage top-down segmentation approach (TSTD) that can be adaptive to different time series without a tedious parameter-tuning process and with a guarantied approximation accuracy. In general, the first stage decompose a time series in a top-down decomposition structure using a data-independent threshold strategy. A time series is partitioned into non-overlapping intervals, and is roughly approximated by the piecewise interpolation lines. The second stage is a fine-tune step, which adjusts the approximation model using a regression technique.

5.3.1 A New Data-Independent Threshold Strategy

A challenging problem in time series segmentation is to find an appropriate threshold strategy to break down a time series efficiently and reliably in presence of noises and outliers. Most of the current threshold strategies are data-specific and are rather unreliable for dynamic non-stationary time series data. It is desirable to have a data-independent threshold strategy that controls the approximation accuracy directly regardless exact data values.

In this section, we propose a data-independent threshold based on the statistical measure, the coefficient of determination R^2. R^2 is widely used to verify the goodness of fit for a linear regression model. It is the proportion of variability in a data set that is accounted for by the approximation model (e.g., a linear model).

In particular, for a time series segment $X = x_1, x_2, \ldots, x_n$, the 'variability' of the time series is measured by the total sum of squares $SS_{tot} = \sum_{i=1}^{n}(x_i - \bar{x})^2$, where $\bar{x}$

is the mean value of the time series. The regression sum of squares is calculated by $SS_{reg} = \sum_{i=1}^{n}(f_i - \bar{x})^2$, where f_i is the approximated value at time i generated by the regression model. The sum of squared residuals is calculated by $SS_{err} = \sum_{i=1}^{n}(x_i - f_i)^2$. The standard definition of the coefficient of determination R^2 is defined as follows

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}}. \tag{5.1}$$

The value of R^2 represents that how much of the data variability is explained by an approximation model. It generally ranges within [0, 1]. If a R^2 is close to 1, it means that most of the data variance can be explained by the approximation model. Otherwise, if a R^2 is close to 0, it means that the approximation model cannot explain most of the variance, and thus this model is not appropriate to describe the data.

The value of R^2 is data-independent and can be easily applied to evaluate the fit of goodness for data with different scales. This notable property is just desirable in time series segmentation. Intuitively, one can perform linear regression of a time series segment, and check the obtained R^2 value. If R^2 is greater than some threshold (such as 0.8), one can consider the current linear fit is appropriate. Otherwise, we need to decompose this time series segment further. From this point of view, one can use R^2 as the threshold measure to decompose a time series recursively, and represent the time series by the regression lines of the decomposed time series segments. However, a previous study by Shatkay and Zdonik [144] found that this regression-line based method has difficulties in finding appropriate partitioning intervals of a time series. The linear regression-line based method produced poorer results than the interpolation methods, which approximate a time series by connecting its key turning points. Our numerical experiments also show that the direct application of R^2 lead to unreliable and poor segmentation performance. Mainly due to this reason, the interpolation-line based methods are most frequently employed in time series segmentation.

The original R^2 has difficulties to identify segmentation intervals for a time series. To tackle this problem, we propose an interpolation-line based R^2, we call it R^2_{kp}. For a time series segment $X = (x_1, x_2, \ldots, x_n)$, instead of the regression line of X, we employ

the interpolation line that connects the two endpoints (x_1 and x_n), denoted by L, to roughly approximate the trend of the time series. The calculation procedure of R^2_{kp} is similar to the standard R^2, except the calculation of the error term, which calculated as follows,

$$e_i = x_i - l_i, \tag{5.2}$$

where x_i is the time series value at time i; and l_i is the value of the interpolation line L at time point i, it is calculated by

$$l_i = x_1 + (x_n - x_1)\frac{i-1}{n-1}. \tag{5.3}$$

Then the interpolation-line based sum of squared errors is calculated by

$$SS^{kp}_{err} = \sum_{i=1}^{n}(e_i - \bar{e})^2, \tag{5.4}$$

and the interpolation-line based coefficient of determination R^2_{kp} is obtained by

$$R^2_{kp} = 1 - \frac{SS^{kp}_{err}}{SS_{tot}}. \tag{5.5}$$

The value of R^2_{kp} indicates the goodness of fit by the current interpolation-line approximation. The value of R^2_{kp} is generally located within [0, 1], and it can also be negative under some situations when a time series is very badly fitted by the interpolation line. If R^2_{kp} is close to 1, it means that the time series temporal patterns can be perfectly explained by the interpolation line. The smaller the R^2_{kp}, the less goodness of fit the current interpolation line. If R^2_{kp} is close to 0 or even negative, it means that the interpolation line cannot explain most of the time series temporal patterns at all, thus it is necessary to perform a further decomposition for this time series segment.

The R^2_{kp} defined above is a data-independent measure. It can be conveniently applied in a time series segmentation process in a recursive way. One can use R^2_{kp} to control the approximation accuracy of each decomposed time series segment directly.

For example, a threshold R^{2*}_{kp} of 0.9 guarantees that at least 90% temporal variations can be explained by the endpoint-interpolation-line for each time series segment. With this impressive property, it can be embedded in a time series decomposition to possess various time series data with guaranteed approximation accuracy and without a tedious threshold tuning procedure. In the next subsection, we will present the first stage of the proposed segmentation approach that employs R^2_{kp} in a top-down framework.

5.3.2 Stage One: Top-Down Decomposition Using A Data-Independent Threshold

The top-down segmentation starts from the whole time series sequence, and decompose the time series recursively until all the partitioned time series segments satisfy a threshold-based stop criterion. In this section, we first introduce a data-independent threshold to measure the goodness of linear-fit for each partitioned time series segment. The data-independent threshold measure is defined as follows.

Definition 1. Given a time series $Y = (y_1, y_2, \ldots, y_p)$, the interpolation line that connects its two endpoints (y_1 and y_p) is denoted by $L = (l_1, l_2, \ldots, l_p)$. The data-independent threshold measure, called the interpolation-line based coefficient of determination R^2_{kp} is defined by

$$R^2_{kp} = 1 - \frac{SS^{kp}_{err}}{SS_{tot}}. \tag{5.6}$$

where $SS_{tot} = \sum_{i=1}^{p}(y_i - \bar{y})^2$ is the total sum of squared values of Y, $\bar{y}$ is the mean value of the time series values; $SS_{err} = \sum_{i=1}^{p}(y_i - l_i)^2$ is the sum of squared residuals between Y and the interpolation line L, and $l_i = y_1 + (y_p - y_1)(i-1)/(p-1)$ is the ith value of the interpolation line L.

The value of R^2_{kp} is totally data-independent. It is used to measure if the trend of the time series can be approximately represented by its endpoints interpolation line. A R^2_{kp} value is generally within [0, 1], and can be negative under when a time series is very badly fitted by the interpolation line. If R^2_{kp} is close to 1, it means that the

temporal patterns of a time series can be perfectly explained by the interpolation line. The smaller the R^2_{kp}, the worse the trend-fit of the interpolation line. If R^2_{kp} is close to 0 or even negative, it means that the interpolation line cannot represent the temporal trend of the time series at all.

Using R^2_{kp} as a data-independent threshold, the R^2_{kp}-based top-down segmentation framework is illustrated in Figure 5.2 (first stage part). At each step of the top-down decomposition, we first calculate R^2_{kp} for all the partitioned time series segments (X_1, X_2, ..., X_d, where d is the number of partitioned segments at the current step. Given a threshold value R^{2*}_{kp}, the R^2_{kp}-based decomposition decision-making rule is as follows

- If $min(R^2_{kp}(X_i)) \geq R^{2*}_{kp}$, all the time series partitioned segments approximately follow linear trend, and stop the decomposition process.
- Otherwise, partition the time series segment with the minimum R^{2*}_{kp} into two parts according to the breaking-point rule defined in Definition 2.

The most prominent feature of The R^2_{kp}-based top-down framework is that it is capable of achieving a guaranteed level of approximation accuracy irrespective of time series data values. Thus, the above top-down decomposition process gets rid of a tedious threshold tuning procedure when applied to various different time series data sets. For example, a requirement of 90% approximation accuracy corresponds to a R^{2*}_{kp} of 0.9. With this impressive property, one can conveniently process various time series with a guaranteed approximation accuracy. Due to the intrinsic statistical theory, the proposed new decomposition measure is robust to noises and outliers. Moreover, in the proposed top-down decomposition procedure, it is also very convenient to control the compression rate directly by introducing another measure, the minimum length of the partitioned time series. That is, if the time series segment is less than the length threshold, it will not be partitioned any more.

Definition 2. Given a time series $Y = (y_1, y_2, \ldots, y_p)$, a point y_{kp} is the breaking point of Y if it satisfies

$$D_v(y_{kp}) = max(D_v(y_i),\ \mathrm{i} = 1,\ 2,\ \ldots,\ \mathrm{p}), \tag{5.7}$$

where $D_v(y_i)$ is the vertical distance between y_i and the interpolation line connecting

the two endpoints of the time series Y. The $D_v(y_i)$ is calculated by

$$\begin{aligned} D_v(y_i) &= \mid y_i - L_i \mid, \\ &= \mid y_i - y_1 - (x_p - x_1)(i-1)/(p-1) \mid . \end{aligned} \tag{5.8}$$

A demonstration of the break-point selection rule is illustrated in Figure 5.1.

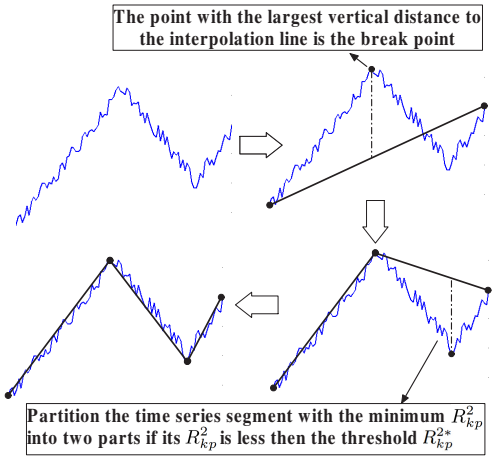


Figure 5.1: A demonstration of the top-down time series segmentation procedure at stage one. The break-point selecting rule defined in Definition 2 is also illustrated in the figure.

5.3.3 Stage Two: Fine-Tune Approximation Model

In the first stage, an interpolation-line based piecewise linear model is obtained. To reduce approximate error, we designed a second stage to fine-tune the interpolation-line approximation model.

For a time series X, suppose it has been partitioned into m time series segments denoted by $(X_1, X_2, \ldots, X_m)$ in the first stage. In the second stage, we perform the least-squares linear regression (LSLR) on each partitioned time series segment. For a time series segment $X_i = (x_{p_1}, x_{p_1+1}, \ldots, x_{p_2})$, where p_1 and p_2 are the time indices,

the LSLR linear regression model of X_i is represented by

$$\hat{X}_i = a_i + b_i X_i, \tag{5.9}$$

where a_i and b_i are the slope and intercept of the linear model, which can be calculated by

$$b_i = \frac{\sum_{k=p_1}^{p_2} x_k k - \sum_{k=p_1}^{p_2} x_k \sum_{k=p_1}^{p_2} k/n_i}{\sum_{k=p_1}^{p_2} x_k^2 - (\sum_{k=p_1}^{p_2} x_k)^2}, \tag{5.10}$$

$$a_i = \sum_{k=p_1}^{p_2} k/n_i - b_i \sum_{k=p_1}^{p_2} k/n_i, \tag{5.11}$$

where $n_i = p_2 - p_1 + 1$ is the length of the time series subsequence X_i.

We perform LSLR on each partitioned time series intervals, and calculate the intersection points of each pair of neighboring regression lines. The final approximation model is the piecewise lines that connect the intersection points of the regression lines.

The whole work flow of the two-stage segmentation procedure is shown in Figure 5.2. The pseudocode of the TSTD algorithm is shown in Algorithm 1. And a demonstration of the two-stage segmentation precess is shown in Figure 5.3. The second fine-tune stage greatly reduces the sum of squared residuals from 170.00 to 48.52. It achieves a 71.5% reduction of approximation error.

5.3.4 Rationale for the Two-Stage Segmentation Algorithm

Currently, a time series piecewise linear approximation is generally achieved by either using the interpolation line of each segment's endpoints, or by calculating the linear regression line through each segment. Both of them have distinct merits and limitations. From a previous study by Shatkay and Zdonik [144], it was found that the latter often produces poorer results although it has superior intrinsic mathematical (statistical) properties. Based on our experiments, we also find that a decomposition only based on a regression criterion can generate rather unreliable decomposition results compared

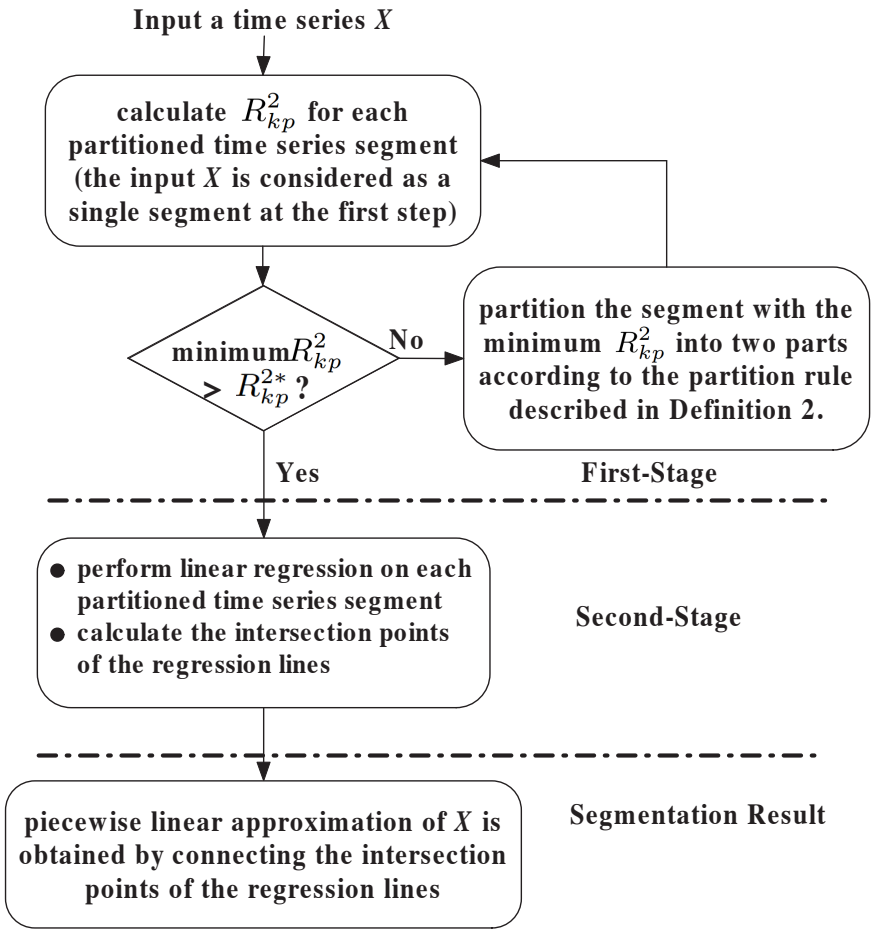


Figure 5.2: The flowchart of the TSTD segmentation algorithm for time series segmentation. The first stage employs a R^2_{kp}-based top-down decomposition rule to partition a time series into piecewise intervals. The time series segments in these intervals approximation follows a linear trend. The second stage is a fine-tune stage, which applies the linear regression technique to adjust the approximation line for each partitioned time series segment.

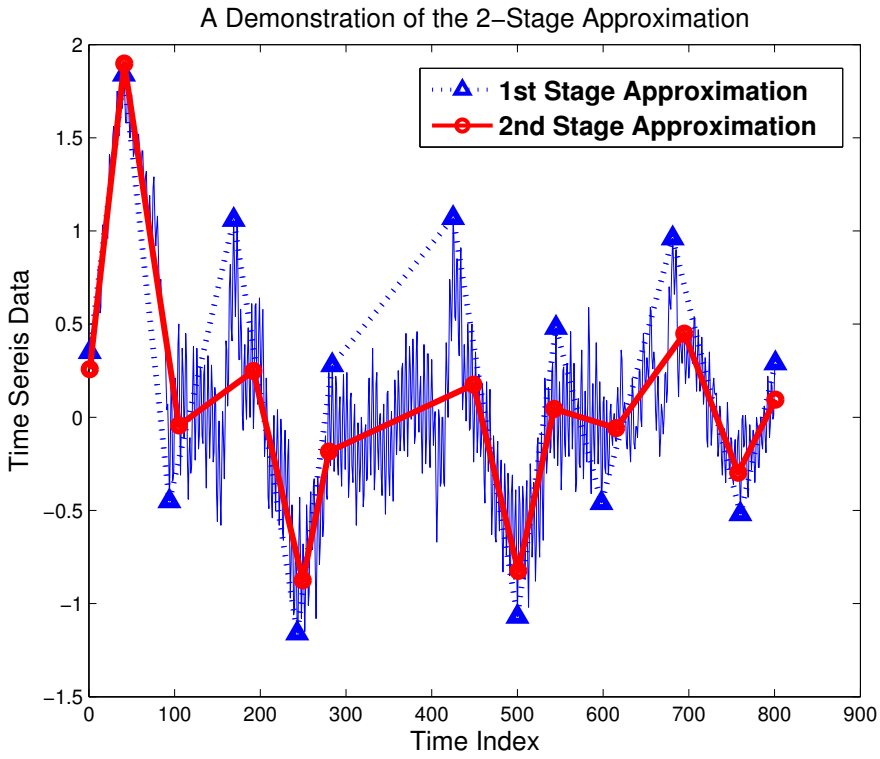


Figure 5.3: A demonstration of the decomposition procedure of the top-down time series segmentation approach. The sum of squared residuals of the two stages are 170.00 and 48.52, respectively. The second stage effectively reduces the approximation error.

Algorithm 1 The Two-Stage Top-Down Segmentation (TSTD) Algorithm

Input: time series $X = (x_1, x_2, \ldots, x_N)$, and the decomposition threshold R^{2*}_{kp}
Output: piecewise linear segments $L^*_{kp} = (L^*_1, L^*_2, \ldots L^*_m)$, the key turning points $S^* = (s^*_1, s^*_2, \ldots, s^*_{m+1})$
procedure TSTD(X, R^{2*}_{kp})
    The First Stage:
    Initial: $L_1 = X$, $L_{kp} = [L_1]$, $R = [R^2_{kp}(L_1)]$, $S = [1, n]$
    while $min(R) < R^{2*}_{kp}$ **do**
        $L_j = argminR$;
        $(R = [R^2_{kp}(L_1), \ldots, R^2_{kp}(L_i), \ldots, R^2_{kp}(L_k)])$,
        where k is number of segments at this step.)
        $[L_I, L_{II}, s_{kp}] = KPseg(L_j)$;
        (partitioned into 2 parts by the key point)
        $S = concat(S, s_{kp})$
        $R^2_{kp}(L_I) = cal\_R2kp(L_I)$;
        $R^2_{kp}(L_{II}) = cal\_R2kp(L_{II})$;
        $L_{kp} = takeout(L_{kp}, L_j)$; (delete L_j)
        $R = takeout(R, R^2_{kp}(L_j))$;
        $L_{kp} = addinto(L_{kp}, L_I, L_{II})$;
        $R = addinto(R, R^2_{kp}(L_I), R^2_{kp}(L_{II}))$;
    end while
    The Second Stage:
    Initial: $S^* = [1, n]$, $L^* = [\ ]$, $M = length(S)$;
    for $i = 1 : M - 1$ **do**
        $L^*_i = regression(X_{L_i})$;
        $L^*_{i+1} = regression(X_{L_{i+1}})$;
        $s^*_i = intersection(L^*_i, L^*_{i+1})$;
        $S^* = addinto(S^*, s^*_i)$;
        $L^* = addinto(L^*, L^*_i)$;
    end for
end procedure

to the interpolation method. This is mainly because the regression-line based methods have difficulties in finding appropriate partitioning intervals due to various dynamic changing patterns in a time series. However, it is the best linear fit if a segment of time series data approximately follow a linear trend. On the other hand, the interpolation-line based methods are generally good at identifying the key turning points in a time series. While an interpolation-line (connecting two time series endpoints) is generally not the best linear approximation for a segment of time series. And sometimes the slope of the interpolation line can be considerably deviated from the true time series trend (the best fitting regression line).

Based on this observation, we developed the two-stage approximation framework, which combines the two techniques in a top-down segmentation framework. In the first stage discussed above, a time series is approximated by a series of interpolation lines that connecting a set of 'optimal' breaking points. Then in the second stage, a linear regression is performed on each segment of time series. The second step fine tunes the piecewise approximation lines into unbiased best-fitting regression lines with least squared errors for each segmented time series. The final approximation is the connected regression lines. The proposed two-stage approach makes the best of the two techniques to achieve a better approximation than those only use one. Most importantly, the two-stage method further enhance the robustness of the algorithm by eliminating the influences of the values of the key turning points.

However, the interpolation line in each partitioned interval is generally not the best linear approximation for that time series segment. For a time series segment, the slope of its interpolation line is often considerably deviated from its best linear trend defined by its regression line.

5.3.5 Complexity Analysis

Given a time series $X = (x_1, x_2, \ldots, x_n)$ with n points, the computational complexity of a top-down algorithm is generally $O(Kn^2)$ according to Keogh et al. [74]. In this section, we show that the proposed two-stage top-down segmentation algorithm achieves a complexity of $O(Kn)$ as follows.

- **The first stage**: at the first decomposition, the complexity to calculate the approximation residuals is $O(n)$; the complexity to calculate the decomposition criterion R^2_{kp} is $O(n)$; and finally the complexity to pick up the breaking point is $O(n)$. Thus the complexity of a full step of decomposition is $O(3n)$. For all decompositions after the first one, the $O(3n)$ is the upper bound of their complexity, since the decomposed time series subsequences have fewer and fewer number of data points. Define K the number of partitioned intervals obtained in the first stage, then there are $K-1$ decompositions, and the upper bound of the

complexity of the first stage is $O(3n(K-1))$.

- **The second stage**: denote the number of points contained in all the time series subsequences above are $(n_1, n_2, \ldots, n_K)$. Then for an interval of time series with n_i points, the complexity to calculate its regression line is $O(n_i)$ using the closed-form formulas 5.10 and 5.11. The total complexity of the second stage is $\sum_{i=1}^{K} O(n_i) = O(\sum_{i=1}^{K} n_i) = O(n)$.

Based on the analysis above, the complexity of the two-stage top-down time series segmentation algorithm is $O(3(K-1)n) + O(n) = O((3K-2)n)$, where $K \ll n$ in most cases.

5.3.6 Experimental Results

This section presents the experimental results based on three performance measures. We first investigate the segmentation performance of the proposed TSTD algorithm with respect to the threshold R^{2*}_{kp}. And then we compare TSTD with two other popular approaches, including the classic bottom-up (BU) method and the adaptive piecewise constant approximation (APCA) method.

5.3.7 Performance Measures

Three measures are employed to evaluate the performance of a time series segmentation algorithm. For a time series $X = (x_1, x_2, \ldots, x_n)$, assume its piecewise linear model has M piecewise linear segments denoted by $L_X = (L_1, L_2, \ldots, L_M)$, and the lengths (number of data points) of the segments are denoted by $q_1, q_2, \ldots, q_M$. The three measures are described as follows.

- The first measure considers overall approximation accuracy of the whole time series. It is desirable to measure how much of the overall time series variation is accounted by the piecewise linear model. Denote the sum of squared residuals between X and L_X by $SSR(X, L_X)$, and the sum of squared values of the time series by $SS(X)$, then the global approximation performance of the whole time

series is calculated by

$$P_{off} = \frac{SS(X) - SSR(X, L_X)}{SS(X)} \quad (5.12)$$

$$= 1 - \frac{\sum_{i=1}^{n}(x_i - l_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x_i})^2}. \quad (5.13)$$

- The second measure is compression rate. It is used to evaluate the dimensionality reduction of a segmentation approach. The compression rate in this study is defined by

$$CR = n/N_{L_X}, \quad (5.14)$$

where n is the number of time series data points, and N_{L_X} is the number of parameters to represent the piecewise linear model L_X.

- The third measure is the computing time, denoted by T. To deal with massive time series data, a segmentation algorithm is definitely desirable to work as fast as possible.

5.3.7.1 Performance Characteristics of TSTD with Different Threshold Values

We employed a popular type of artificial time series, namely random walk, to investigate the TSTD with respect to the threshold R_{kp}^{2*}. A random walk time series simulates a trajectory that consists of successive random steps. It has been widely applied to model a stochastic process in many fields including ecology, economics, psychology, computer science, physics, and chemistry [159]. Thus, it can be a valuable testbed to investigate the properties of time series segmentation algorithms that may be applied to diverse fields.

The TSTD was applied to approximate 100 random-walk time series with very different amplitude scales. For each time series, we changes the threshold R_{kp}^{2*} values from 0.05 to 0.95 at a step length of 0.05. The averaged results over 100 time series are shown in Figure 5.4. The subplot (a) shows the approximation accuracy of the TSTD

with respect to threshold R_{kp}^{2*}. As R_{kp}^{2*} increases, the overall approximation accuracy P is monotonically increased from around 0 to almost 1. A nearly 1 of P indicates that most of the time series variation can be explained by the piecewise linear approximation model. On the other hand, the subplot (b) shows that the compression ratio CR is monotonically decreased from more than 800 to around 50. This comparison clearly shows that the threshold R_{kp}^{2*} makes a trade-off between approximation accuracy and compression rate directly, and does not rely on the characteristics of a time series, since the tested 100 random walk time series have very different amplitude scales. This notable data-independent property of the R^{kp} makes it very convenient in practical applications. In addition, the computing time is also shown in subplot (c). The averaged computing time increases linearly as R_{kp}^{2*} increases. This is because a higher value of R_{kp}^{2*} lead to more decompositions in the top-down segmentation process.

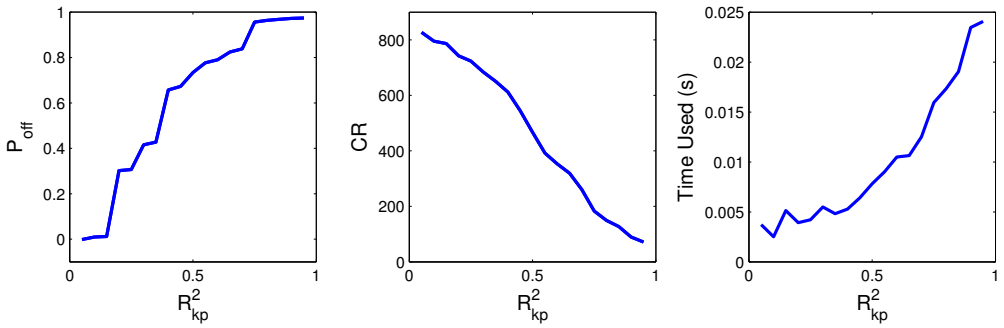


Figure 5.4: The performance of the TSTD with respect to the setting of R_{kp}^{2*}. The results were averaged over the 100 experiments on a 'random walk' time series. It shows that R_{kp}^{2*} is the key to control the trade-off between the approximation accuracy and compression rate. As R_{kp}^{2} increases from 0.1 to 0.9, the P_{off} increases monotonically from around 0 to 1, while the compression rate is decreased from 800 to around 50.

5.3.7.2 Performance Demonstration by Several Non-Stationary Time Series

In this section, we demonstrate the segmentation performance of the online SWTD through a sensor signal contaminated heavily by noises. This types of signals are very common in practice collected from various sensors. The segmentation result is shown in

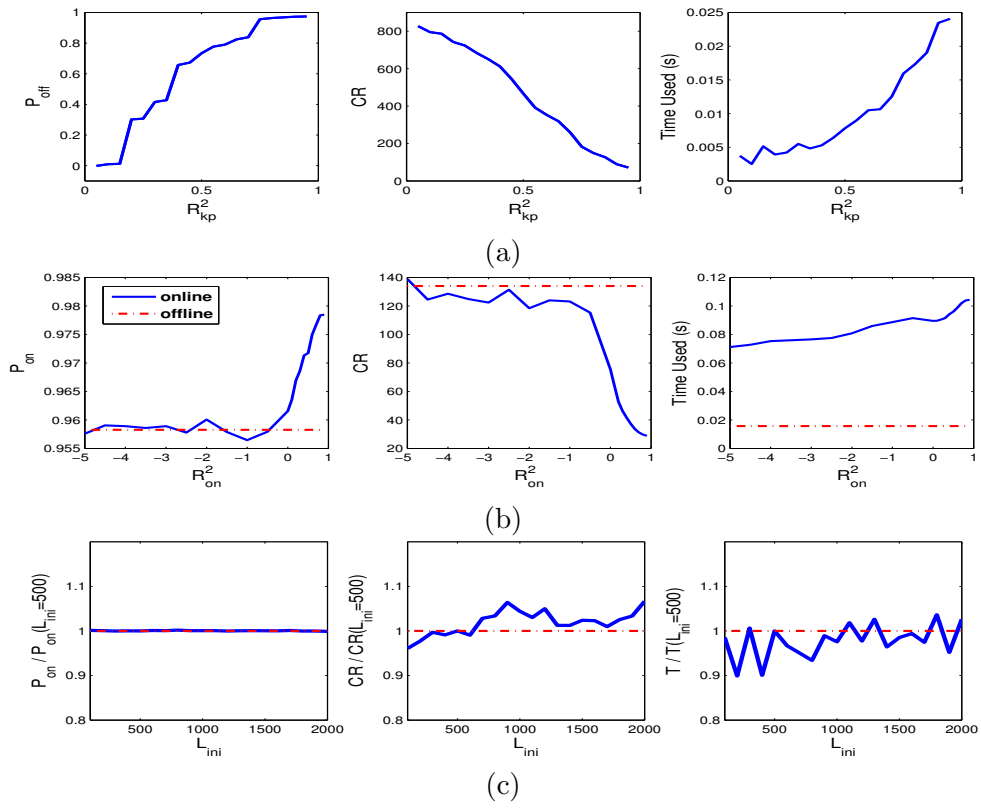


Figure 5.5: The segmentation performances of the offline TSTD and the online SWTD algorithm with respect their parameter settings. The results were averaged over experiments of 100 'random walk' time series with 3000 samples. (a) The performance of TSTD with respect to R^2_{kp}, (b) The performance of SWTD with respect to R^2_{on} using $L_{ini} = 500$ and $R^2_{kp} = 0.9$. (c) Using the performances at $L_{ini} = 500$ as a reference, the relative performances of SWTD with respect to L_{ini} are shown based on $R^2_{on} = -0.5$ and $R^2_{kp} = 0.9$. The red dotted lines in (b) and (c) represent the performances of the offline TSTD using $R^2_{kp} = 0.9$.

Figure 5.6. The proposed SWTD is capable of capturing the major temporal patterns of the time series in presence of the heavy noises. As a comparison, the result of the popular online approach SWAB is also shown in Figure 5.15. The SWAB employs a 'maximum error' threshold. We varied threshold values of 'maximum error', and made it achieve a similar segmentation with the SWTD. Although the approximation accuracy of SWAB (0.68) is a little higher than SWTD(0.66), the SWAB approach took 100 times more time (16.58s) than the SWTD (0.16s). The computationally speed is vital for online monitoring of massive time series data. This example demonstrates the great potential of our proposed SWTD in online real-time applications.

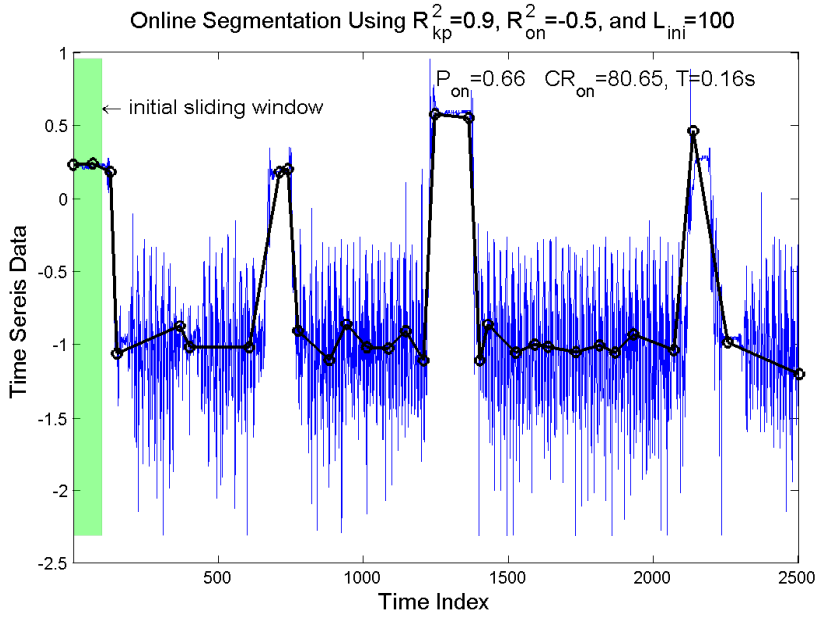


Figure 5.6: The segmentation performance of SWTD on a very noisy sensory signal.

5.3.7.3 Comparison to Other Segmentation Techniques

In this section, we compare the proposed TSTD with two popular time series segmentation approaches BU and APCA. The numerical experiments were conducted on 24 real-world time series data sets from various fields including neurophysiology, finance, industry, medicine, biology, and geography. The data were selected from from the UCR time series data archive [76].

The threshold of TSTD R_{kp}^{2*} was set at 0.8 for all the data sets. Since both BU and APCA rely on a data-dependent threshold, the 'maximum approximation error'. To make a fair comparison, we chose the thresholds such that the BU and APCA algorithms have approximately the same compression ratio for each time series. The segmentation results of TSTD, BU, and APCA are summarized in Table 5.1.

Given almost the same compression ratio, the proposed TSTD achieved the highest approximation accuracy on average with the least standard deviation, that is $P(TSTD) = 0.97 \pm 0.02$. The APCA has the lowest approximation accuracy with the largest standard deviation, $P(APCA) = 0.90 \pm 0.08$. The performance of BU algorithm is in the middle with $P(BU) = 0.95 \pm 0.06$. Most importantly, it is noticed that the proposed

TSTD approach achieved higher accuracies at an even lower computing time. The TSTD approach shows a huge superiority of computational efficiency compared to BU and APCA. For the 24 time series, the TSTD approach has an averaged computing time of 0.08 second. While the average computing times of BU and APCA are 1.48 and 1.61 second, respectively. The proposed TSTD is about 20 times faster than the BU and APCA approaches.

Moreover, a big advantage of the proposed TSTD is that it can automatically choose an appropriate segmentation according to the accuracy requirement. For the 24 time series with dramatically different data scales and statistics, we can use the same threshold value of R_{kp}^{2*}, and the approximation accuracy can be guaranteed. On the other hand, the approaches, such as BU and APCA, generally require a user to set a data-dependent threshold (e.g.,'maximum error') or to set the required number of segments. The parameters are not related to the approximation accuracy directly, and one have to discover an appropriate setting in an inefficient trial-and-error manner.

5.3.8 Summary of The Proposed Offline TSTD Algorithm

From the literature study, we find that most of the current segmentation approaches are incapable of controlling the approximation accuracy and the compression rate efficiently and directly. The approximation accuracy is generally controlled indirectly through some data-specific threshold strategy, which often requires a tedious threshold tuning procedure to process various time series data in practice. Also we notice that the current threshold strategies are not statistically robust due to their intrinsic heuristic structures. Thus the resulting segmentation approaches are generally not robust to time series noises, outliers, and time-varying properties. It is still an open question to determine 'optimal' levels of these segmentation thresholds for various time series data. This limitation seriously hampers the potentials of the time series segmentation techniques in practical applications.

To tackle this problem, we provide an effective solution to piecewise linear segmentation of time series data. The proposed time series segmentation approach employs a

Table 5.1: The segmentation results of TSTD, BU, and APCA for 24 time series data sets.

		P			CR			T		
Time Series Data		TSTD	BU	APCA	TSTD	BU	APCA	TSTD	BU	APCA
1	ERP 1	0.98	0.98	0.95	11.32	11.28	11.28	0.27	5.47	6.06
2	ERP 2	0.98	0.98	0.95	11.41	11.36	11.36	0.09	4.92	5.30
3	ERP 3	0.98	0.98	0.96	10.91	10.87	10.87	0.09	4.91	5.33
4	EOG	0.99	1.00	0.98	29.77	29.77	29.77	0.03	4.13	4.42
5	Steamgen 1	0.98	0.96	0.87	73.17	71.43	71.43	0.02	5.11	5.42
6	Steamgen 2	0.98	0.99	0.97	8.85	8.82	8.82	0.11	4.78	5.31
7	Steamgen 3	0.96	0.96	0.94	5.61	5.60	5.60	0.19	4.59	5.20
8	Foetal ECG 1	0.89	0.89	0.83	6.11	6.10	6.10	0.19	4.67	5.24
9	Foetal ECG 2	0.97	0.99	0.94	6.98	6.98	6.98	0.13	3.88	4.30
10	TOR95	0.96	0.94	0.87	7.43	7.43	7.43	0.06	2.22	2.45
11	Power Data	0.98	1.00	0.99	6.63	6.63	6.63	0.11	3.03	3.33
12	Burst	0.99	0.99	0.98	47.19	46.31	46.31	0.02	4.16	4.42
13	Fluid Dynamics	0.95	0.96	0.89	7.31	7.31	7.31	0.11	3.84	4.25
14	PH Data 1	0.95	0.88	0.80	16.01	15.88	15.88	0.06	3.20	3.42
15	PH Data 2	1.00	1.00	1.00	8.10	8.07	8.07	0.09	3.08	3.36
16	Shuttle 1	0.99	1.00	0.99	40.00	38.46	38.46	0.01	1.55	1.67
17	Shuttle 2	1.00	1.00	1.00	13.70	13.51	13.51	0.03	1.52	1.63
18	Shuttle 3	1.00	1.00	1.00	14.93	14.71	14.71	0.03	1.50	1.64
19	Greatlakes 1	0.98	0.92	0.90	6.19	6.15	6.15	0.05	1.39	1.56
20	Greatlakes 2	0.97	0.94	0.94	7.24	7.24	7.24	0.05	1.42	1.58
21	Greatlakes 3	0.97	0.82	0.78	5.93	5.93	5.93	0.06	1.38	1.55
22	Flutter	0.98	0.95	0.77	9.23	9.14	9.14	0.03	1.47	1.61
23	Wool	0.98	0.96	0.78	8.75	8.83	8.83	0.05	1.45	1.61
24	Attas	0.93	0.77	0.75	17.31	17.31	17.31	0.05	2.86	3.06
AVE.		0.97	0.95	0.90	15.45	15.25	15.25	0.08	3.12	3.41
Median		0.02	0.06	0.08	16.05	15.63	15.63	0.06	1.48	1.61

data-independent threshold strategy in a top-down decomposition framework. A two-stage procedure was developed to enhance the robustness and approximation accuracy of the segmentation performance. The proposed TSTD has been validated by extensive experiments on various real-world time series data sets. The numerical studies show that the proposed TSTD generated a superior overall performance over two other popular time series segmentation algorithms in terms of the accuracy and computational efficiency given the same compression ratio. The proposed TSTD algorithm can generally find the key skeleton points of a noisy time series efficiently without many redundant 'pseudo' key points in most of the cases.

In addition, we also noticed that the vast majority of the time series segmentation approaches are offline algorithms, only a few of them have been extended for online segmentation of time series streams with a low (linear) computational complexity [97,

112, 45]. In the next section, we will extend the offline TSTD into an efficient online algorithm using closed-form incremental online updating formulas.

5.4 An Efficient Approach for Automated Online Segmentation of Time Series

The time series segmentation algorithm proposed in the previous section is an offline algorithm. Due to its underlying closed-form based formulations, it is able to achieve a great computational efficiency. Most importantly, the closed-form formulas can be reformulated into an incremental online version that does not require an expensive storage and manipulation of huge historical time series data. The incremental formulas extends our proposed offline algorithm into an online version conveniently.

5.4.1 A Fast and Efficient Online Segmentation Framework

We extend the offline TSTD algorithm into an online segmentation algorithm, called sliding window and top-down (SWTD) approach. An online measure, called online coefficient of determination denoted by R^2_{on}, is proposed to make a segmentation decision online. Basically, we use a sliding window approach to monitor a time series, and update R^2_{on} online as each point arrives. If R^2_{on} is below a threshold, we apply TSTD to perform time series segmentation in the sliding window; otherwise, continue to read in a new data. The flowchart of the online SWTD approach is shown in Figure 5.7.

5.4.2 An Incremental Online Decision-Making Measure

An online segmentation algorithm needs to process incoming time series continually and make decomposition decision in real-time. We propose an online decision-making measure, called online coefficient of determination denoted by R^2_{on}, which is defined as follows.

Definition 3. Online monitoring and segmentation of a time series using SWTD (from left to right), a time series point x_n is read in the sliding window at time index

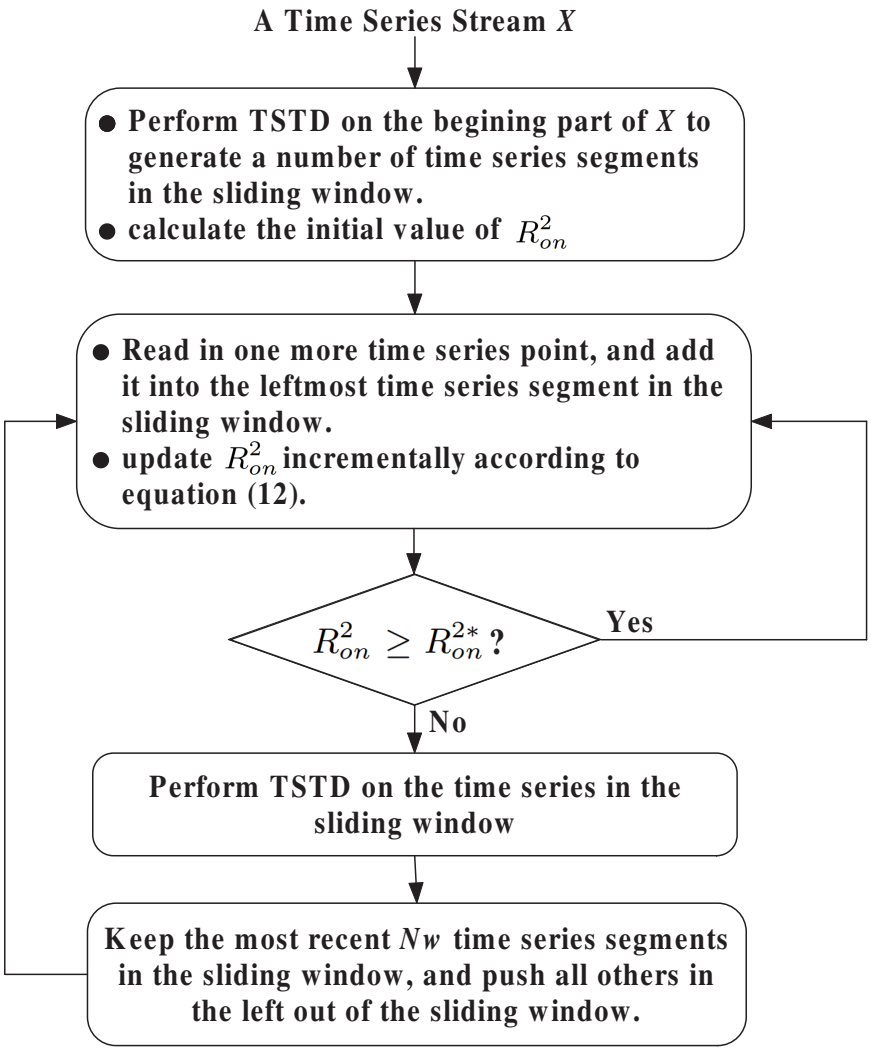


Figure 5.7: The flowchart of the online time series segmentation of SWTD. The online measure R^2_{on} is calculated incrementally in time $O(1)$ as a new point arrives. If R^2_{on} is smaller than a threshold, it trigger a TSTD segmentation on the time series in the sliding window.

n. After adding in x_n, the rightmost time series segment in the sliding window is $X_i = (x_p, x_{p+1}, \ldots, x_n)$, where i indicates that it is the ith the segment since the time zero. Also denote the rightmost approximation line L_i, which is obtained by TSTD in the most recent segmentation update. The slope and intercept of L_i are a_i and b_i, respectively. We design the online coefficient of determination $R^2_{on}(n)$ to measure that if the recent time series trend can still be represented by the rightmost segmentation

line. The online coefficient of determination at time n is defined as follows

$$\begin{aligned} R^2_{on}(n) &= 1 - \frac{SS^{on}_{err}(n)}{SS^{on}_{tot}(n)} \\ &= 1 - \frac{\sum_{k=p}^{n}(x_k - f_k)^2}{\sum_{k=p}^{n}(x_k - \mu_n)^2}, \end{aligned} \tag{5.15}$$

where $\mu_n = \sum_{i=p}^{n} x_i/(n-p+1)$, and f_k is the value of the approximation line L_i at the time index k, and is given by $f_k = a_i + b_i k$, a_i and b_i is the slope and the intercept of L_i.

Compared to the offline measure R^2_{kp}, the R^2_{on} is not calculated by the interpolation line connecting the two endpoints of a time series. Instead, it uses the extension of the rightmost approximation line to fit the most recent time series trend. As more and more new time series data stream in, the approximation line is extended to fit the new data. If R^2_{on} is much lower than 1 (such as 0, or become negative), it means that the rightmost approximation line becomes inappropriate to represent the recent time series trend, thus a new segmentation update in the sliding window is required using TSTD.

Most importantly, the online measure R^2_{on} can be calculated incrementally online using a closed-form formula. The incremental update property of R^2_{on} make the online SWTD algorithm work super fast to monitoring and segmentation of time series streams in real-time. The incremental formulation of R^2_{on} is presented below.

Theorem 1. *The online coefficient of determination R^2_{on} defined in equation 5.15 can be updated with a closed-formula in time $O(1)$ when a new time series point is read in the sliding window.*

Proof. According to equation 5.15, we just need to derive that $SS^{on}_{err}(n)$ and SS^{on}_{tot} can be updated incrementally. Assume, at time index $n+1$, a new time series point x_{n+1} arrives and is added to the rightmost segment $X_i = (x_p, x_{p+1}, \ldots, x_n)$, and becomes

$X_i = (x_p, x_{p+1}, \ldots, x_n, x_{n+1})$. The mean of the growing time series segment X_i can be calculated incrementally as follows:

$$\mu_{n+1} = \mu_n + \frac{1}{N(X_i)}(x_{n+1} - \mu_n). \tag{5.16}$$

where $N(X_i) = n - p + 2$ is the updated length of X_i, and $\mu_n = \sum_{i=p}^{n} x_i/(n-p+1)$ is the mean of X_i at time index n, and μ_{n+1} is the mean of X_i at time index $n+1$ after adding in the new point x_{n+1}.

A numerically stable formulation for an incremental online calculation of sample variance was provided and proved in both Knuth [79] and Finch [42]. Based on the incremental closed formula of sample variance, the online incremental calculation of the total sum of squares SS_{tot}^{on} can be formulated as follows

$$\begin{aligned} SS_{tot}^{on}(n+1) &= \sum_{k=p}^{n+1}(x_k - \mu_{n+1})^2 \\ &= SS_{tot}^{on}(n) + (x_{n+1} - \mu_n)(x_{n+1} - \mu_{n+1}). \end{aligned} \tag{5.17}$$

The $SS_{tot}^{on}(n+1)$ formulated in equation 5.17 can be directly calculated from its previous value $SS_{tot}^{on}(n)$, the new time series point x_{n+1}, the segment mean μ_n at time n, and the segment mean μ_{n+1} at time $n+1$. Since the segment mean can be calculated incrementally according to 5.16, the SS_{tot}^{on} is updated in time $O(1)$ when a new data arrives.

The sum of approximation residuals SS_{err}^{on} can also be updated in an incremental formula as follows

$$
\begin{aligned}
SS^{on}_{err}(n+1) &= \sum_{k=p}^{n+1}(x_k - f_k)^2 \\
&= SS^{on}_{err}(n) + (x_{n+1} - f_{n+1})^2 \\
&= SS^{on}_{err}(n) + (x_{n+1} - a_i - b_i \times (n+1))^2.
\end{aligned}
\tag{5.18}
$$

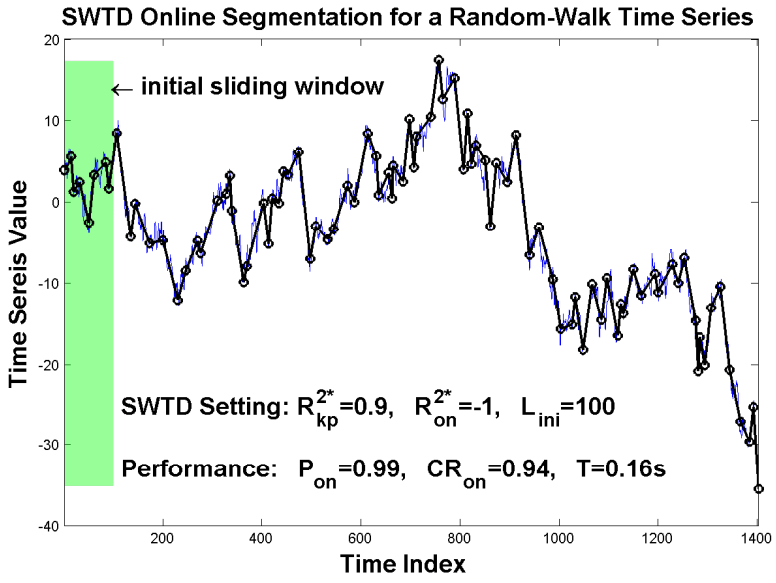
where a_i and b_i are the slope and the intercept of the rightmost approximation line L_i, which is obtained by TSTD in a most recent update. The $SS^{on}_{err}(n+1)$ can be calculated directly from its previous value $SS^{on}_{err}(n)$, the new time series point x_{n+1}, and the paraments a_i and b_i of the rightmost approximation line. Thus the SS^{on}_{err} is updated in time $O(1)$ when adding in a new data point.

Finally, the $R^2_{on}(n+1)$ can be incrementally calculated online as follows

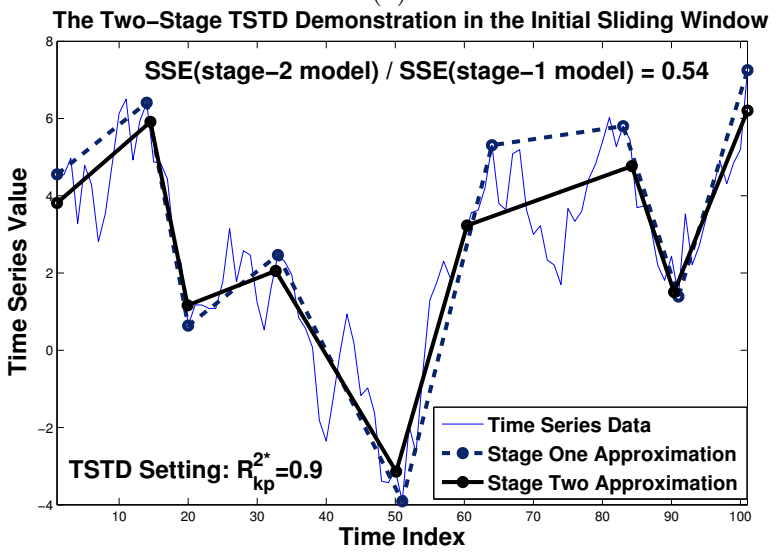
$$
\begin{aligned}
R^2_{on}(n+1) &= 1 - \frac{SS^{on}_{err}(n+1)}{SS^{on}_{tot}(n+1)} \\
&= 1 - \frac{SS^{on}_{err}(n) + (x_{n+1} - a_i - b_i(n+1))^2}{SS^{on}_{tot}(n) + (x_{n+1} - \mu_n)(x_{n+1} - \mu_{n+1})}.
\end{aligned}
\tag{5.19}
$$

The $R^{on}_2(n+1)$ can be calculated directly from $SS^{on}_{err}(n)$, $SS^{on}_{tot}(n)$, x_{n+1}, μ_n, μ_{n+1}, a_i, and b_i. Since the $SS^{on}_{err}(n)$, $SS^{on}_{tot}(n)$, x_{n+1}, μ_n, and μ_{n+1} can be calculated incrementally in $O(1)$, and the a_i, and b_i are known regression parameters at time $n+1$. Thus the $R^{on}_2(n+1)$ can be updated incrementally in time $O(1)$ when a new data point is added in. □

Instead of manipulating massive historical time series data, the proposed measure R^2_{on} only needs to store and manipulate three incremental variables (μ, SS^{on}_{tot}, and SS^{on}_{err}) and two parameters (the intercept a_i and the slope b_i)of the most recent approximation line L_i. The equation 5.19 is of great importance since it achieves an efficient incremental online calculation. The complexity of the incrementally online update of R^2_{on} is extremely low, only **O(1)**. This notable property makes it an ideal criterion to



(a)



(b)

Figure 5.8: (a) A demonstration of the online segmentation of SWTD on a random-walk time series with 1500 data points. The sliding window is initialized by the first 100 points, and the threshold values are $R^2_{kp} = 0.90$ and $R^2_{on} = -1$. The performance of segmentation is very robust to the time series noises, and can achieve an overall accuracy of 99% with a compression rate of 15.57. (b) A demonstration of the two-stage TSTD segmentation on the time series of the initial sliding window with 100 data points. The sum of squared approximation errors (SSE) is reduced by about 50% after the second fine-tune stage.

verify the goodness-of-fit of the most recent approximation line in real time, even if the sampling rate is very high. In the next section, we will present the R^2_{on}-based online time series segmentation framework.

5.4.3 The SWTD Online Segmentation Framework

The online segmentation procedure of the SWTD algorithm using R^2_{on} and R^2_{kp} is summarized in the following five steps. The flowchart of the SWTD online segmentation is shown in Figure 5.7. The SWTD framework can be summarized into the following steps.

- **step 1**: Determine the threshold values of R^{2*}_{kp} and R^{2*}_{on}. The higher the R^{2*}_{on}, the more frequently the online algorithm will be updated. The higher the R^{2*}_{kp}, the more accurate a time series will be approximated using more line segments. In addition, we use N_w to control the number of time series segments in the sliding window for online monitoring.

- **step 2**: The procedure is initialized by the beginning part of a time series, the length of which is denoted by L_{ini}. The time series segment $(x_1, x_2, \ldots, x_{L_{ini}})$ forms the starting sliding window. The TSTD algorithm is performed within the initial window, and generates a number of initial approximation lines (or one line).

- **step 3**: At each time step, a new data point is read in. Calculate the R^2_{on} using the incremental formula given in equation 5.19.

- **step 4**: Make a segmentation decision:

  - If $R^2_{on} \geq R^{2*}_{on}$, the current regression line and its extension line can still explain the required level of variance of the corresponding time series data.
  - If $R^2_{on} < R^{2*}_{on}$, the current approximation line and its extension cannot 'approximately' represent the recent time series. The TSTD algorithm is performed within the sliding window using the R^{2*}_{kp}.

- **step 5**: If segmentation is performed in step 4, the most recent N_w time series segments are kept in the sliding window, and all others and their approximation lines leave the sliding window. If no segmentation is performed in step 4, the size of the current sliding window is increased by one, and the algorithm continues to read in the next time series data point.

A demonstration example of SWTD on a random-walk time series with 1400 data points is shown in Figure 5.8. During the online monitoring process, we allow the sliding window to keep a few (e.g., $N_w = 5 - 6$) most recent time series segments for a robustness purpose. Some short segments in the sliding window caused by noises may merged into bigger ones after several updates later on when more data points enter the sliding window. The time series segmentation lines that have left the sliding window are the finally form of the approximation, and will not be updated again.

The pseudocode of the SWTD algorithm is shown in Algorithm 2. A demonstration of the online segmentation of a random-walk time series is shown in Figure 5.9.

Algorithm 2 The Sliding Window Top Down (SWTD) Algorithm

Input: online time series $X = (x_1, \ldots, x_i, \ldots)$, the thresholds R^{2*}_{kp} and R^{2*}_{on}, and the initial window size L_{ini}.
Output: piecewise linear segments $L^*_{on} = (L^*_1, L^*_2, \ldots L^*_m)$, the key turning points $S^*_{on} = (s^*_1, s^*_2, \ldots, s^*_{m+1})$
procedure SWTD(X, R^{2*}_{kp}, R^{2*}_{on}, L_{ini})
    Initial: perform TSTD on $X_{1:L_{ini}}$, get R^2_{on}, SS^{on}_{err}, and SS^{on}_{tot}, set $S^*_{on} = [\ ]$, $L^*_{on} = [\ ]$;
    while a data x_k is read in **do**
        $R^2_{on} \leftarrow cal\_R2on(R^2_{on}, x_k)$; (eq. 5.19)
        if $R^2_{on} < R^{2*}_{on}$ **then**
            $[L^*_{kp}, S^*] =$ TSTD(time series in window);
            $[l^*, s^*] = leftmost\_segment(L^*_{kp}, S^*)$;
            $S^*_{on} = addinto(S^*_{on}, s^*)$;
            $L^*_{on} = addinto(L^*_{on}, l^*)$;
        else
            continue to read in data.
        end if
    end while
end procedure

5.4.4 Complexity Analysis

The computational complexity of SWTD is also verified on 100 'random walk' time series with varying lengths ranged from 2^{10} to 2^{17}. The averaged computing times are shown in Figure 5.10 (left plot). The computing times of the TSTD using the same data are also shown in the same plot, they are much smaller. One should notice that much of the computing time of the online SWTD is taken by the online reading-in time

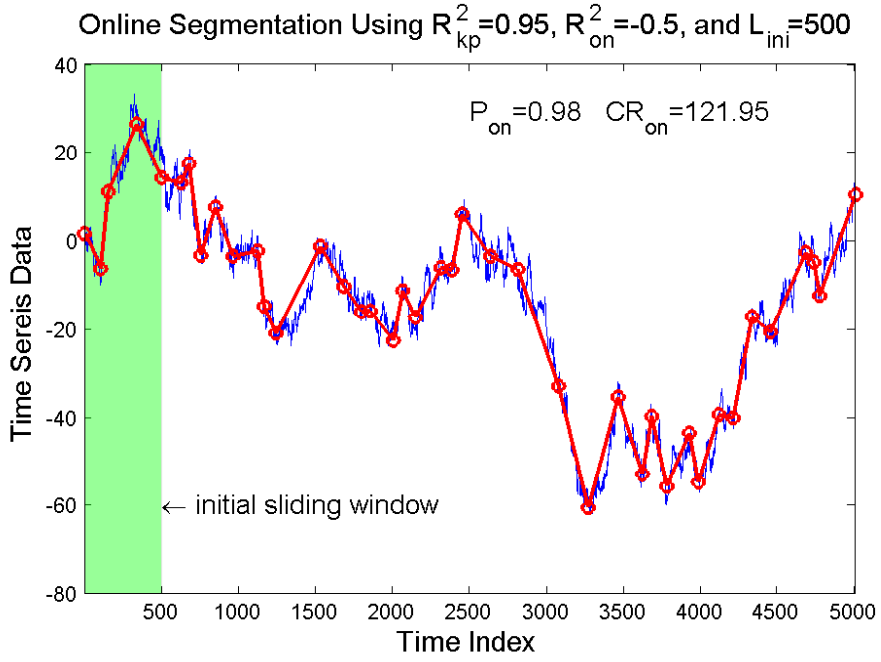


Figure 5.9: A demonstration of the online segmentation of a random-walk time series with 5000 data points. The sliding window is initialized by the first 500 points, and the threshold values are $R^2_{kp} = 0.95$ and $R^2_{on} = -0.5$. The performance of segmentation is very robust to the time series noises, and can achieve an overall accuracy of 98% with a high compression rate of about 122.

series data point by point. The SWTD algorithm itself has a very low complexity of $O(1)$ to make an online decomposition decision using closed-form formulas. Once a decomposition update is required, the complexity of this update is $O(K_{sw}n_{sw})$, where n_{sw} is the number of points in the sliding window, and the K_{sw} is the number of segments within the sliding window.

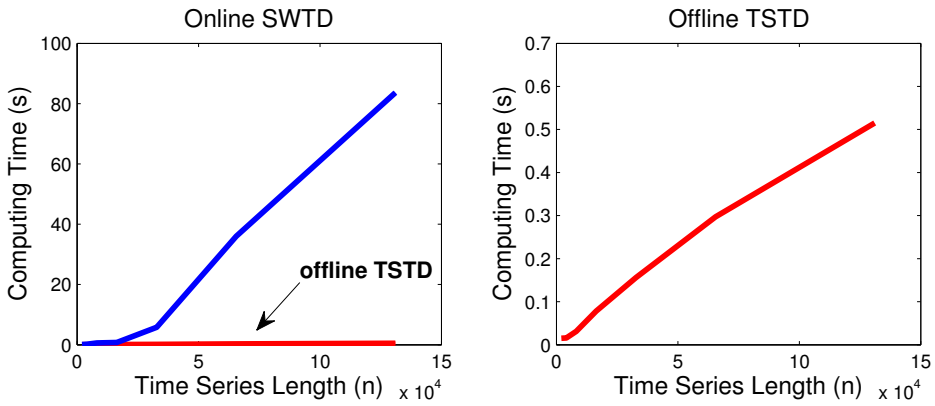


Figure 5.10: The computing time of the online SWTD and the offline TSTD with respect to the length of a time series. The results were averaged over experiments of 100 'random walk' time series with lengths ranged from 2^{10} to 2^{17}.

5.5 Experimental Results

In this section, we employ three performance measures to evaluate the proposed online segmentation approaches, and compare the segmentation results with two other popular online algorithms, including the classic sliding window (SW) method and the sliding window and bottom-up (SWAB) method.

5.5.1 Performance Measures

For a time series $X = (x_1, x_2, \ldots, x_n)$, assume its piecewise linear model has M line segments denoted by $L_X = (L_1, L_2, \ldots, L_M)$, and the lengths (number of data points) of the segments are denoted by $q_1, q_2, \ldots, q_M$. The three measures are described as follows.

- The first measure evaluates the overall approximation accuracy. It calculates

how much of the overall time series variation is accounted by the piecewise linear model. Denote the sum of squared residuals between X and L_X by $SSR(X, L_X)$, and the sum of squared values of the time series by $SS(X)$, then the global approximation performance of the whole time series is calculated by

$$P_{on} = \frac{SS(X) - SSR(X, L_X)}{SS(X)} \tag{5.20}$$

$$= 1 - \frac{\sum_{i=1}^{n}(x_i - l_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x_i})^2}. \tag{5.21}$$

- The second measure is the compression rate, which is defined by

$$CR = 1 - N_{L_X}/n, \tag{5.22}$$

where n is the number of time series data points, and N_{L_X} is the number of parameters to represent the piecewise linear model L_X. The value of CR is ranged in [0, 1] and indicates how much the data size is reduced after segmentation.

- The third measure is the computing time, denoted by T. To deal with massive time series streams, an online segmentation algorithm definitely has to work as fast as possible.

5.5.2 Characteristics of the Online SWTD

We did experiments on 100 'random walk' time series with 3000 data samples to investigate the performance of SWTD with respect to the threshold R^{2*}_{on} and R^{2*}_{kp}. The initial sliding window $L_{ini} = 100$. Figure 5.11 shows the averaged results over the 100 time series. The plot (a) shows that P_{on} and CR almost keep at the same level as R^{2*}_{on} is increased from -5 to 0.9. This indicates that the threshold R^{2*}_{on} does not control the approximation accuracy and compression rate. However, R^{2*}_{on} controls the frequency of online update. The higher the R^{2*}_{on}, the more segmentation updates are made during online monitoring. Thus the computing time T increases when R^{2*}_{on} increases, as shown in the left plot of Figure 5.11(a). The plot (b) clearly shows that the threshold

R_{kp}^{2*} makes the trade-off between the approximation accuracy and the compression rate directly. As R_{kp}^{2*} increase from 0.1 to 0.9, the approximation accuracy indicator P_{on} is monotonically increasing, and CR is monotonically decreasing. Since a higher R_{kp}^{2*} leads to more segmentation steps in TSTD, and computing time T is increasing with R_{kp}^{2*}.

5.5.3 Performance Demonstration by a Noisy Sensor Signal

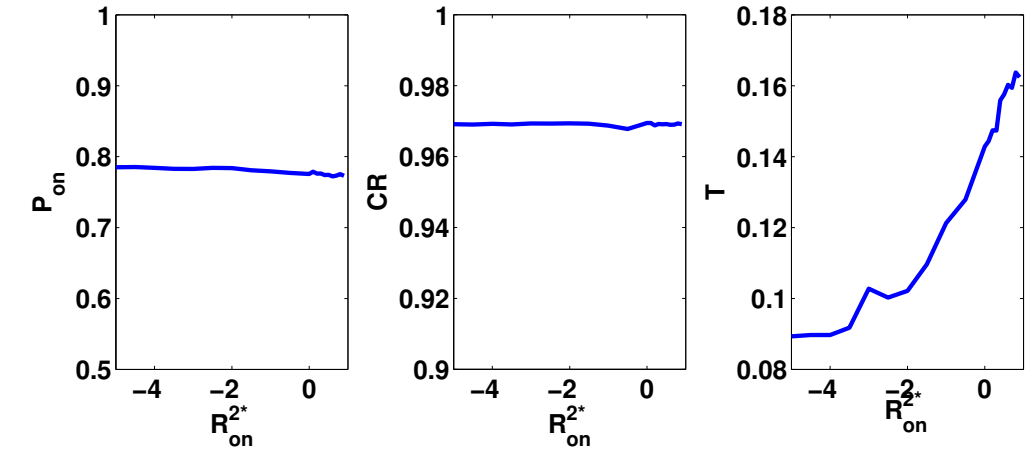
In this section, we demonstrate the segmentation performance of the online SWTD through a sensor signal contaminated heavily by noises. This types of signals are very common in practice collected from various sensors. The segmentation result is shown in Figure 5.14. The proposed SWTD is capable of capturing the major temporal patterns of the time series in presence of the heavy noises. As a comparison, the result of the popular online approach SWAB is also shown in Figure 5.15. The SWAB employs a 'maximum error' threshold. We varied threshold values of 'maximum error', and made it achieve a similar segmentation with the SWTD. Although the approximation accuracy of SWAB (0.68) is a little higher than SWTD(0.66), the SWAB approach took 100 times more time (16.58s) than the SWTD (0.16s). The computationally speed is vital for online monitoring of massive time series data. This example demonstrates the great potential of our proposed SWTD in online real-time applications.

Table 5.2: The online segmentation performance of SW, SWAB, and SWTD.

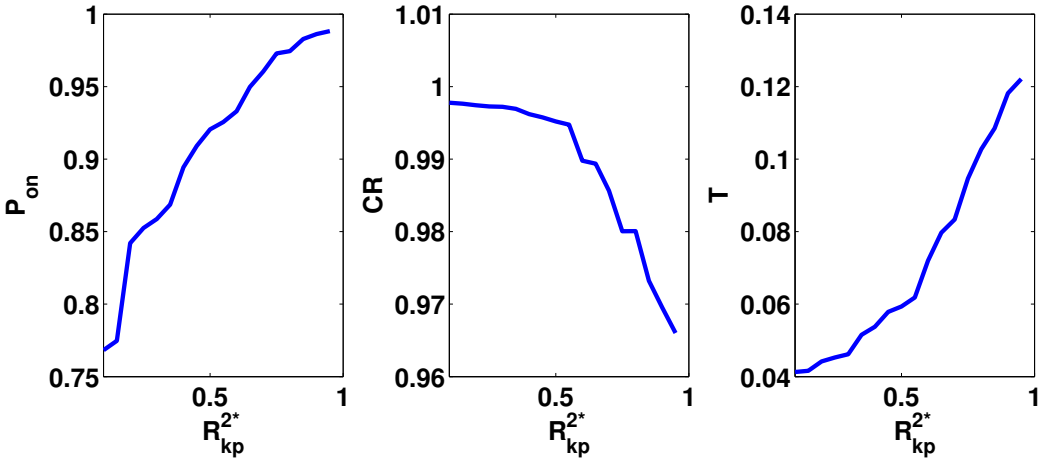
	R_{on}	CR_{on}	T
SW	0.64	113.64	2.47
SWAB	0.68	83.33	16.69
SWTD	0.66	80.65	0.16

5.5.4 Performance Comparisons for Various Time Series

We compare the proposed SWTD with two popular online approaches SWAB and SW, by 24 real-world time series data from various fields, including neurophysiology, industry, medicine, and geography. The data are public available from the UCR time series data archive [76].



(a) P_{on}, CR, and T with respect to R_{on}^{2*} (keep $R_{kp}^{2*} = 0.9$)



(b) P_{on}, CR, and T with respect to R_{kp}^{2*} (keep $R_{on}^{2*} = -1$)

Figure 5.11: The performance measures P_{on}, CR, and T with respect to the thresholds R_{on}^{2*} and R_{kp}^{2*}. The results were averaged over experiments of 100 'random walk' time series with 3000 samples, the initial window size $L_{ini} = 100$. (a) The performance measures P_{on}, CR, and T as R_{on}^{2*} increases from -5 to 0.9 using $R_{kp}^{2} = 0.9$. (b) The performance measures P_{on}, CR, and T as R_{kp}^{2*} increases from 0.1 to 0.9 using $R_{on}^{2} = -1$. The observations: R_{on}^{2*} does not control the approximation accuracy and compression rate, however, it controls the frequency of online update; R_{kp}^{2*} makes the trade-off between the approximation accuracy and the compression rate directly. Most importantly, the compression rate can be automatically adjusted to the analyzed time series. In this example, the CR value is only decreases from 0.99 to 0.96 when R_{kp}^{2*} is increased from 0.1 to 0.9.

Since both SW and SWAB rely on a data-dependent decomposition criterion, called 'maximum error'. To make a fair comparison, we made many numerical experiments to investigate the effects of different settings of the 'maximum error'. Finally, we choose the relative 'maximum error', denoted by E_{max} as the threshold for SWAB and SW.

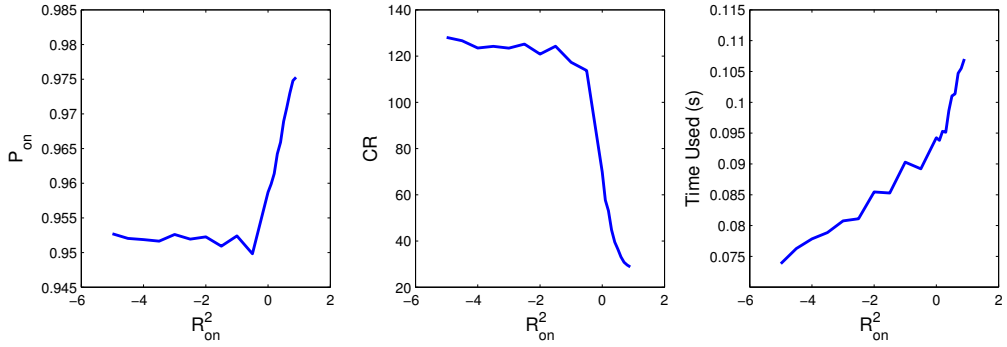


Figure 5.12: The performance of the SWTD with respect to the setting of R^2_{on}. The results were averaged over experiments of 100 'random walk' time series with a length of 3000, and L_{ini} is 500. The red dotted line in each plot represents the performance of the offline algorithm TSTD.

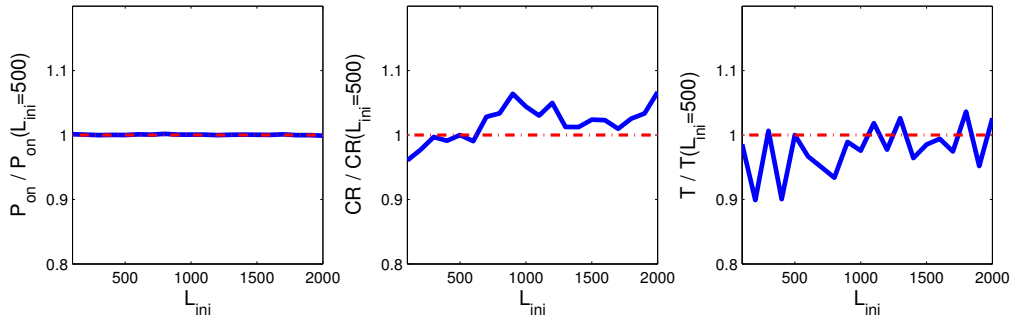


Figure 5.13: The performance of the SWTD with respect to the setting of L_{ini}. The results were averaged over experiments of 100 'random walk' time series with a length of 3000, and the L_{ini} was increased from 100 to 1500. Using the performance at $L_{ini} = 500$ as the reference P^{500}_{on}, CR^{500}, T^{500} , the relative performance of other settings of L_{ini} is calculated by P_{on}/P^{500}_{on}, CR/CR^{500}, and T/T^{500}.

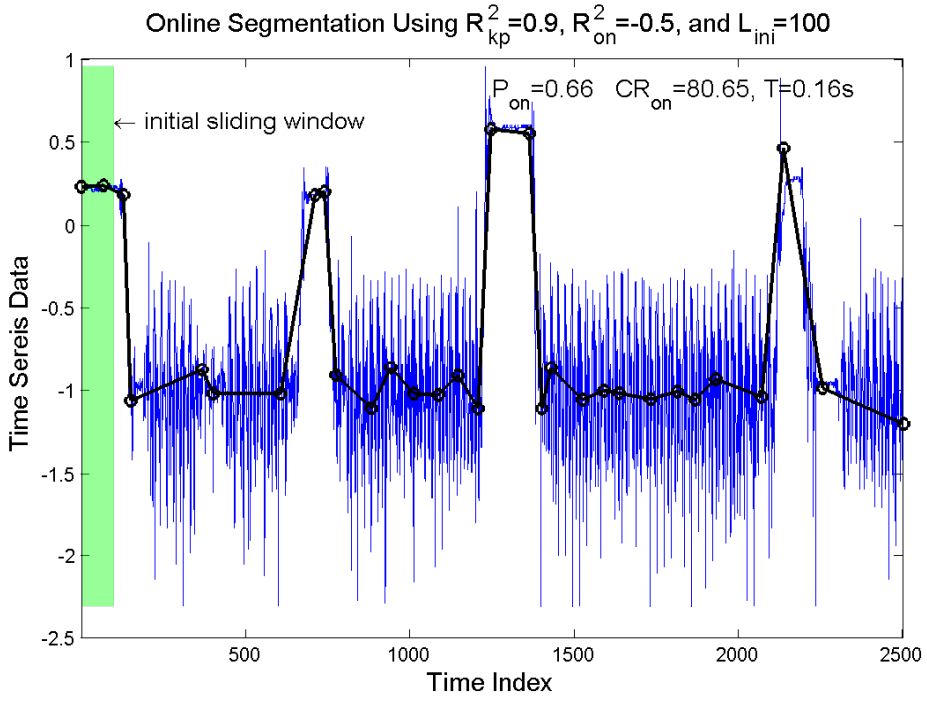


Figure 5.14: The segmentation performance of SWTD on a very noisy sensory signal.

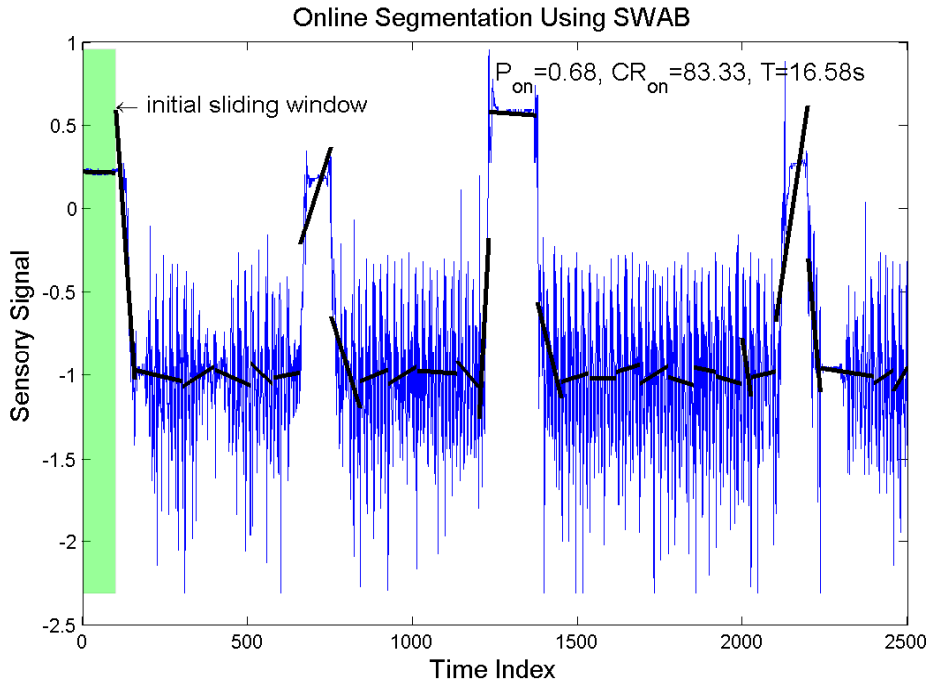


Figure 5.15: The segmentation performance of SWAB on a very noisy sensory signal.

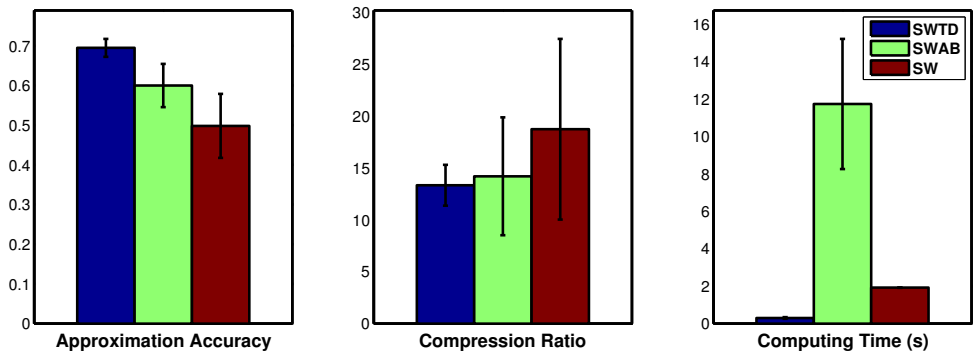


Figure 5.16: Comparison of the approximation accuracy, compression ration, and computing time of the online approaches SWTD, SWAB, and SW.

The relative 'maximum error' E_{max} is defined by the percentage of 'maximum error' to the range of time series values. we used 10 settings of E_{max} from 1% to 10%. These settings are commonly used for SW and SWAB approaches, and the error range is within 10% to avoid very coarse segmentations. The SWTD approach just employs one setting of $R_{kp} = 0.95$ and $R_{on} = -1$ for all the data sets.

The boxplot of the segmentation results of SWTD, SW, and SWAB are shown in Figure 5.17. A detailed results in each time series data are summarized in Table 5.3. Overall, the proposed SWTD has the comparable approximation accuracy and compression rate to those of SWAB and SW. However, the proposed SWTD algorithm works much faster than SWAB and SW. In particular, the median computing times of SWTD, SWAB, and SW are 0.53, 10.57, and 1.91, respectively. The proposed SWTD is about 20 times faster than SWAB, and about 4 times faster than SW. Most importantly, it is very convenient to setup the parameters of the proposed SWTD, and work for various time series data from different fields. On the other hand, the parameters of SWAB and SW are not related to the approximation accuracy directly, and thus require a lot more 'trial and error' process to make a good trade-off between approximation accuracy and compression rate to meet some accuracy and compression requirements.

It is noted that, as demonstrated in Figure 5.18, the piecewise approximation model of SWAB and SW approaches consist of disconnected approximation lines. The disconnected approximation lines reduce the approximation errors considerably; however,

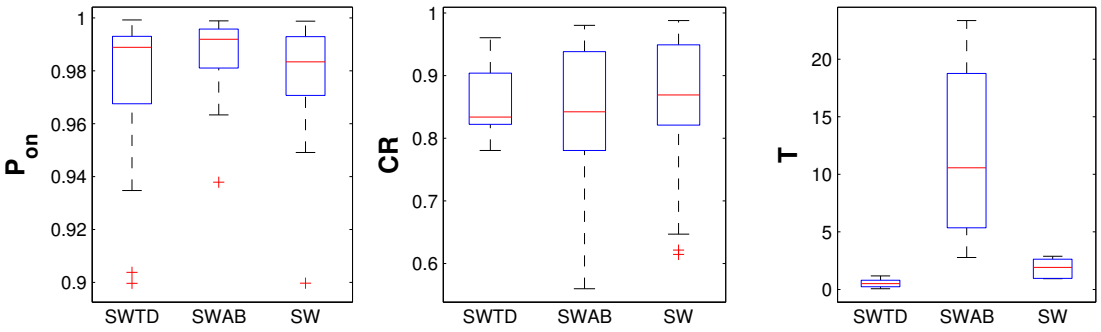


Figure 5.17: The boxplot of P_{on}, CR, and T of SWTD, SWAB, and SW on 24 real-world time series data. Overall, the proposed SWTD has the comparable approximation accuracy and compression rate to those of SWAB and SW. However, the proposed SWTD algorithm works much faster (around 20 time faster than SWAB and 4 time faster than SW). Most importantly, it is very convenient to setup the parameters of the proposed SWTD, and work for various time series data from different fields. We employed only one parameter setting ($R^{2*}_{kp} = 0.95$ and $R^{2*}_{on} = -1$) in this experiment, and achieved a high approximation accuracy for all data sets. On the other hand, the parameters of SWAB and SW are not related to the approximation accuracy directly, and thus require more 'trial and error' process to make a good trade-off between approximation accuracy and compression rate to meet some accuracy and compression requirements.

the resulting model are not perceptually plausible. On the other hand, the proposed SWTD approach is more perceptually reasonable to capture time series patterns than the models with disconnected lines.

5.6 Conclusions

The current segmentation approaches highly rely on some data-specific decomposition strategies, which lead to a tedious parameter tuning procedure in practice. Another bottleneck problem of online segmentation algorithms is the high computational complexity. In this chapter, we propose an online time series segmentation approach that is accurate, fast, and easily applicable to various time series with different scales. In particular, the proposed online segmentation framework has three important features:

- employs a data-independent decomposition strategy, which employs a scaled universal statistical threshold measure to control approximation accuracy directly regardless of data values.

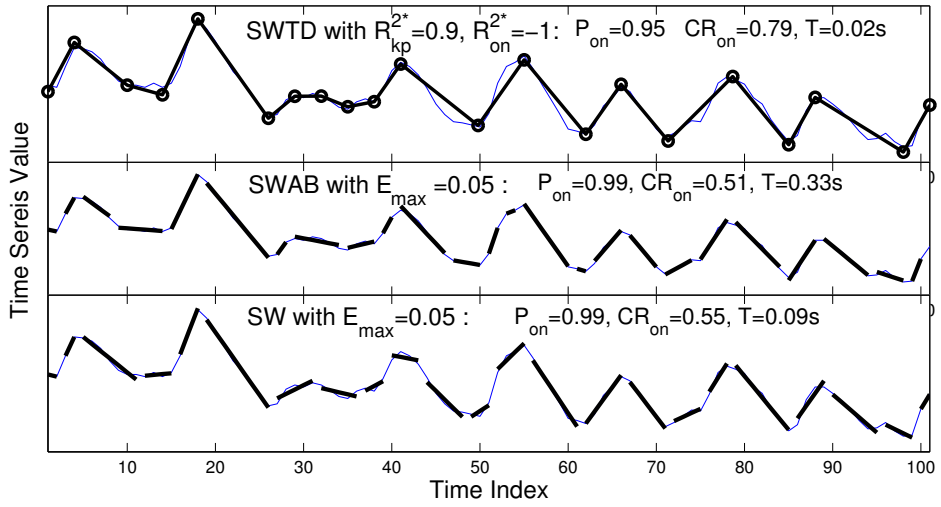


Figure 5.18: A demonstration of the segmentation results of SWTD, SWAB, and SW. Another big advantage of the proposed SWTD is that it can provide a better representation for time series temporal patterns. The piecewise connected-line model of the proposed SWTD is more perceptually reasonable than the approximation models with disconnected lines. The main objective to use disconnected lines is to increase approximation accuracy, however, it sacrifice some temporal pattern information by doing so. Our proposed algorithm is capable of generating similar approximation accuracy while keep a better representation for temporal time series pattern.

- employs a novel two-stage top-down segmentation algorithm, which is capable of achieving a guaranteed approximation accuracy for various time series without a tedious threshold turning process.

- employs a closed-form online updating formulas and achieves a very low computing cost to process massive time series streams online. The complexity of processing a new incoming data point is only $O(1)$.

It is very easy to setup the parameters of SWTD compared with many others that employ data-dependent threshold strategies. We employed only one parameter setting ($R^{2*}_{kp} = 0.95$ and $R^{2*}_{on} = -1$) in the numerical experiments of 24 real-world time series. The experimental results showed that the proposed online SWTD works very fast online while achieving a high approximation accuracy for all data sets with only one parameter setting. The proposed online segmentation approach SWTD has a great potential to work well for online monitoring and processing of highly nonstationary time

Table 5.3: The online segmentation performances of SWTD, SWAB, and SW on 24 real-world time series data sets that are public available at the UCR time series data archive [76].

		P_{on}			CR			T		
Time Series Data		SWTD	SWAB	SW	SWTD	SWAB	SW	SWTD	SWAB	SW
1	ERP 1	0.99	0.98	0.97	0.82	0.84	0.87	1.47	17.82	2.87
2	ERP 2	0.99	0.97	0.96	0.83	0.87	0.89	0.89	20.1	2.86
3	ERP 3	0.99	0.98	0.97	0.81	0.87	0.89	0.95	20.23	2.86
4	EOG	1	1	0.99	0.86	0.96	0.97	0.31	19.7	2.39
5	Steamgen 1	1	0.99	0.99	0.89	0.96	0.96	0.53	23.36	2.86
6	Steamgen 2	0.99	0.98	0.97	0.82	0.86	0.88	1.2	19.95	2.86
7	Steamgen 3	1	1	0.99	0.81	0.9	0.91	1.31	23.01	2.88
8	Foetal ECG 1	0.88	0.96	0.95	0.78	0.79	0.82	1.13	10.47	2.38
9	Foetal ECG 2	0.98	0.98	0.97	0.79	0.84	0.87	0.86	12.5	2.38
10	TOR95	0.96	0.99	0.98	0.79	0.57	0.62	0.77	4.33	1.43
11	Power Data	0.99	0.99	0.99	0.78	0.8	0.83	0.58	10.17	1.91
12	Burst	0.99	0.99	0.99	0.94	0.97	0.97	0.27	17.83	2.4
13	Fluid Dynamics	0.93	0.94	0.9	0.82	0.84	0.87	0.95	14.12	2.38
14	PH Data 1	0.96	1	1	0.93	0.78	0.83	0.36	10.66	1.9
15	PH Data 2	1	1	1	0.84	0.92	0.94	0.22	12.2	1.91
16	Shuttle 1	0.99	1	1	0.93	0.98	0.99	0.11	4.94	0.95
17	Shuttle 2	1	1	0.99	0.88	0.96	0.96	0.2	7.76	0.95
18	Shuttle 3	1	1	0.99	0.89	0.95	0.96	0.28	7.81	0.95
19	Greatlakes 1	0.98	0.99	0.98	0.81	0.58	0.65	0.44	2.87	0.94
20	Greatlakes 2	0.99	0.99	0.98	0.81	0.68	0.73	0.5	3.25	0.94
21	Greatlakes 3	0.89	0.99	0.97	0.82	0.56	0.61	0.52	2.78	0.95
22	Flutter	0.74	0.99	0.98	0.9	0.77	0.83	0.34	4.73	0.98
23	Wool	0.99	0.99	0.98	0.89	0.79	0.83	0.69	6.7	1.72
24	Attas	0.99	0.99	0.99	0.78	0.78	0.81	0.19	5.77	0.98
AVE.		0.97	0.99	0.98	0.84	0.83	0.85	0.63	11.79	1.90
Median		0.99	0.99	0.98	0.82	0.84	0.87	0.525	10.565	1.91

series without a tedious parameter tuning process. In the future, we will explore more potentials of SWTD on nonstationary time series data.

Chapter 6

A General Framework for Online Prediction of Time Series Events

In the previous chapter, we proposed an efficient time series segmentation algorithm to extract key skeleton points of noisy time series. The proposed TDTD and SWTD transform a time series into a much lower dimensionality representation, while the 'big picture' of the temporal patterns largely preserved. The extracted time series skeleton points allows more efficient storage, visualization, and computational analysis.

With the main objective of online prediction in mind, we are dedicated to develop a new framework for online prediction of time series events. In this chapter, we propose a general prediction framework for online prediction of complex time series events from nonstationary multivariate time series data.

6.1 Traditional Time Series Prediction

Traditional time series prediction is to predict the next few values of a time series. There is a huge number of approaches have been developed in this area. The most popular time series modeling and prediction approaches are ARIMA models for stationary processes and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) models for non-stationary time series. Many data mining techniques have also been widely employed in time series prediction analysis, such as unsupervised neural networks [81], and Support Vector Machines (SVM) [48]. Sfetsos and Siriopoulos also proposed a hybrid technique combining clustering and function approximation for time series prediction [143].

6.2 Time Series Pattern Discovery and Event Prediction

For many real-world time series, the predictive patterns are hidden, unobvious, and heavily contaminated with noises, such as brain activity in EEG time series data. It is very difficult to discover hidden time series patterns using the traditional time series analysis methods, such as the well-known ARIMA and GARCH modeling approaches. The traditional time series modeling approaches are mainly applied to predict the upcoming future values of a time series, and do not consider the event-related time series temporal patterns. New approaches are in great need to investigate the online prediction of target events from nonstationary and noisy time series data.

Many researchers have been devoted to apply data mining and machine learning techniques to find patterns for time series event. In general, data mining of time series event is the process of extracting previously unknown and useful event-related patterns from historical data sets and then using the discovered event-pattern information to make accurate future decisions. Berndt and Clifford employed a dynamic programming approach to find time series patterns that match a predefined set of pattern templates [11]. Rosenstein and Cohen employed a time-delay embedding process to find time series patterns given a set of pattern templates. Keogh and Simith [75] applied piecewise linear segmentation to represent time series patterns, and proposed a probabilistic method to find template patterns from time series streams. These types of pattern mining approaches generally require a priori knowledge or template for the interested temporal patterns.

A number of approaches have been proposed to discover unknown and hidden time series patterns without pattern templates. Povinelli and Feng [122] proposed a prediction framework for characterization and prediction of time series events, such as the sudden rise of a stock price. A genetic algorithm was used to find temporal patterns based on a phase-space representation. Sun et al. [153, 152] studied the pattern discovery for multiple time series events in a time series data. The basic idea is to find the frequent temporal patterns that are related to each type of event, and then set up classification rules for pattern discovery.

It is noted that online prediction of complex time series event is still a young research topic with few publications. The current approaches are not convenient to be applied to solve many prediction problems due to the sophisticated parameter tunning process and the expensive computational load for online applications. In this research, we are dedicated to develop new efficient online prediction frameworks for time series events. Given a multivariate time series stream, we want to discover predictive patterns for a specific target event, and then use the discovered patterns to predict future occurrences of the target event. The objective of the proposed research is to create a general online prediction framework for complex time series event, especially for nonstationary multivariate time series streams.

6.3 Problem Statement

The basic idea of online prediction of target events (such as seizure onsets, different mental states, illegal financial transactions, etc.) is to capture specific pre-event patterns that characterize the conditions preceding each target event. The fundamental hypothesis in this research is that there exist pre-event patterns that occur frequently before the target event but occur much less frequently in the periods far from the event. The objective of this research is to discover such pre-event patterns in the process of online monitoring of a time series stream. The 'online' property indicates the prediction is prospective without using any future information, which is very important in practical applications. In summary, the online prediction problem is defined as follows.

Definition 4. Consider a multivariate time series stream

$$X = x_1(t), x_2(t), \ldots, x_n(t), t = 1, 2, \ldots. \tag{6.1}$$

where t is the time index, n is the number of time series. A sliding window of length L_{mw} is applied to monitor the time series X with a step length of L_{step}. Given a target event E and a prediction horizon H, the time series pattern extracted from each sliding

window is used to predict the potential occurrences of the event online. A prediction is made if the extracted pattern of a sliding window is more similar to event-related patterns than non-event patterns. An event is correctly predicted if there is at least one prediction within its preceding prediction horizon. The online prediction problem is also illustrated in Figure 6.3.

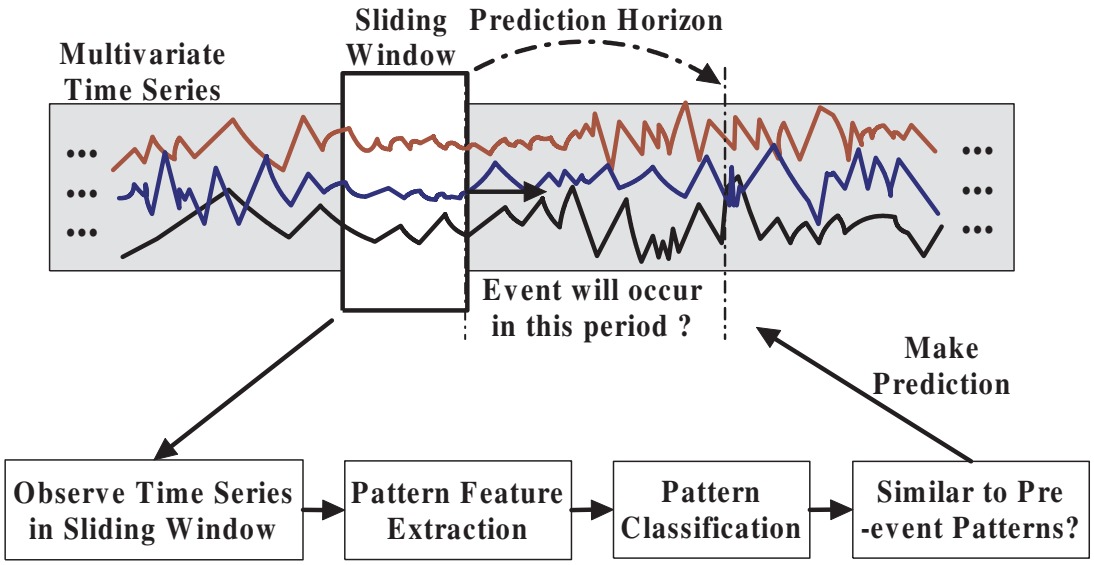


Figure 6.1: Block diagram of online prediction of a time series event.

6.4 Temporal Feature Extraction and Pattern Cluster

The proposed key-turning-point extraction technique is very efficient to deal with noisy time series data and perform dimensionality reduction. In this section, we will propose an efficient high-level pattern representation technique based on the key turning points. By doing so, we are able to represent time series patterns in a very low-dimensional space.

To achieve a higher-level of representation, we summarize the extracted key turning points into four statistical features. In particular, for a time series $X = (x_1, x_2, ..., x_n)$, its key turning points are shown in Figure 6.2. There are six sub-sections, three of

which (segment a, c, e) show increasing trend, and three of which (segment b, d, f) have decreasing trends. The increasing and the decreasing trends indicate the degree of fluctuation of the time series. The following four important features are proposed to represent time series fluctuation patterns:

- Feature 1: accumulated vertical decrease in the segmented piecewise linear time series, which is calculated as

$$F_1 = H(a) + H(c) + H(e), \tag{6.2}$$

  where the function $H(.)$ means the vertical distance from the starting point to the ending point of a sub-segment.

- Feature 2: accumulated vertical increase in the segmented piecewise linear time series, which is calculated as

$$F_2 = H(b) + H(d) + H(f), \tag{6.3}$$

- Feature 3: percentage of the decreasing line segments, which is calculated as

$$F_3 = T(a + c + e)/T(X), \tag{6.4}$$

  where $T(.)$ is the horizontal distance from the starting point to the ending point of a sub-segment.

- Feature 4: range of the time series, which is calculated as

$$F_4 = max(X) - min(X), \tag{6.5}$$

  where $max(X)$ and $min(X)$ means the maximum and minimum values of the segmented time series, respectively.

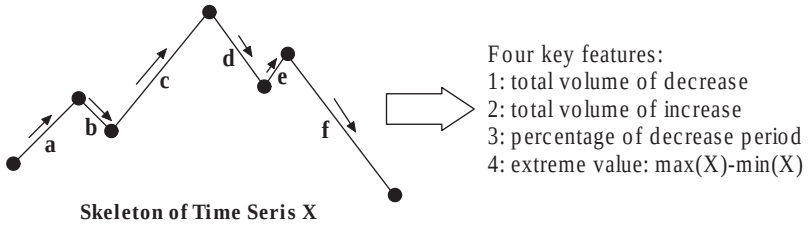


Figure 6.2: Four skeleton-point-based features are employed to represent the temporal fluctuation pattern of a time series.

With the four features F_1, F_2, F_3, and F_4, we partition each feature space into a number of non-overlap intervals. The time series patterns that fall in the same interval in each feature space represent a set of close-by patterns with similar statistical properties. We consider a set of such time series patterns as a pattern cluster. The concept of pattern cluster is illustrated in Figure 6.4. The two time series can be represented by the same pattern cluster, namely 1325.

Using the concept of pattern cluster, one can represent millions or billions of time series patterns by a fixed number of pattern clusters representing groups of similar time series patterns. As shown in the example, there are four features, and each feature space is partitioned into five intervals, then the total number of pattern clusters is only $5^4 = 625$ for a single time series. For multivariate time series, one can concatenate the features of each time series into a big feature vector. For example, if there are two time series, the total number of features becomes $4 \times 2 = 8$, and the total number of pattern clusters becomes $5^8 = 625^2$.

With this new high-level representation technique, we are capable of dealing with numerous complicated time series patterns by a limited number of pattern clusters. This property is really attractive to analyze chaotic nonstationary time series patterns. We do not need to worry about an increasing database of recorded pattern clusters, since the maximum number of possible pattern clusters is known fixed number.

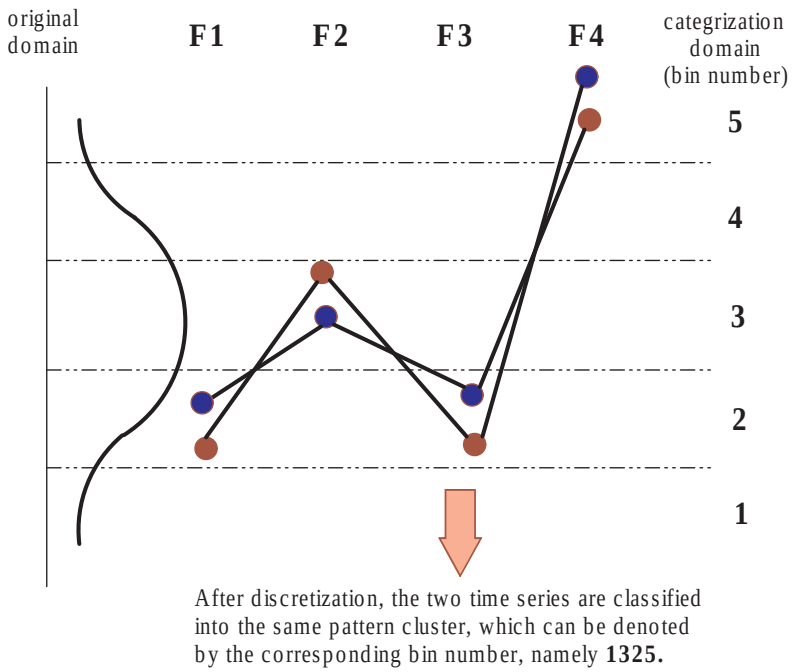


Figure 6.3: A demonstration of the concept of pattern cluster in discretized feature space.

6.5 Adaptive Online Prediction Framework

In this section, we propose a new adaptive online prediction framework for time series event. The proposed prediction framework has the following significant features:

- focuses on the identification of temporal time series patterns that are characteristic of target events.
- employs the newly developed online time series segmentation algorithm SWTD to extract skeleton points of a time series.
- propose a set of feature extraction techniques to extract temporal features from the skeleton points of each time series.
- employs a feature selection technique to determine the feature vector of a time series epoch. That is to determine which feature time series and which temporal features are used to represent a time series epoch.
- build a feature pattern library which stores the feature vector of each monitored

time series epoch. The relationship between the stored patterns and the target events are investigated to construct the online prediction rule.

- proposes two adaptive prediction approaches to investigate the relationship between the stored patterns and the target events based on the pattern library. The prediction decision boundaries of the adaptive prediction schemes are capable of being updated and optimized online after each occurrence of a target event as the system monitors a time series stream over time.

The whole pattern mining and prediction procedure has three stages including feature selection, training stage and testing stage. Figure 6.5 presents the whole framework of the proposed online monitoring and prediction method. Given a prediction goal (target events), a brief outline of the three stages are presented in the following.

- **Feature Selection Stage**. The whole feature selection procedure is illustrated in Figure 6.5. This step is to select the most prominent temporal features that are characteristic of the target event. For each time series, one has many choices of features to characterize the time series. Such as univariate features (e.g., mean, standard deviation, signal power, etc.), bivariate features (e.g., pairwise correlation, pairwise distances, and phase synchronization, etc.), and time-frequency features (e.g., wavelet coefficients, frequency band analysis). We call these types of features of raw time series are **first-level features**. Not all first-level features of all time series are relevant to the target event. Thus we employ feature selection technique to select the most relevant features for the target event. As shown in Figure 6.5, given a training time series with the timing of target events and the assumed pre-event time length (prediction horizon H), the first step is to clean the data. One can perform band-pass filer to remove the low and high frequency noises. Then we employ a sliding window to extract the first-level features for each time series. Now the raw time series are represented by a set of characteristic feature time series. According to the event timing information, we extract time series epoch from pre-event an non-event periods. For each time series of each epoch, we first extract the skeleton points using TSTD algorithm proposed in last

Figure 6.4: Block diagram of the proposed adaptive online monitoring and prediction approach, which has three stages including feature selection, training stage and testing stage.

chapter, then we extract four temporal features from the skeleton points. Each epoch is now can be represented by a long feature vector which concatenates the temporal features in all the first-level feature time series. We employed the Pudil's floating search to select which temporal features of which first-level features have the discrimination power to separate the pre-event and non-event epochs. The outcome of the feature selection stage determines the final form of feature vector of a time series epoch. The feature vector represent the concept of 'pattern' of a time series epoch.

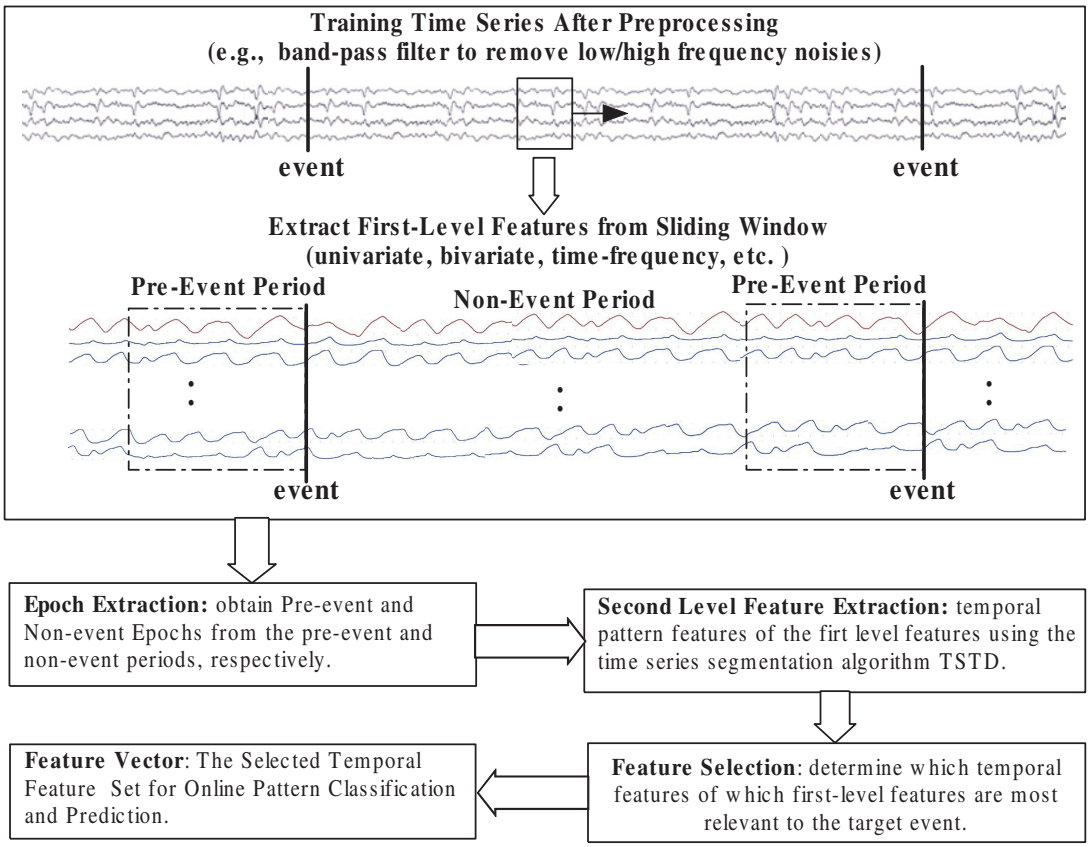


Figure 6.5: Flowchart of the proposed feature extraction and feature selection procedure.

- **Training Stage**. The objective of the training stage is to find the best parameter settings for the online monitoring and prediction framework. There are mainly three types of parameters including the prediction horizon H, the sizes L_{mw} and step lengths L_{step} of the two-level sliding windows. These parameters are generally

unknown in a prediction problem of complex time series event. We employ the training stage to find the most appropriate estimation of the length of pre-event period and the length of the hidden pre-event patterns. The length of pre-event period is estimated by the prediction horizon, and the length of the hidden pre-event patterns is estimated by the length of the second-level sliding window. Given a section of training time series, we apply different settings of prediction horizon, sliding window size and step length. For each parameter setting, we perform the prospective online prediction on the training dataset. The block digram of the prospective online prediction process is shown in Figure 6.5. The settings of H, L_{mw}, and L_{step} with the best prediction performance on the training dataset are selected to perform online prediction on the testing dataset.

- **Testing Stage**. This stage is to perform prospective online prediction on a testing time series using the trained parameter settings. The online monitoring and prediction process is shown in Figure 6.5. The temporal patterns are extracted online by two levels of sliding windows. The first-level sliding window is applied to extract the first-level characteristic features from raw time series. The second-level sliding window is applied to extract temporal patterns of the first-level features using the proposed online time series segmentation algorithm SWTD. The feature vector of each sliding window and its relationship to the target event (pre-event or non-event) are stored in a pattern library. The relationship is also called pattern label in this study. It is noted that the label of each pattern is not obtained at the same time with the feature vector. It is obtained in a retrospective manner using the pre-assumed pre-event period (=prediction horizon H). In other words, the label of a pattern is only known either one prediction horizon later if no event occurs in between or at the moment of event occurrence within one prediction horizon. With the labeled temporal patterns, we designed two data mining techniques to discover the pre-event patterns. The two adaptive pattern identification methods will be discussed in the next two sections. The two adaptive methods share the same property in that the prediction rule (decision boundary/threshold) is automated updated and optimized after each event occurrence. In the online

monitoring process, the system gives a prediction if the pattern vector of a sliding window is classified as 'pre-event' by the prediction rule. If a target event is detected, then the decision boundary is updated and optimized based on the updated pattern library.

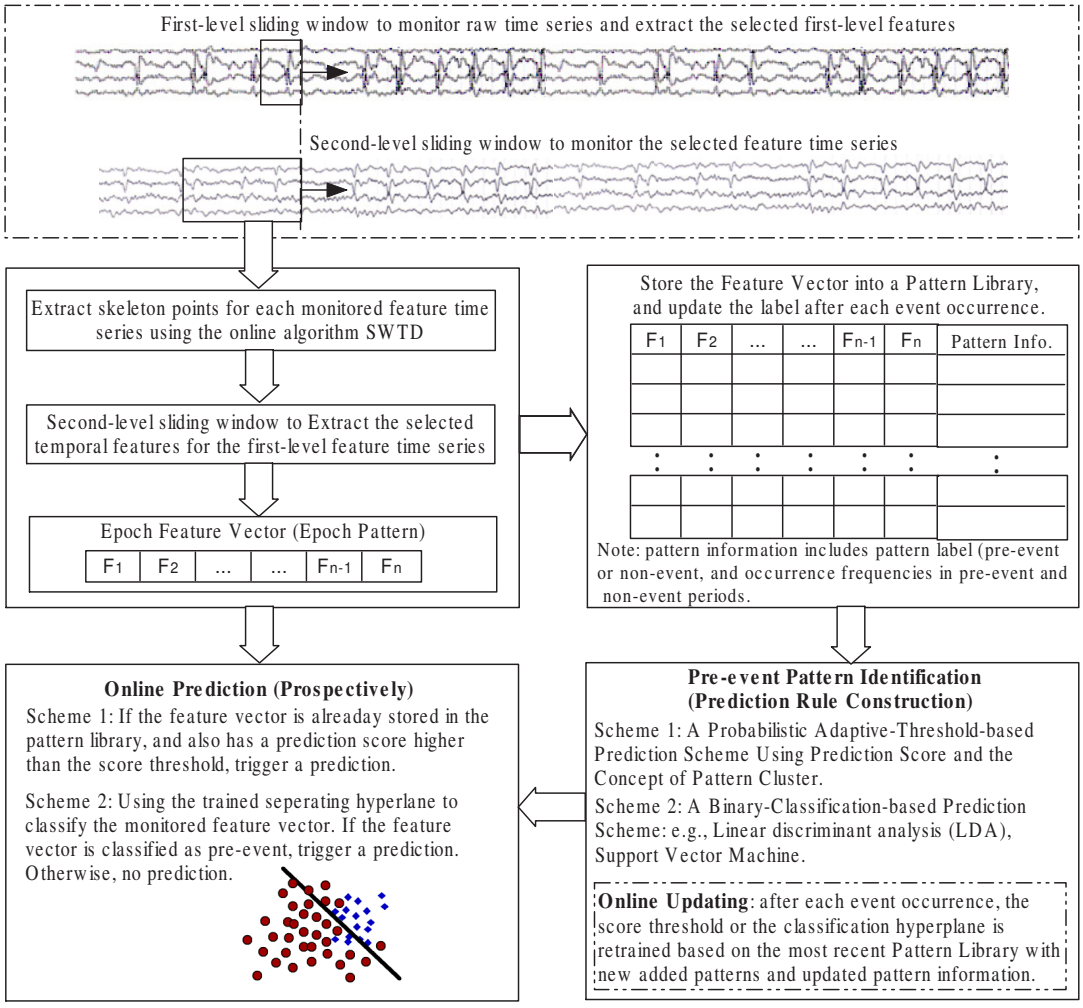


Figure 6.6: Flowchart of the proposed general framework for online monitoring and prediction of a time series target event.

6.6 A Probabilistic Adaptive-Threshold-Based Online Prediction Scheme

In the previous section, we proposed the concept of pattern-cluster to manipulate numerous time series patterns. The pattern cluster representation has a very low dimensionality, and it allows a very efficient storage, visualization, and computational analysis. More importantly, it becomes possible to apply probabilistic theory to analyze the predictability of pattern clusters. In this section, we propose a probabilistic prediction framework to discover the hidden pattern clusters that are predictive to seizure onset. The flowchart of the proposed pattern-cluster based probabilistic online prediction scheme is shown in Figure 6.7.

6.6.1 Definition of Prediction Score

The adaptive learning of the predictive power of the stored pattern clusters is of vital importance in our prediction framework. In this section, we present the probabilistic formula in detail to calculate the predictive score of each pattern cluster.

Definition 5. Given a time series pattern cluster, indexed as the kth cluster in the pattern recording table, its prediction score S_k is defined as follows:

$$S_k = \frac{N_{pre}/N_{tot}}{R_{pre}} \times \frac{N_{pre}^{dist}}{N_{evt}}, \tag{6.6}$$

where N_{pre} is the number of occurrences of the pattern cluster in all monitored pre-event periods; and N_{pre}^{dist} is the number of pre-event periods such that the pattern cluster appears at least once in each of them; N_{tot} is the total number of occurrences of the pattern cluster; and N_{evt} is the total number of events that have occurred. For example, if two events have been monitored, a pattern cluster occurs three times in the first pre-event period, 2 times in the non-event periods, and does not show up in the second pre-event period, then $N_{pre} = 3$, $N_{pre}^{dist} = 1$, $N_{tot} = 5$, and $N_{evt} = 2$. Finally,

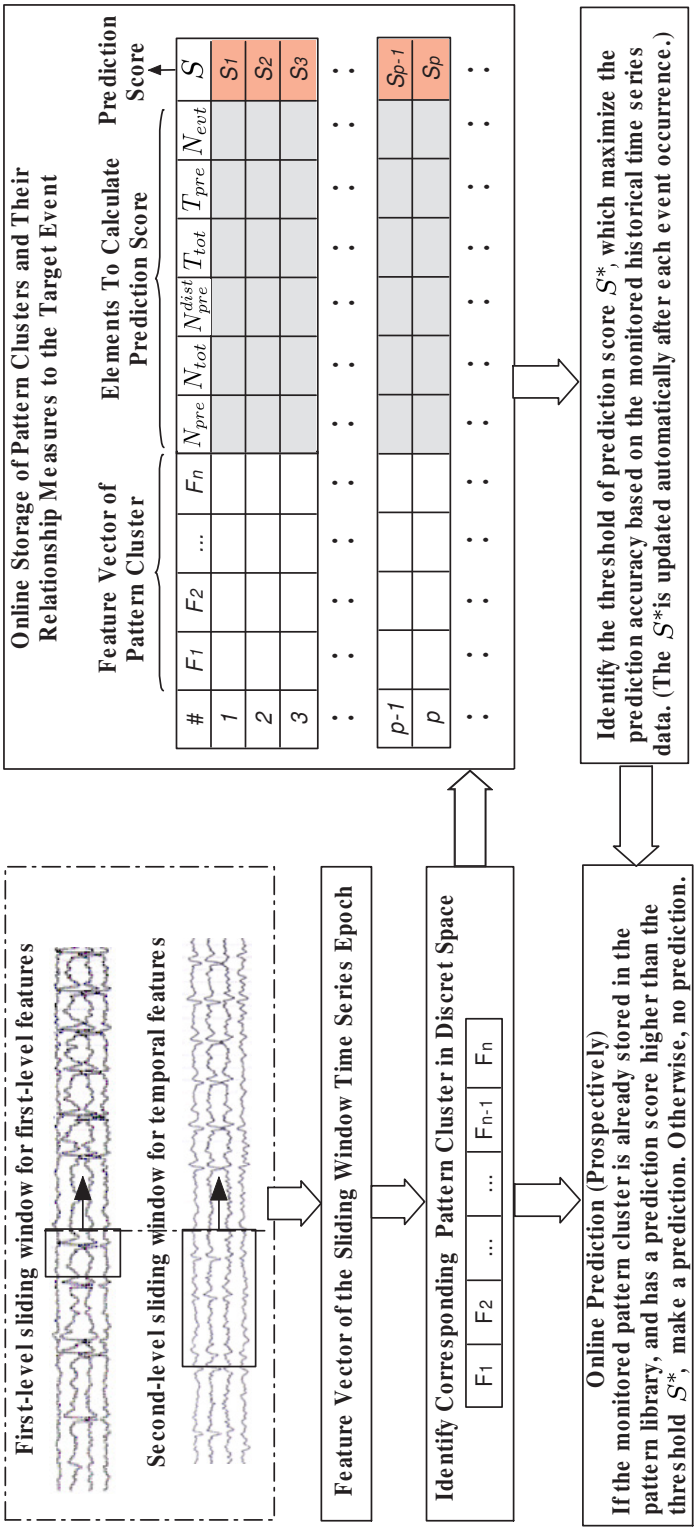


Figure 6.7: Flowchart of the probabilistic adaptive-threshold-based online prediction scheme using the concept of pattern-cluster in discrete feature space.

R_{pre} is the time ratio between pre-event periods and non-event periods. In particular, it is calculated as follows:

$$R_{pre} = \frac{T_{pre}}{T_{tot} - T_{pre}} = \frac{N_{evt} \times T_{hrzn}}{T_{tot} - N_{evt} \times T_{hrzn}}, \tag{6.7}$$

where T_{pre} is the total length of monitored pre-event periods, T_{tot} is the total length of monitored EEG time series; and T_{hrzn} is the length of prediction horizon.

The predictive score proposed in formula 6.6 indicates how strong a pattern cluster is associated with event onset. In particular, in the first term of formula 6.6, the N_{pre}/N_{tot} is the percentage of the pattern cluster appear in pre-event periods. This percentage value is compared with R_{pre} to evaluate if the pattern cluster occurs in pre-event periods at a random level. If the pattern cluster occurs equal-likely in both pre-event and non-event periods, then the expected value of N_{pre}/N_{tot} should be equal to the expected value of R_{pre}. In particular, we can summarize the following properties of the first term of formula 6.6:

- If the pattern is pure random in both pre-event periods and non-event periods, then

$$E(N_{pre}/N_{tot}) = E(R_{pre}). \tag{6.8}$$

- If the pattern occurs more frequently in pre-event periods than the non-event periods, then we have

$$E(N_{pre}/N_{tot}) > E(R_{pre}). \tag{6.9}$$

  The higher the ratio value, the more likely the pattern cluster is associated with event onset.

- If the pattern occurs less frequently in pre-event periods than the non-event periods, then we have

$$E(N_{pre}/N_{tot}) < E(R_{pre}). \tag{6.10}$$

As discussed above, the ratio of N_{pre}/N_{tot} and R_{pre} (the first term in formula 6.6) is an important factor to identify the prediction power of a pattern cluster. However, it is noted that this ratio alone is sometime unreliable and un-robust under some extreme situations. For example, a pattern cluster occurs many times within one prediction horizon (may due to noises or unusual situations), and appears much less frequently or never occurs in other pre-event periods. In such cases, the ratio can be temporally high due to its very high occurrence frequency in only a few pre-event periods. And thus lead to a high predictive score. Although the ratio may return toward its expected value in long-term if N_{nm} could increase over time. However, it may take a long time and many false predictions may have been made during this period due to this 'bad' pattern cluster.

To remedy this limitation, we introduce the second term in formula 6.6, N_{pre}^{dist}/N_{evt}, which considers the percentage of the pattern occurrences in different pre-event periods. Ideally, we assume that a good candidate for prediction should appear in a large portion of the monitored pre-event periods, not only in one or in a few of them. In particular, we expect an ideal predictive pattern cluster should have the following property:

$$\frac{NN_{pre}^{dist}}{N_{evt}} \approx 1, \tag{6.11}$$

which means that the pattern cluster occurs in almost all of the monitored pre-event periods. The multiplication of the first and the second term in formula 6.6 estimates the likelihood of a pattern cluster in pre-event periods and reduces the bad effects of some extreme situations. In general, the higher the prediction score, the higher probability the pattern cluster appears in pre-event periods, and thus the more prominent it is to predict events.

6.6.2 Probabilistic Online Prediction Rule

Figure 6.7 present the structure of the proposed probabilistic online prediction framework. Each time series epoch in the sliding window is represented by a pattern cluster.

The system stores each pattern cluster into the pattern library as well as six measures that are used to calculate its prediction score according to formula 6.6. The higher the prediction score of a pattern cluster, the more likely it is a pre-event pattern cluster. We employ an adaptive threshold on the prediction score to discriminate the pre-event and non-event pattern clusters online. More specifically, the threshold is defined as follows.

Definition 6. The threshold $S*$ is defined as the value that maximizes the prediction performance (sensitivity + specificity) in the monitored historical time series. The threshold $S*$ is updated after each occurrence of a target event.

In the online prediction process, each time series epoch in the sliding window is converted to a pattern cluster. Given a pattern cluster, indexed as the kth cluster in the pattern library, its prediction score is denoted as S_k, then the prediction rule of the probabilistic online prediction scheme is defined by:

$$predictor = \begin{cases} 1, & \text{if } S_k \geq S* \text{ (make an prediction)} \\ 0, & \text{otherwise (no prediction);} \end{cases}$$

6.7 A Classification-Based Online Prediction Scheme

The previous prediction scheme investigate the temporal pattern features in discrete space. In many cases, it might not be convenient to identify a set of appropriate discretization criterion in each feature space. On the other hand, one can also employ the existing data mining techniques to identify pre-event patterns in the pattern library. In this study, we employed the popular binary classification technique, Fisher's Linear Discriminant Analysis, to classify the pre-event and non-event patterns in the pattern library. The flowchart of the LDA-based online prediction scheme is shown in Figure 6.8.

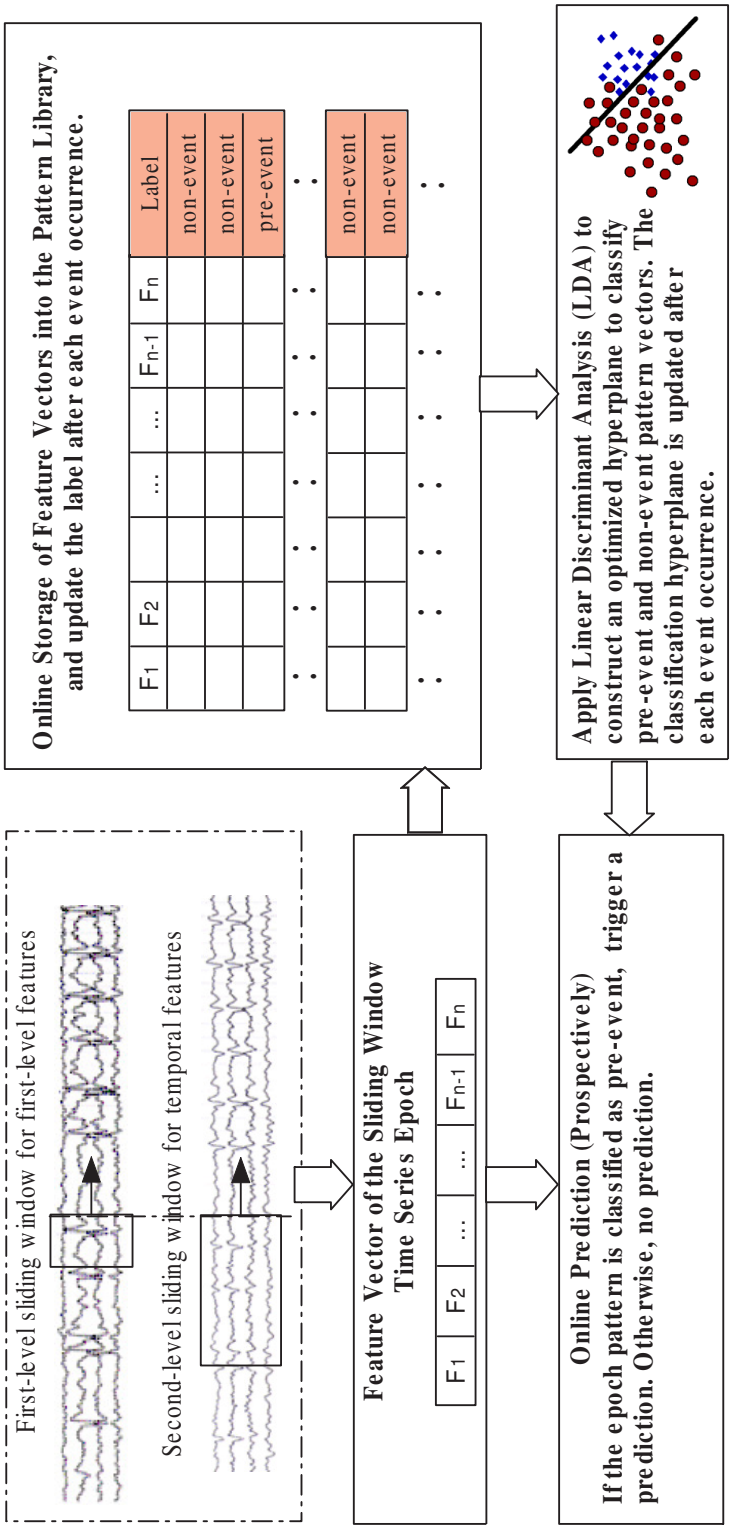


Figure 6.8: Flowchart of the LDA-based online prediction scheme.

6.7.1 Fisher's Linear Discriminant Analysis

Fisher's LDA aims to find an optimal projection by minimizing the intraclass variance and maximizing the distance between the two classes simultaneously [46]. Mathematically, LDA tries to find an optimal direction $\omega^* \in R^{n\times k}$ as a solution of the following optimization problem:

$$\omega^* = argmax_\omega \frac{\omega^T S_b \omega}{\omega^T S_{\omega^*} \omega}, \tag{6.12}$$

where ω is the direction of the hyperplane that is used to separate the two data sets. S_b and S_ω are the interclass and intraclass covariance matrix, respectively. They are defined as follows

$$S_b = (m_1 - m_2)^T (m_1 - m_2), \tag{6.13}$$

$$S_\omega = \sum_{i\in 1,2} \sum_{i\in D_i} (Y_i - m_i)^T (Y_i - m_i), \tag{6.14}$$

where m_1 and m_2 are the means of the feature vectors Y in the two data sets D_1 and D_2, respectively. They can be calculated by

$$m_1 = \frac{1}{p} \sum_{Y\in D_1} Y = \frac{1}{p} \sum_{i=1}^{p} Y_i, \tag{6.15}$$

$$m_2 = \frac{1}{q} \sum_{Y\in D_2} Y = \frac{1}{p} \sum_{i=p+q}^{p+1} Y_i. \tag{6.16}$$

When S_ω is not singular, the above optimization problem can be solved by applying the eigen-decomposition to the matrix $S_\omega^{-1} S_b$. The eigenvector corresponding to the largest eigenvalue forms the optimal direction w^* by

$$\omega^* = S_\omega^{-1} (m_1 - m_2). \tag{6.17}$$

When S_ω is singular, an identity matrix with a small scalar multiple can be used to

tackle this problem [102]. The optimal w^* then becomes

$$\omega^* = (S_\omega + \lambda I)^{-1}(m_1 - m_2). \tag{6.18}$$

Once ω^* is obtained, the optimal decision boundary of LDA can be represented by

$$\omega^{*T} Y + b = 0, \tag{6.19}$$

where b is the bias term. There is no general rule to determine the bias term, a most commonly used bias term is $b = -\omega^{*T}(m_1 + m_2)/2$. The class of a feature vector Y depends on which side of the hyperplane it is on. In particular, given a feature vector Y_{new}, the prediction rule is as follows

$$\begin{cases} \omega^{*T} Y_{new} + b > 0, & l_{new} = 1 \text{ (pr-event pattern)}, \\ \omega^{*T} Y_{new} + b < 0, & l_{new} = -1 \text{ (non-event pattern)}. \end{cases}$$

6.7.2 LDA-based Online Prediction Rule

Figure 6.8 presents the structure of the LDA-based online prediction scheme. Each time series epoch in the sliding window is represented by a feature vector in countinous feature space. The system stores each pattern vector into the pattern library as well as its class label (pre-event or non-event). The pattern library contains feature vectors of two classes. Thus we can formulate the problem as a typical binary classification problem. That is to find an optimal hyperplane to separate pattern vectors of the two classes with highest accuracy. The trained LDA hyperplane is then used to classify a feature vector of a sliding window online.

In the online prediction process, each time series epoch in the sliding window is represented by a feature vector. Given a feature vector X_k stored in the kth row of the pattern library, then the prediction rule of the LDA-based online prediction scheme is

defined by:

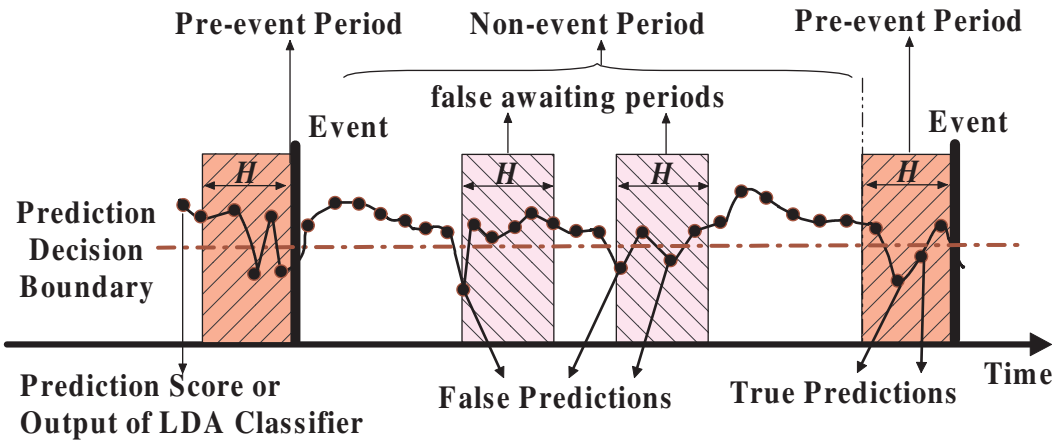
$$predictor = \begin{cases} 1, & \text{if } \omega^{*T} X_k + b > 0 \\ 0, & \text{if } \omega^{*T} X_k + b \leq 0; \end{cases}$$

6.8 Evaluation of Prediction Performance

The most commonly used prediction performance measures are specificity and sensitivity. However, the traditional definition of specificity and sensitivity only focus on the correctness of each individual prediction, and do not consider prediction horizon and event information. They are inappropriate to be applied to measure prediction performance directly for the online event prediction problem, which has to consider the effects of prediction horizon. In this study, we formulate sensitivity and specificity by considering the time effects of prediction horizon, and make them more appropriate to measure the realistic prediction performance in real-life applications.

In this studies, sensitivity, denoted as sen_{blk} is defined as the number of correctly predicted events divided by the total number of events. An event is considered to be correctly predicted if there is at least one true prediction within its preceding prediction horizon.

To estimate the prediction specificity, many event-prediction studies employed the measure of false prediction rate, which is defined by the number of false predictions per hour (or unit time). However, false prediction rate also ignore the time effects of prediction horizon on the prediction performance. For example, given the same false prediction rate, an algorithm with a 1-hour prediction horizon will give a patient much longer false awaiting time than the one with a 10-minute prediction horizon. To overcome this bias, Mormann et al. [104] suggested that a prediction specificity can be estimated by quantifying the portion of time during the non-event period that is not considered to be false awaiting time. We herein employed this definition of specificity, denoted as spe_{blk}. A demonstration of the sen_{blk} and spe_{blk} quantification is shown in Figure 6.9.



Prediction Performance Measures

***specificity** = The portion of non-event period that is not in false awaiting state.*

***sensitivity** = The portion of the corrected predicted events, each of which has least one true prediction one predition horizon preceding the event occurrence.*

Figure 6.9: A demonstration of the definition of time-block-based sensitivity (sen_{blk}) and specificity (spe_{blk}) for event-prediction problems which have to consider the time effects of prediction horizon in real-life applications.

6.9 Summary of the Online Prediction Framework

This chapter presents a general online monitoring and prediction framework for time series event. The proposed prediction framework has the following important properties:

- propose a feature selection stage to select the event-related first-level characteristic features of raw time series.
- propose a two-sliding-window approach for online monitoring time series and temporal feature extraction. The first-level sliding window extracts the selected first-level characteristic features from raw time series; and the second-level sliding window extracts the temporal patterns of the first-level features.
- propose a pattern-library approach to store the window-monitored time series patterns and some statistics of their occurrence history related to a target event (such as occurrence frequency in pre-event and non-event period, occurrence spectrum in different pre-event periods.)

- propose an adaptive probabilistic online pattern-discovery and prediction scheme based on pattern clusters in discrete feature space. A probabilistic formula is proposed to estimate the pre-event likelihood (prediction score) of each stored pattern based statistics of its occurrence history. An optimized score threshold that maximizes the prediction performance over the monitor history is identified to discriminate pre-event and non-event patterns. The score threshold is re-optimized after each occurrence of a target event. A big advantage of this approach is that the size of pattern-library is limited by the maximum number of pattern cluster. The drawback is that some efforts are needed to find the most appropriate discretization criterion in each feature space.

- proposed an adaptive classification-based online pattern-discovery and prediction scheme in continuous feature space. In this scheme, the pattern library only stores the monitored feature vectors and their class labels (pre-event or non-event). The the pattern-discovery problem can be formulated as a typical classification problem. A classification technique (such as LDA) is employed to to construct an optimal hyperplane to classify the two classes of feature vectors. The hyperplane is retrained after each occurrence of a target event. The most advantage of this scheme is that optimization and data mining techniques can be employed to find the optimized hyperplane in continuous feature space. However, one drawback of this approach is that the size of pattern-library keeps on increasing over time. One solution to this problem is to use the recent monitored patterns, and discard the far away ones.

In the next chapter, we will apply the proposed adaptive online prediction framework to solve two challenging real-world prediction problems.

Chapter 7

Real-world Applications

In this chapter, we apply the proposed adaptive online monitoring and prediction framework to solve two challenging real-world event-prediction problems based on EEG time series data. The proposed new prediction approach generated superior prediction performance over the existing data mining techniques. The genetic structure of the online monitoring and prediction framework enable it to be applicable to a wide range of time series data. The time series temporal feature extraction and prediction approaches introduced by this dissertation are fundamental contributions to the fields of time series data mining, especially for the analysis of non-stationary chaotic time series.

7.1 Adaptive Online Prediction of Epileptic Seizures

In Chapter 4, we have proposed a reinforcement learning-based online prediction framework for epileptic seizure. In this chapter, we will also evaluate the new adaptive prediction framework by this challenging problem.

7.1.1 Computational Settings

The proposed two prediction schemes have been tested on the EEG recordings of 10 patients with epilepsy using three choices of prediction horizons, seven choices of window length, and seven choices of step length. The complete parameter settings of the prediction system are summarized in Table 7.1.

A brief outline of the experimental setup is as follows:

- feature extraction: 29 first-level features are extracted for each raw time series epoch in the first-level sliding window. In particular, 26 of them are univariate

Table 7.1: Computational settings of the online prediction framework for epileptic seizure prediction.

Parameter Setting	Setting Choices
Prediction Horizon	30, 90, 150 minutes
1st-level sliding window (monitor raw time series)	window size: 10 minutes moving step length: 1 minutes
2nd-level sliding window (monitor feature time series)	window size: 15, 30, 60, 90, 120, 150, 180 minutes window size: 1, 3, 6, 9, 12, 15, 18 minutes
Online Prediction Scheme	1. Adaptive Probabilistic Prediction Scheme 2. Adaptive LDA-based Prediction Scheme
Feature Selection Method	Pudil's floating search based on 1-Nearest Neighbour leave-one-out classification performance.
1st-level features	1-26: Lyapunov exponents of 26 channels of raw EEG 27: averaged pair-wise Euclidean distances 28: averaged pairwise T-ststistics 29: averaged pairwise correlations.
2nd-level features (temporal pattern feature)	1. accumulated vertical increase 2. accumulated vertical decrease 3. percentage of decline periods 4. amplitude range

features, we extract the largest Lyapunov exponent from each of the 26 EEG channels. Three of them are bivariate features, they are averaged pairwise Euclidean distance, T-ststistc, and Pearson correlation, respectively. And four temporal features are extracted for the time series of each first-evel feature. Thus each raw time series epoch is converted into a $29 \times 4 = 106$ temporal feature candidates.

- feature selection: we employ the popular feature selection approach Pudil's floating search to select the optimal subset of temporal features. The criterion of feature selection is the 1-Nearest Neighbour leave-one-out classification performance. The selected optimal feature subset has the highest leave-one-out classification accuracy. In this study, we select 8 most important temporal features from the 106 candidates.

- pattern cluster formulation: we discretized each feature space into five equal bins. Since each time series epoch is transformed into a feature vector of 8 selected features, the total number of possible pattern clusters are $5^8 = 390625$. Thus the maximum size of the patter library is 390625×8.

- performance measure: prediction performance is evaluated by the time-block-based sensitivity sen_{blk} and the time-block-based specificity spe_{blk}. Both of them have been defined and discussed in the previous chapter. The overall prediction accuracy (PA) is defined as the average of sen_{blk} and spe_{blk}. That is $PA = (sen_{blk} + spe_{blk})/2$.

- training and testing: For each patient, the EEG recordings were divided into training and testing dataset. The training dataset is the EEG recordings that contain the first half of seizure occurrences. It is used to perform feature selection and train the best parameter settings of H, L_{mw}, and L_{step}. The testing dataset is the EEG recordings that contain the second half of seizure occurrences. It is used to test our prediction approach ***prospectively*** using the best parameter setting found from the training dataset. The best parameter setting is the one with the highest prediction accuracy. In addition, to find the most appropriate trade-off between sensitivity and specificity, we also added a constraint that the sen_{blk} must be greater than 0.6, and the spe_{blk} must be greater than 0.5. If none of the settings meet this constraint, we simply selected the one with the highest value of prediction accuracy.

7.1.2 Prediction Performance of The Adaptive-Threshold-Based Prediction (ATP) Scheme

Table 7.2 summarizes the training and testing prediction performance of the ATP scheme for the three prediction horizons. To demonstrate the effectiveness of the adaptive prediction scheme, the prediction performance of a non-update scheme and two random prediction schemes (periodic and Poisson) are also reported in the table. The 'non-update' scheme employed the trained threshold obtained from training data, and kept the threshold unchanged in the testing dataset. The prediction periods of the periodic and Poisson schemes for each patient are equal to the averaged length of inter-seizure intervals of the patient. The proposed adaptive scheme ATP achieved much

better prediction performance than the non-update scheme and the two random predictiors. With the best parameter settings (sliding window width and step length), the overall prediction accuracies (PA) of the proposed prediction framework ATP for prediction horizons of 30, 90, and 150 minutes are 78%, 71%, and 66%, respectively. The PAs of the non-update scheme and the two random preditors are all less than 60%. This strong contrast indicates that the proposed ATP scheme was indeed effective to improve the prediction performance online over time. In addition, we notice that the prediction horizon of 30 minutes generated the best prediction performances than the other two horizon choices. This observation implies that the proposed ATP scheme is promising to achieve high prediction accuracy using a short prediction horizon, and thus provide early warnings accurately and timely. Figure 7.1, Figure 7.2, and Figure 7.3 show the prediction outcome of the ATP prediction scheme for patient 10, 4, and 2, respectively, using the best training parameter settings. The adaptive threshold and the prediction alarms are also shown in the Figures.

Our previously developed reinforcement-learning-based prediction scheme achieved an overall prediction accuracy of 70%. The new ATP scheme increased the overall prediction accuracy by 8%. The outcome of this study confirmed that the proposed adaptive-threshold sheme is effective to predict complex time series event from nonstationary chaotic time series data.

7.1.3 Prediction Performance of The Adaptive-LDA-Based Prediction (ALP) Scheme

Table 7.3 summarizes the training and testing performance characteristics of the ALP scheme for three prediction horizons, respectively. The ALP scheme generated very promising prediction using the prediction horizon of 30 minutes, which has an overall prediction accuracy of 91%. While the overall prediction accuracies using the prediction horizons of 90 and 150 minutes are 82% and 73%, respectively. The prediction performance of the adaptive scheme ALP is much better than those of the non-update scheme and the two random preditors. Compared with the state-of-the-art seizure prediction

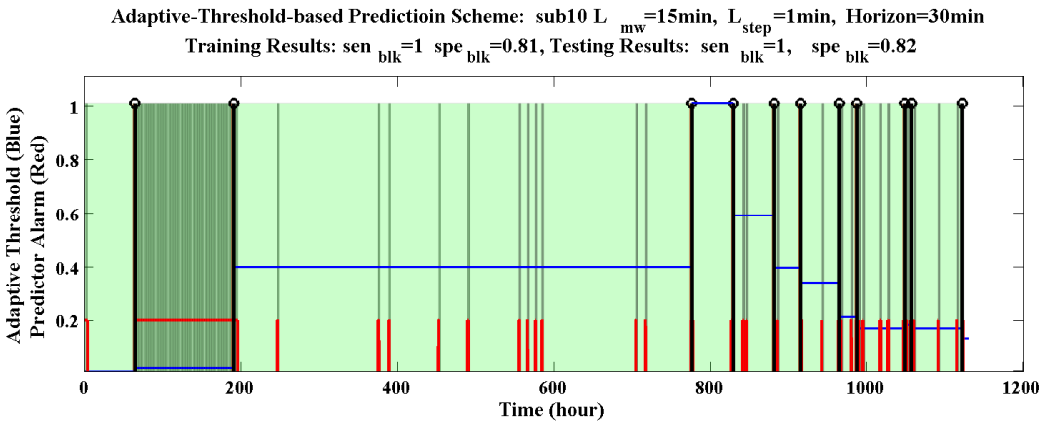


Figure 7.1: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 10 using a prediction horizon of 30 minutes with L_{mw} = 15 minutes and L_{step}= 1 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The piecewise horizontal line represents the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate' prediction alarms.

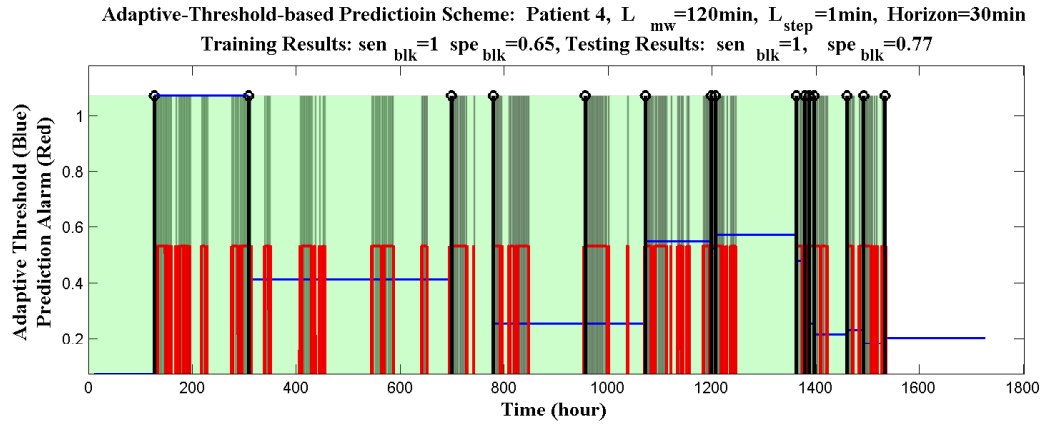


Figure 7.2: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 4 using a prediction horizon of H=30 minutes with L_{mw} = 15 minutes and L_{step}= 12 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The piecewise horizontal line represents the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

Table 7.2: The training and testing performance characteristics of the adaptive-threshold-based ATP prediction framework for three prediction horizons, respectively. The 'Non-Update' scheme employed the trained threshold of prediction score, and kept the threshold unchanged in the testing dataset. The prediction performance on the testing dataset is presented in the table. The prediction performances of two random prediction schemes (periodic and Poisson) are also reported. The prediction periods of the periodic and Poisson schemes for each patient are equal to the averaged length of inter-seizure intervals of the patient.

		Setting		Training		Testing		Non-Update		Possion		Periodic	
Horizion	Patient	L_{mw}(min)	L_{step} (min)	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}
30 min	1	15	1	1.00	0.87	1.00	0.53	0.00	1.00	0.00	0.53	0.00	0.53
	2	30	1	1.00	0.39	1.00	0.66	0.00	1.00	0.00	0.35	0.00	0.35
	3	60	15	0.90	0.56	1.00	0.45	0.33	0.69	0.17	0.52	0.17	0.52
	4	120	1	1.00	0.65	1.00	0.77	0.00	1.00	0.00	0.72	0.07	0.72
	5	60	9	0.86	0.64	1.00	0.63	1.00	0.58	0.00	0.39	0.00	0.39
	6	120	1	0.75	0.77	1.00	0.40	0.75	0.74	0.13	0.34	0.00	0.34
	7	30	15	0.75	0.80	1.00	0.41	0.00	1.00	0.05	0.52	0.00	0.52
	8	15	15	0.86	0.38	0.83	0.56	0.00	1.00	0.06	0.77	0.12	0.77
	9	15	3	0.89	0.76	1.00	0.34	1.00	0.63	0.05	0.95	0.05	0.95
	10	15	1	1.00	0.81	1.00	0.82	1.00	0.08	0.00	0.70	0.09	0.70
	Ave.			0.89	0.66	0.97	0.61	0.38	0.70	0.06	0.96	0.06	0.96
	PA			0.78		0.79		0.54		0.51		0.51	
90 min	1	15	1	1.00	0.69	1.00	0.84	0.00	1.00	0.00	0.47	0.33	0.47
	2	30	1	1.00	0.38	0.67	0.45	0.00	1.00	0.00	0.17	0.00	0.17
	3	120	1	0.90	0.32	1.00	0.30	0.00	1.00	0.30	0.15	0.39	0.15
	4	90	1	1.00	0.44	1.00	0.68	0.00	1.00	0.07	0.39	0.07	0.39
	5	30	3	0.71	0.72	1.00	0.41	1.00	0.39	0.00	0.15	0.00	0.15
	6	90	1	1.00	0.28	1.00	0.39	1.00	0.41	0.25	0.00	0.25	0.00
	7	90	1	0.88	0.57	1.00	0.17	0.00	1.00	0.21	0.22	0.21	0.22
	8	120	18	0.86	0.22	1.00	0.18	1.00	0.46	0.18	0.57	0.18	0.57
	9	120	3	0.89	0.56	0.89	0.43	1.00	0.32	0.11	0.79	0.16	0.79
	10	15	1	1.00	0.72	1.00	0.61	1.00	0.07	0.18	0.24	0.18	0.24
	Ave.			0.90	0.52	0.96	0.49	0.49	0.58	0.15	0.88	0.19	0.88
	PA			0.71		0.73		0.54		0.52		0.54	
150 min	1	15	1	1.00	0.59	0.67	0.78	0.00	1.00	0.33	0.56	0.33	0.56
	2	30	1	1.00	0.39	1.00	0.33	0.00	1.00	0.00	0.10	0.00	0.10
	3	90	1	0.90	0.16	1.00	0.34	0.00	1.00	0.70	0.25	0.65	0.25
	4	90	1	1.00	0.47	1.00	0.65	0.00	1.00	0.13	0.25	0.13	0.25
	5	30	1	0.57	0.58	1.00	0.40	1.00	0.40	0.13	0.09	0.00	0.09
	6	30	1	0.75	0.76	0.75	0.50	1.00	0.38	0.13	0.17	0.13	0.17
	7	180	1	0.88	0.47	1.00	0.12	0.00	1.00	0.37	0.19	0.37	0.19
	8	150	15	0.86	0.16	1.00	0.18	0.00	1.00	0.24	0.47	0.24	0.47
	9	15	1	1.00	0.12	1.00	0.55	1.00	0.32	0.32	0.49	0.26	0.49
	10	15	1	1.00	0.66	1.00	0.46	1.00	0.07	0.09	0.25	0.27	0.25
	Ave.			0.89	0.43	0.96	0.47	0.34	0.63	0.29	0.82	0.28	0.82
	PA			0.66		0.72		0.49		0.56		0.55	

approaches, an overall prediction accuracy around 90% using a prediction horizon of 30 minutes is very attractive. The significant experimental reslts confirmed that the proposed ALP prediction sheme is very effective to achieve online prediction of time series events. The adaptive property of the prediction structure makes it very convinient to achieve persionalized seizure prediction in real clinical applications.

Figure 7.5, 7.6, and 7.7 show the prediction outcomes of the ALP prediction scheme for patient 9, 3, and 1, respectively, using the best training parameter settings.

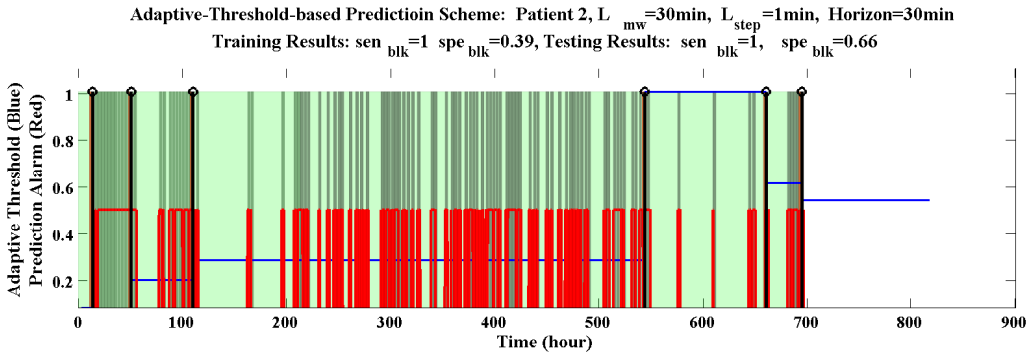


Figure 7.3: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 2 using a prediction horizon of H=30 minutes with L_{mw} = 30 minutes and L_{step}= 1 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The piecewise horizontal line represents the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

7.2 Online Prediction of A Mental State in A Simulated Driving Environment

In this section, we apply the proposed adaptive online prediction framework to solve another challenging prediction problem. A virtual-reality scene was created to simulate the real-world driving experience in big cities. Each subject is provided with a map which is city map in the simulation. Each subject start from the same place, and is asked to go to an objective destination pointed out in the map. The map is placed beside the simulation screen while the subject is driving. The function of the map is very similar to a GPS map in real driving environments. The subject will look at the map from time to time in case he/she was uncertain about the following driving route, for example, if turn left, turn right or go straight ahead at the next intersection. There are 24 subjects were recruited in the simulated driving experiment, their EEG time series data were recorded during their driving.

The occurrence timing of two important events were recorded. They are

- event I: a subject begins to look at the map.
- event II: a subject looks back to the driving screen from the map.

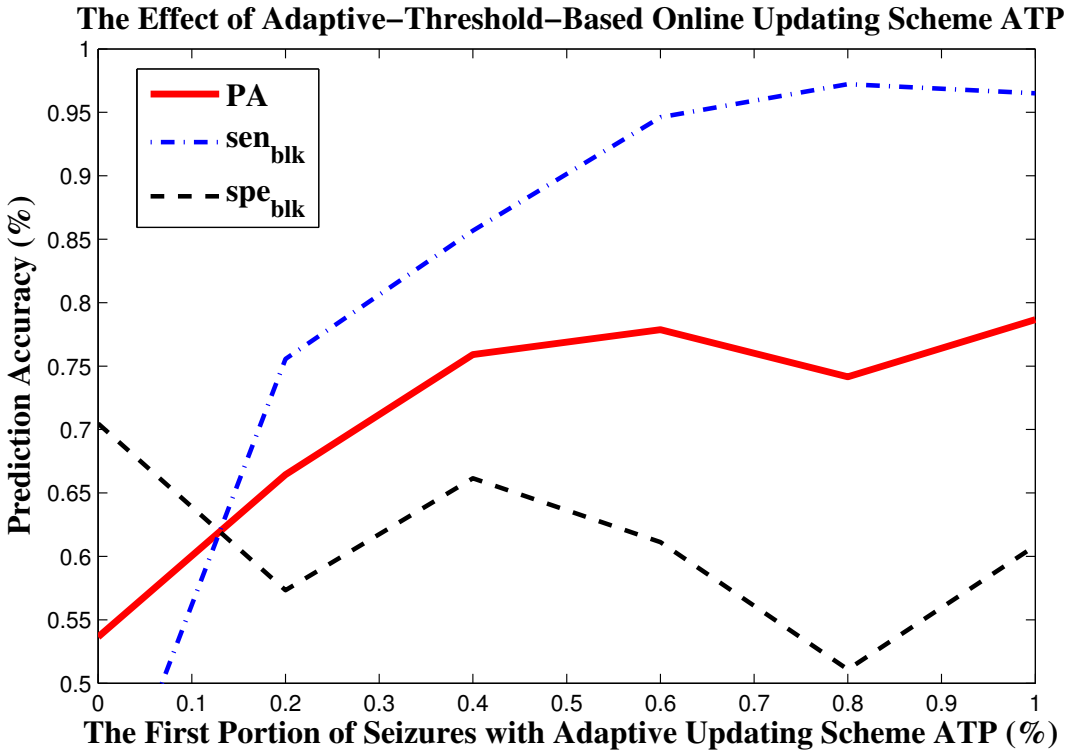


Figure 7.4: The effectiveness of the adaptive online updating scheme ATP. The ATP scheme was only performed on the EEG with the first portion of seizures, and the obtained score threshold was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ATP scheme was actively performed. The point 0 indicates that the initial classification hyperplane of LDA was unchanged throughout the prediction process; and the point 1 means that the LDA classification hyperplane was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more seizures were used to train the LDA classifier. The strong increase trend of prediction accuracy indicates that the adaptive updating scheme ATP is effective to increase online prediction performance over time.

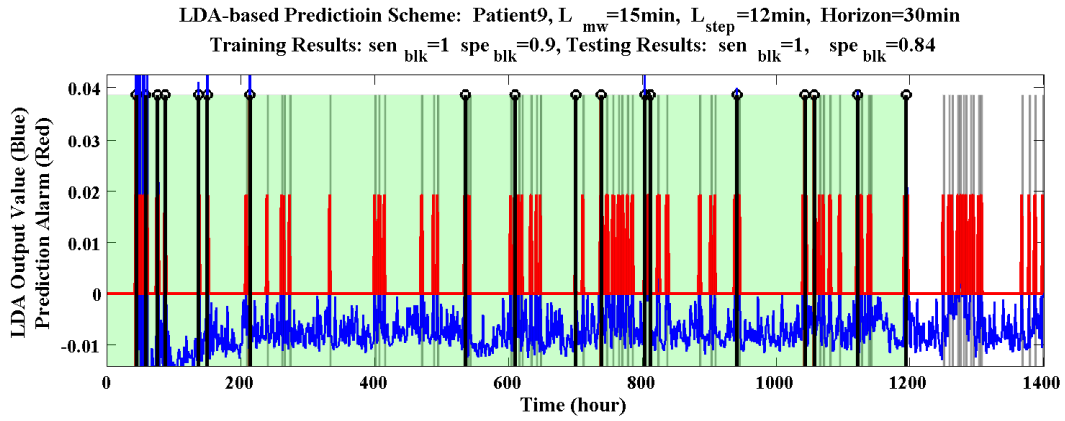


Figure 7.5: The prediction outcome of the LDA-based ALP prediction scheme for patient 9 using a prediction horizon of H=30 minutes with L_{mw} = 15 minutes and L_{step}= 12 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The blue line represents the prediction value of the LDA classifier. If the prediction value is higher than 0, a monitored pattern is classified as pre-seizure, a warning is triggered; otherwise, the pattern is classified as non-event. The LDA hyperplane is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

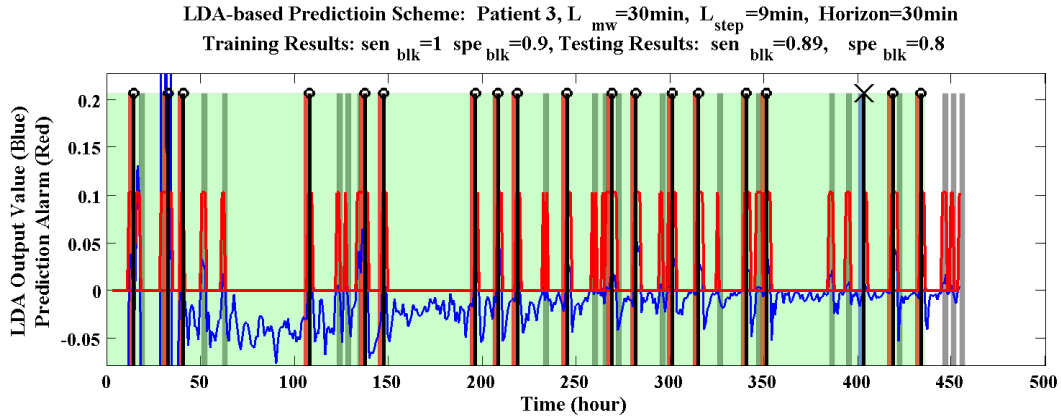


Figure 7.6: The prediction outcome of the LDA-based ALP prediction scheme for patient 3 using a prediction horizon of H=30 minutes with L_{mw} = 30 minutes and L_{step}= 9 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The piecewise horizontal line indicates the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

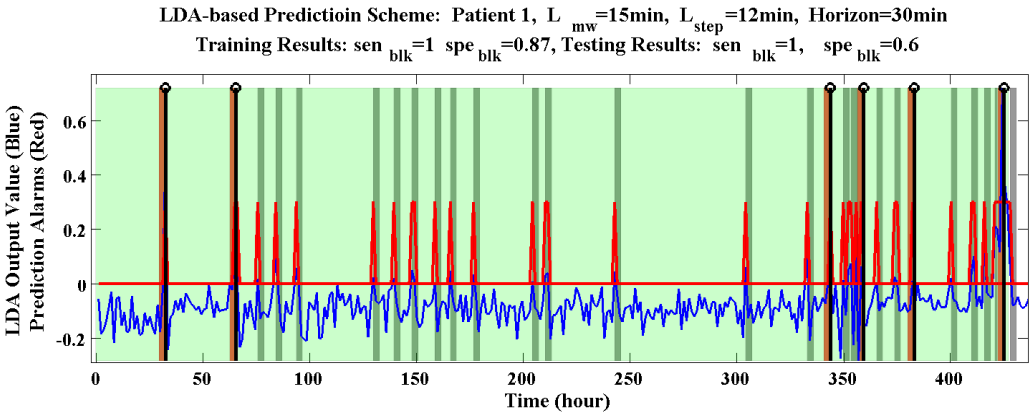


Figure 7.7: The prediction outcome of the LDA-based ALP prediction scheme for patient 1 using a prediction horizon of H=30 minutes with L_{mw} = 15 minutes and L_{step}= 12 minute. The vertical black lines indicate the timings of seizure onset. The piecewise horizontal line indicates the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

In particular, we pay a special interest on the prediction of event I. Because event I may be highly related to the 'anxiety' and 'uncertainty' of the brain activity. The prediction of such an event in driving EEG can be very insightful for online prediction of various mental states, such as 'uncertainty', 'certainty', 'alertness' and 'drowsiness'. For example, predicting a drowsy state can be applied to provide warning alarms to a driver to avoid fatigue-related driving accidents.

7.2.1 The Driving EEG Acquisition and Preprocessing

During the experiment, EEG data were collected with an EEG cap containing 40 Ag/AgCl electrodes according to the international 10-20 system. There are four electrodes that were used for measuring eye movements to remove muscular artifacts. The rest 36 electrodes were mounted on the scalp and thus used for analyses in this chapter. The placement of the 36 scalp electrodes is shown in Figure 3.1. The signals were amplified by NuAmps Express system (Neuroscan Inc, USA) and sampled at 1000Hz. 24 subjects were recruited from the student body of University at Buffalo. All subjects had driving experience without any motion disability. During the driving experiment

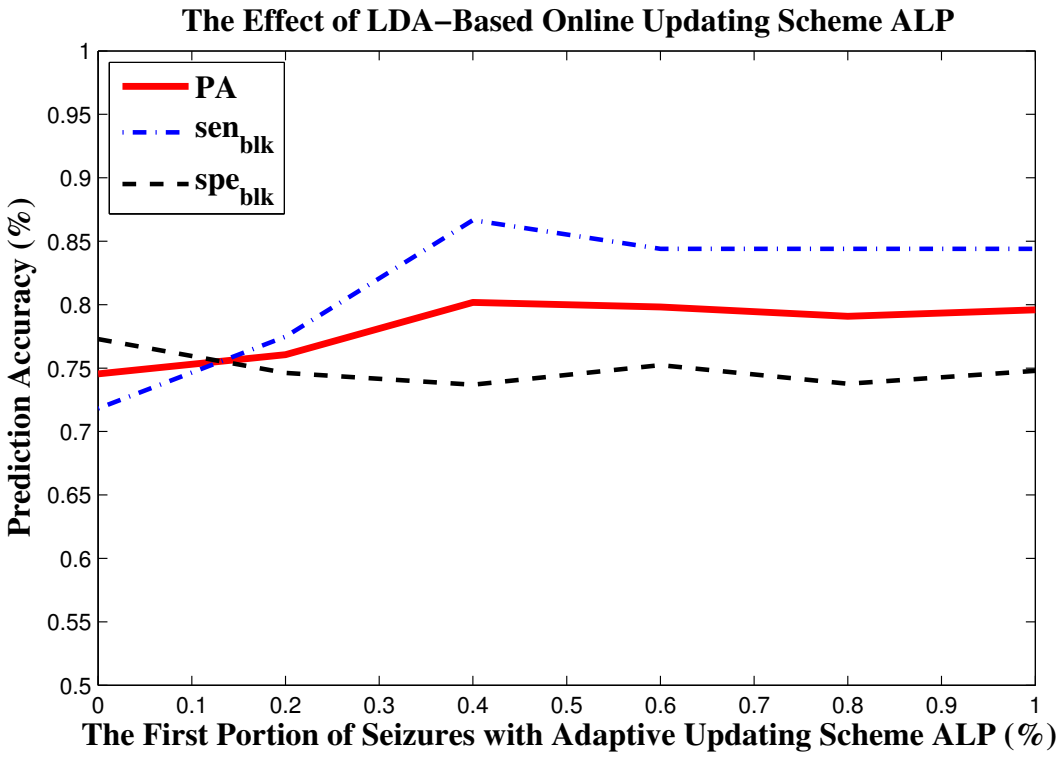


Figure 7.8: The effectiveness of the adaptive online updating scheme ALP. The ALP scheme was only performed on the EEG with the first portion of seizures, and the obtained LDA classification hyperplane was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ALP scheme was actively performed. The point 0 indicates that the initial classification hyperplane of LDA was unchanged throughout the prediction process; and the point 1 means that the LDA classification hyperplane was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more seizures were used to train the LDA classifier. The strong increase trend indicates that the adaptive updating scheme ALP is effective to increase online prediction performance over time.

Table 7.3: The training and testing performance characteristics of the ALP prediction framework for prediction horizon, respectively. The 'Non-Update' scheme employed the trained threshold of prediction score, and kept the threshold unchanged in the testing dataset. The prediction performance on the testing dataset is presented in the table. The prediction performances of two random prediction schemes (periodic and Poisson) are also reported. The prediction periods of the periodic and Poisson schemes for each patient are equal to the averaged length of inter-seizure intervals of the patient.

		Setting		Training		Testing		Non-Update		Possion		Periodic	
Horizion	Patient	L_{mw}(min)	L_{step} (min)	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}
30 min	1	15	12	1.00	0.87	1.00	0.60	0.67	0.74	0.00	0.53	0.00	0.53
	2	30	15	1.00	0.86	1.00	0.76	1.00	0.88	0.00	0.35	0.00	0.35
	3	30	9	1.00	0.90	0.89	1.00	0.67	0.93	0.17	0.52	0.17	0.52
	4	15	12	1.00	0.82	0.86	0.80	0.71	0.85	0.00	0.72	0.07	0.72
	5	60	1	1.00	0.84	0.83	0.66	0.83	0.73	0.00	0.39	0.00	0.39
	6	30	18	1.00	0.57	1.00	0.82	0.25	0.40	0.13	0.34	0.00	0.34
	7	15	6	1.00	0.73	0.86	0.61	0.71	0.68	0.05	0.52	0.00	0.52
	8	180	12	0.86	0.85	0.67	0.63	1.00	0.35	0.06	0.77	0.12	0.77
	9	15	12	1.00	0.90	1.00	0.84	0.67	0.97	0.05	0.95	0.05	0.95
	10	30	15	1.00	0.93	0.60	0.78	0.60	0.82	0.00	0.70	0.09	0.70
	Ave.			0.96	0.85	0.86	0.73	0.73	0.77	0.06	0.96	0.06	0.96
	PA			0.91		0.80		0.75		0.51		0.51	
90 min	1	90	18	1.00	0.50	1.00	0.54	0.50	0.43	0.00	0.47	0.33	0.47
	2	180	18	0.67	0.80	1.00	0.73	0.33	0.71	0.00	0.17	0.00	0.17
	3	30	15	1.00	0.74	0.89	1.00	0.89	0.37	0.30	0.15	0.39	0.15
	4	60	18	1.00	0.70	1.00	0.47	1.00	0.43	0.07	0.39	0.07	0.39
	5	90	18	0.86	0.71	0.83	0.58	1.00	0.23	0.00	0.15	0.00	0.15
	6	150	18	0.75	0.82	0.75	0.47	0.75	0.31	0.25	0.00	0.25	0.00
	7	15	18	0.88	0.55	1.00	0.44	1.00	0.21	0.21	0.22	0.21	0.22
	8	180	9	1.00	0.80	0.83	0.51	1.00	0.25	0.18	0.57	0.18	0.57
	9	120	6	1.00	0.69	0.89	0.50	1.00	0.22	0.11	0.79	0.16	0.79
	10	60	12	1.00	0.57	1.00	0.41	1.00	0.29	0.18	0.24	0.18	0.24
	Ave.			0.96	0.67	0.90	0.51	0.92	0.31	0.15	0.88	0.19	0.88
	PA			0.82		0.71		0.62		0.52		0.54	
150 min	1	90	15	1.00	0.27	1.00	0.33	1.00	0.28	0.33	0.56	0.33	0.56
	2	150	18	0.67	0.92	1.00	0.59	0.33	0.50	0.00	0.10	0.00	0.10
	3	120	12	1.00	0.73	0.89	1.00	0.67	0.91	0.70	0.25	0.65	0.25
	4	120	15	1.00	0.50	1.00	0.19	0.86	0.46	0.13	0.25	0.13	0.25
	5	30	18	1.00	0.48	1.00	0.40	1.00	0.10	0.13	0.09	0.00	0.09
	6	150	18	0.75	0.75	1.00	0.41	1.00	0.22	0.13	0.17	0.13	0.17
	7	15	18	1.00	0.51	1.00	0.26	1.00	0.17	0.37	0.19	0.37	0.19
	8	180	9	0.86	0.73	0.83	0.32	0.67	0.73	0.24	0.47	0.24	0.47
	9	90	18	1.00	0.62	0.89	0.38	1.00	0.17	0.32	0.49	0.26	0.49
	10	180	18	1.00	0.32	0.80	0.41	1.00	0.67	0.09	0.25	0.27	0.25
	Ave.			0.95	0.51	0.93	0.38	0.85	0.40	0.29	0.82	0.28	0.82
	PA			0.73		0.66		0.63		0.56		0.55	

of each subject, the timing of each map-looking activity was recorded.

7.2.2 Target Event Definition

As discussed in the fundamental structure of the prediction framework, it is always useful to perform a preliminary study on the analyzed time series data, and try to find some knowledge about the data prior to event occurrence. Figure 7.2.2 plots the The statistics of the inter-arrival periods of event I (periods of continuous driving without

looking at the map), and the intervals between event I and event II (periods of map-looking). We notice that for those event I with short inter-arrival times (less than 2 seconds) may be clustered as one event. There are often the cases such that a subject drives for a period without looking at the map; however, if the subject feel uncertain about the next driving route, she/he may look at the map back and forth a number of times in a short periods. As a result, these groups of event Is are natural to be considered as a part of learning process, and are less relevant to our interested target event, which is related to 'uncertainty' of future driving directions. In particular, we are more interested in an 'initial' event I after a relative long-term continuous driving without looking at the map. In this study, we consider five second is a reasonable time interval to separate two 'initial' event Is. Thus we selected the event I with at least five seconds preceding continuous driving as the target event. The event Is that have very short inter-arrival times are ignored in the prediction problem.

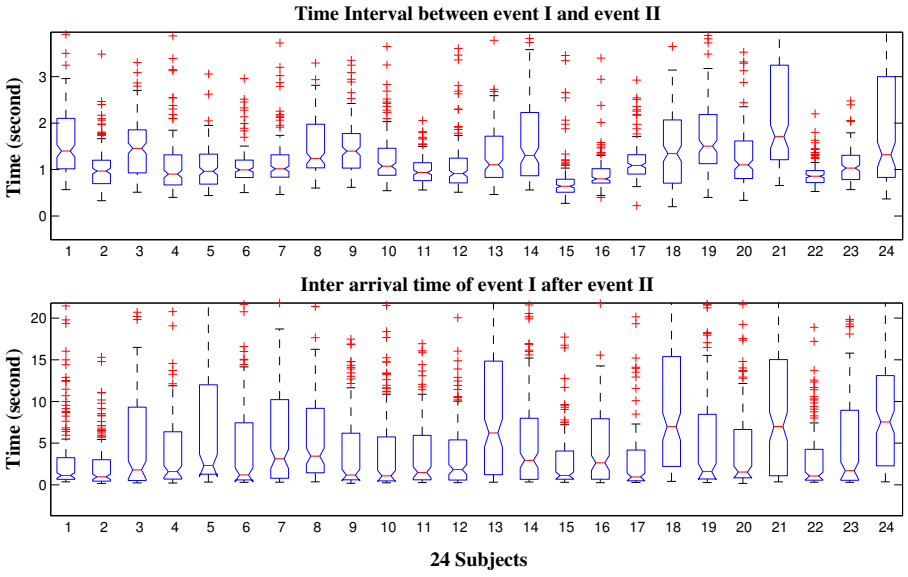


Figure 7.9: The statistics of the inter-arrival intervals of event I (periods of continuous driving without looking at the map), and the intervals between event I and event II (periods of map-looking).

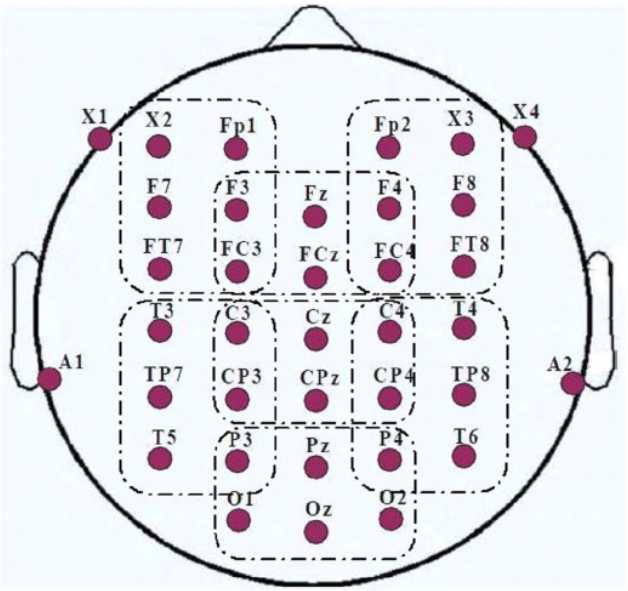


Figure 7.10: The 36 EEG channels are divided into seven channel groups according to their spacial locations. In the feature extraction stage, features are first extracted from each single channel, and then averaged over each channel group.

7.2.3 Data Processing and Feature Extraction

We employed a band-pass filter to decompose the EEG data into four frequency bands, they are 8 to 13 Hz, 13 to 30 Hz, 2 to 50 Hz, and 1 to 100 Hz, respectively. For EEG signals in each frequency band, we performed several univariate, bivariate analysis and time-frequency analysis of the driving EEG data. We also divided the 36 EEG channels into 7 groups according to their spacial locations. We consider the channel groups in the feature extraction procedure. In particular, we extracted the following first-level features from the raw EEG data.

- 9 univariate features: mean, variance, skewness, kurtosis, signal power, curve length, number of peaks, average nonlinear energy, variance to range ratio. Each univariate feature is calculated from the 36 channels, and then the features in the same channel group are averaged to represent the value of that channel group. In other word, each epoch of 36 channels is converted into 7 values of a univariate feature, each value represents a channel group.

- Three bivariate features: pairwise Euclidean distance, pairwise T-statistics, pairwise Pearson correlation. The bivariate features were first calculated in each channel group, and then averaged over all the pairs in each channel group. Each epoch of 36 channels is transformed into 7 values for each bivariate feature.

- One time-frequency feature: wavelet entropy. Wavelet analysis is first used to decompose each channel of EEG data into subbands, and then entropy is computed using the calculated wavelet coefficients. In the last, we average the wavelet entropy values within each channel group. Each time epoch of 36 channels is transformed into 7 values of wavelet entropy for each channel group.

7.2.4 Feature Selection

For a time epoch, its total number of extracted first-level features is $9 \times 7 + 3 \times 7 + 1 \times 7 = 91$. As discussed in the previous chapter, we employed a second sliding window to monitor the feature time series over time. Four temporal features are extracted from each first-level feature time series. Then the total number of temporal features is $91 \times 4 = 364$ features. We employed the Pudil's floating search to select which temporal features have strong discrimination power to separate the pre-event and non-event epochs. In this study, we selected the best 8 temporal features from the 364 candidates. Each EEG epoch of 36 channels is represented by a feature vector of the 8 selected temporal features. In the online monitoring process, each EEG epoch in the sliding window is converted into a 8-dimension feature vector and store in a constructed pattern library.

7.2.5 Computational Settings

The proposed prediction framework has been implemented on the EEG recordings of 24 subjects using four choices of prediction horizons, five choices of window length, and three choices of step length. The complete parameter settings of the prediction system are summarized in Table 7.4.

Table 7.4: Computational settings of the prediction framework for mental-state prediction in a simulated driving environment.

Parameter Setting	Setting Choices
Prediction Horizon	400, 600, 800, 1000 ms
1st-level sliding window (monitor raw time series)	window size: 1 s moving step length: 100 ms
2nd-level sliding window (monitor feature time series)	window size: 1, 2, 3, 4, 5 second moving step length: 100, 200, 300 ms
Online Prediction Scheme	1. Adaptive Probabilistic Prediction Scheme 2. Adaptive LDA-based Prediction Scheme
Feature Selection Method	Pudil's floating search based on 1-Nearest Neighbour leave-one-out classification performance.
1st-level features	Nine univariate features: mean, variance, skewness kurtosis, signal power, curve length, number of peaks average nonlinear energy, variance to range ratio. Three pairwise bivariate measures: Euclidean distance T-statistics, Pearson correlation. One time-frequency measure: wavelet entropy (features are averaged over each channel group as shown in Figure 7.2.2)
2nd-level features (temporal pattern feature)	1. accumulated vertical increase 2. accumulated vertical decrease 3. percentage of decline periods 4. amplitude range

7.2.6 Experimental Results

The averaged training and testing results over the 24 subjects for each prediction horizon and frequency band are summarized in Table 7.5. The best testing performance of the ATP prediction approach was achieved at a sen_{blk} of 0.83 and a spe_{blk} of 0.80 using the prediction horizon of 400ms in frequency band 2-50 Hz. Correspondingly, table 7.6 gives the detailed prediction performances of the 24 subjects using the prediction horizon of 400ms in frequency band 2-50 Hz. Figure 7.11, 7.12, and 7.13 show three best prediction examples of the ATP prediction scheme for subject 2, 5, and 24, respectively, using their best training parameter settings.

The best testing prediction performance of the ALP prediction scheme was achieved at a sen_{blk} of 0.84 and a spe_{blk} of 0.60 using the prediction horizon of 400ms in frequency band 8-13Hz. Correspondingly, table 7.7 gives the detailed prediction performances of the 24 subjects using the prediction horizon of 400ms in frequency band 8-13 Hz.

Figure 7.11, Figure 7.12, and Figure 7.13 show three best prediction examples of the ALP prediction scheme for subject 2, 5, and 24, respectively, using their best training parameter settings. Figure 7.16, 7.17, and 7.18 show three best prediction examples of the ALP prediction scheme for subject 1, 5, and 9, respectively.

From the figure demonstrations, one can observe that the proposed ATP prediction scheme achieved very attractive prediction performance. Averaged over 24 subjects, 83% target events were correctly predicted while 80% of the time is not in false-prediction periods. The LDA-based ALP prediction scheme had a similar sensitivity of 84%. However, it was achieved at a specificity of 60%. That is 40% of the time is under false-prediction periods. From the demonstrations Figures 7.16, 7.17, and 7.18, one can observe that the prediction of ALP scheme is very sensitive around the LDA decision boundary (the horizontal line at zero). In many false prediction cases, the prediction values of the LDA classifier is only slightly higher than zero, and classified as pre-event cases. To make the prediction more robust againt boundary noises, one can shift the decision bounary a little higher to eliminate much of the false predictions. There are also many other options of other classification technqieus to tackle this problem, which is out of the scope of this study. We employ the default decision boundary of LDA throughout this study.

In this application example, it clearly shows that the adaptive-threshold-based ATP prediction scheme is less sensitive to pattern noises. This is because a prediction is only triggered if the monitored pattern is already identified as a pre-event pattern in the pattern library with a prediction score of higher than the score-threshold. All patterns other than the identified pre-event patterns cannot trigger any warning alarms. This makes the proposed probabilistic ATP prediction very attractive in real-life applications.

7.3 Conclusion

In this chapter, we implemented the proposed online monitoring and prediction framework to solve two challenging real-world problems based on EEG time series data. The proposed adaptive prediction schemes ATP and ALP were evaluated. In the seizure

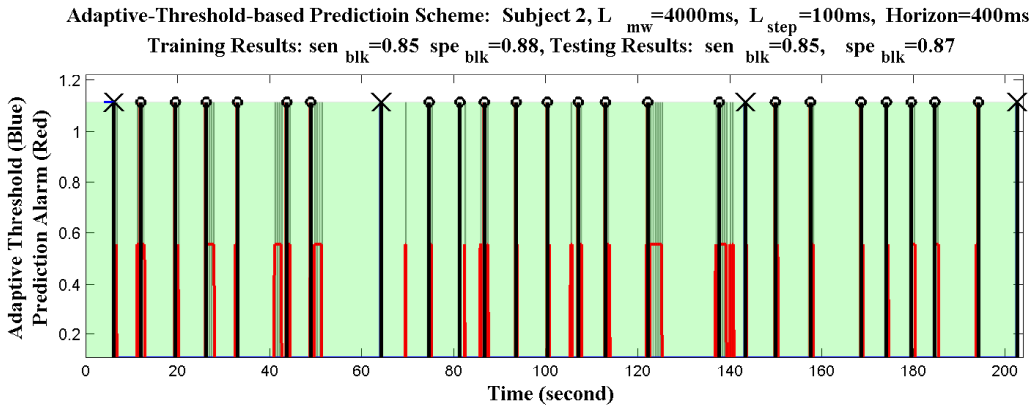


Figure 7.11: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 2 using the prediction horizon of H= 400 ms with L_{mw} = 4000 ms and L_{step}= 100 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

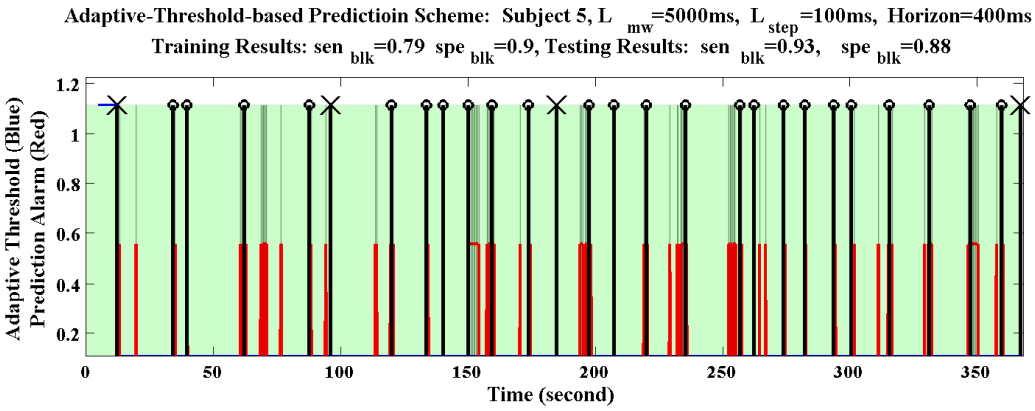


Figure 7.12: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 5 using the prediction horizon of H=400 ms with L_{mw} = 5000 ms and L_{step}= 100 ms. The vertical black lines indicate the onset starting times of the occurred seizures. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

Table 7.5: The averaged training and testing results over the 24 subjects for each prediction horizon and frequency band. The best testing prediction performance of ATP approach was achieved at sen_{blk}=0.83 and spe_{blk} = 0.80 using the prediction horizon of 400ms and the frequency band 2-50 Hz. The best testing prediction performance of the LDA-based prediction scheme was achieved at sen_{blk}=0.843 and spe_{blk} = 0.60 using the prediction horizon of 400ms and frequency band 8-13Hz.

		ATH				LDA			
Frequency	Horizon	Training		Testing		Training		Testing	
Band (Hz)	(100 ms)	Sen_{blk}	Spe_{blk}	Sen_{blk}	Spe_{blk}	Sen_{blk}	Spe_{blk}	Sen_{blk}	Spe_{blk}
8-13 Hz	4	0.71	0.77	**0.77**	**0.63**	0.89	0.65	**0.81**	**0.62**
	6	0.72	0.72	0.76	0.56	0.84	0.63	0.80	0.58
	8	0.85	0.57	0.81	0.51	0.85	0.57	0.81	0.51
	10	0.81	0.55	0.81	0.51	0.81	0.55	0.81	0.51
13-30 Hz	4	0.67	0.75	0.71	0.68	0.9	0.64	0.8	0.56
	6	0.71	0.71	0.77	0.62	0.86	0.63	0.75	0.52
	8	0.73	0.68	0.83	0.55	0.87	0.59	0.8	0.49
	10	0.73	0.66	0.82	0.55	0.89	0.57	0.8	0.47
2-50 Hz	4	0.82	0.90	**0.79**	**0.83**	0.91	0.66	**0.81**	**0.61**
	6	0.87	0.82	0.83	0.75	0.84	0.63	0.80	0.58
	8	0.88	0.8	0.83	0.71	0.84	0.57	0.81	0.51
	10	0.88	0.78	0.85	0.67	0.81	0.55	0.81	0.51
1-100 Hz	4	0.34	0.89	0.39	0.8	0.92	0.65	0.8	0.59
	6	0.48	0.77	0.52	0.68	0.92	0.61	0.81	0.53
	8	0.56	0.72	0.63	0.61	0.88	0.58	0.81	0.48
	10	0.6	0.67	0.69	0.55	0.83	0.59	0.74	0.5

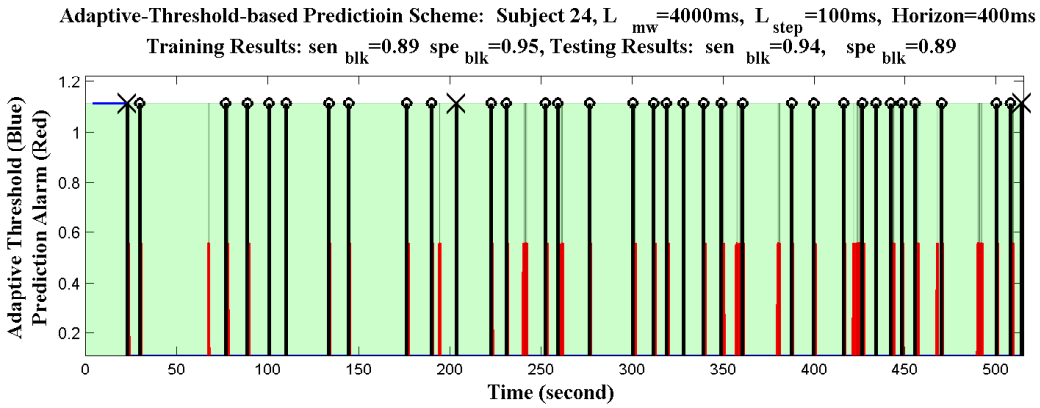


Figure 7.13: The prediction outcome of the adaptive-threshold-based ATP prediction scheme for patient 24 using the prediction horizon of H=400 ms with L_{mw} = 4000 ms and L_{step}= 100 ms. The vertical black lines indicate the onset starting times of the occurred seizures. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

prediction problem. The ALP prediction scheme achieved a little better prediction performance than the ATP prediction scheme. The overall testing prediction accuracies for the ALP scheme and ATP scheme are 80% and 77%, respectively, averaged over 10

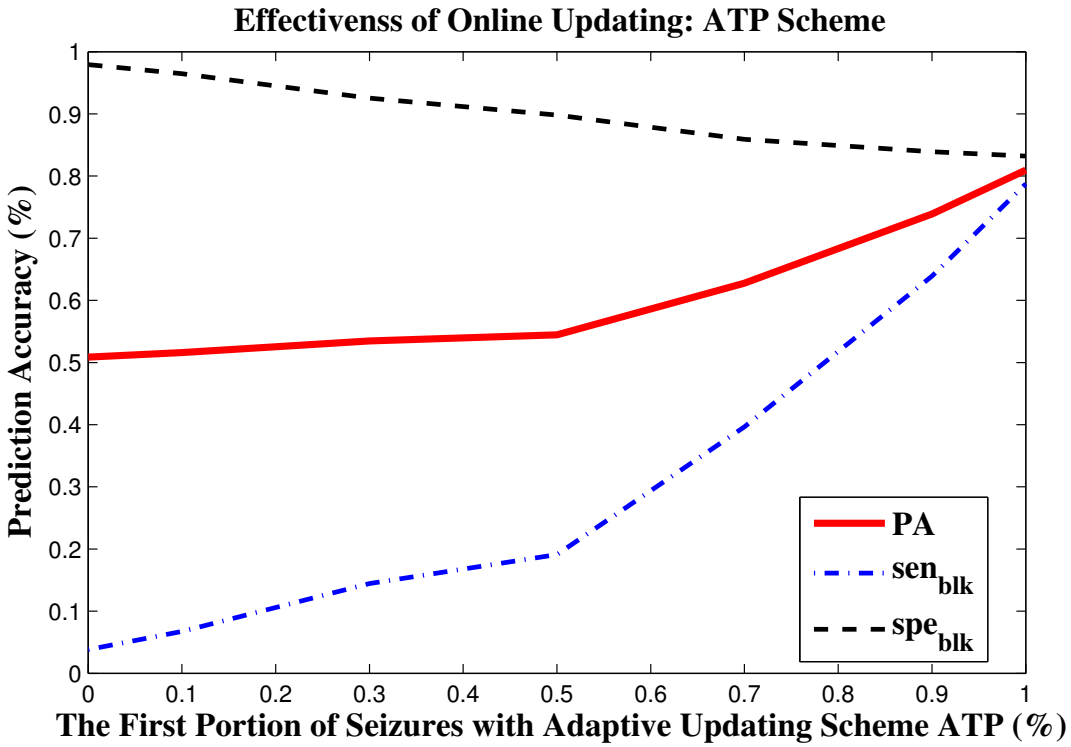


Figure 7.14: The effectiveness of the adaptive online updating scheme ATP using the EEG frequency band 2-50 Hz. The ATP scheme was performed on the EEG with the first portion of total events, and the obtained score threshold was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ATP scheme was actively performed. The point 0 indicates that the initial score threshold was unchanged throughout the prediction process; and the point 1 means that the threshold was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more events were used to train the ATP prediction scheme. The strong increase trend of prediction accuracy indicates that the adaptive updating scheme ATP is effective to increase online prediction performance over time.

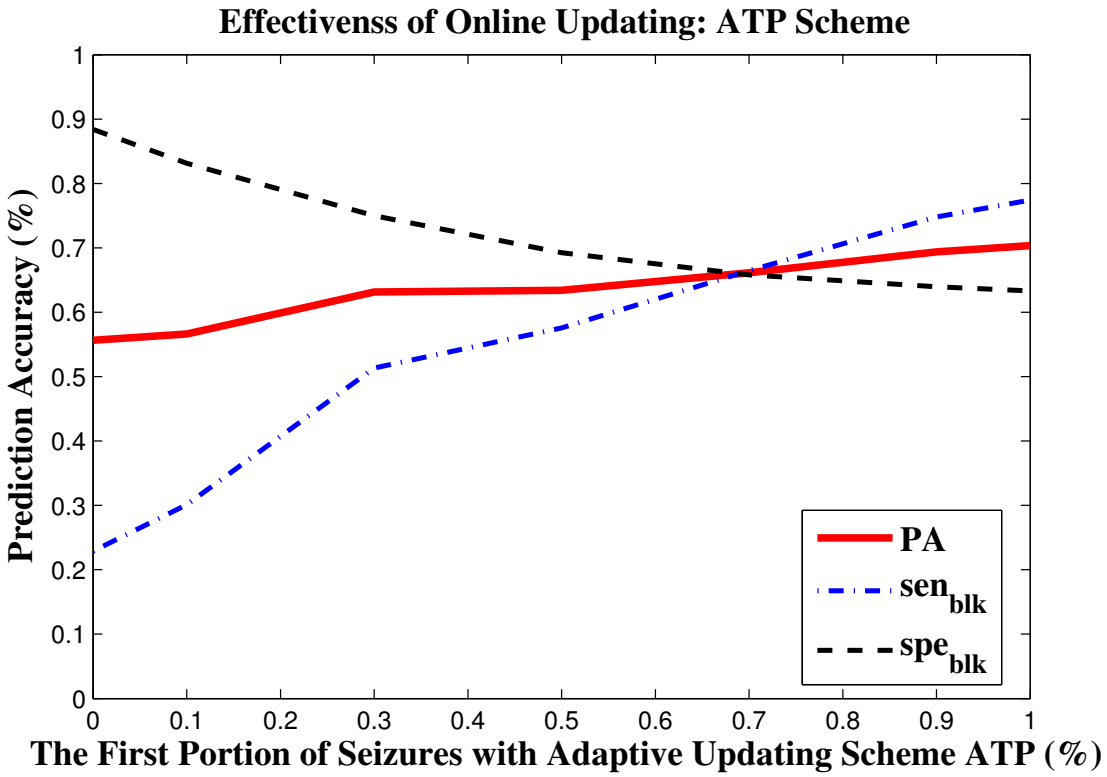


Figure 7.15: The effectiveness of the adaptive online updating scheme ATP using the EEG frequency band 8-13 Hz. The ATP scheme was performed on the EEG with the first portion of total events, and the obtained score threshold was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ATP scheme was actively performed. The point 0 indicates that the initial score threshold was unchanged throughout the prediction process; and the point 1 means that the threshold was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more events were used to train the ATP prediction scheme. The strong increase trend of prediction accuracy indicates that the adaptive updating scheme ATP is effective to increase online prediction performance over time.

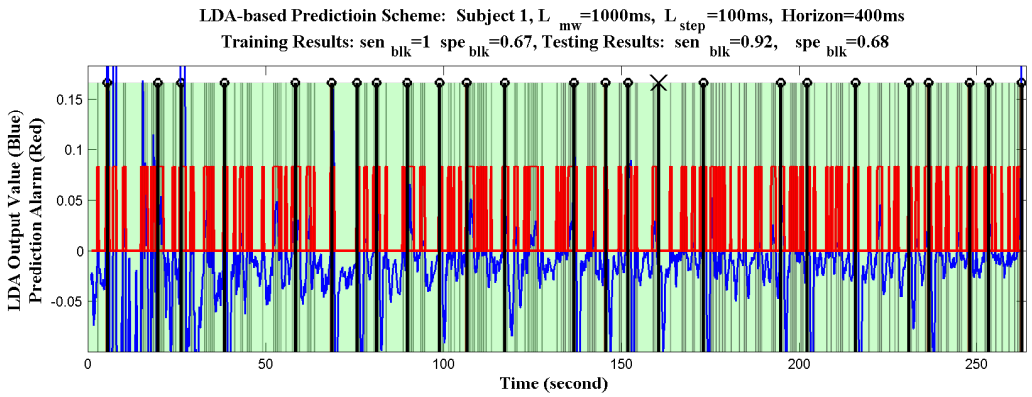


Figure 7.16: The prediction outcome of the LDA-based ALP prediction scheme for patient 1 using the prediction horizon of H= 400 ms with L_{mw} = 1000 ms and L_{step}= 100 ms. The vertical black lines indicate the onset starting times of the occurred seizures. The piecewise horizontal line indicates the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

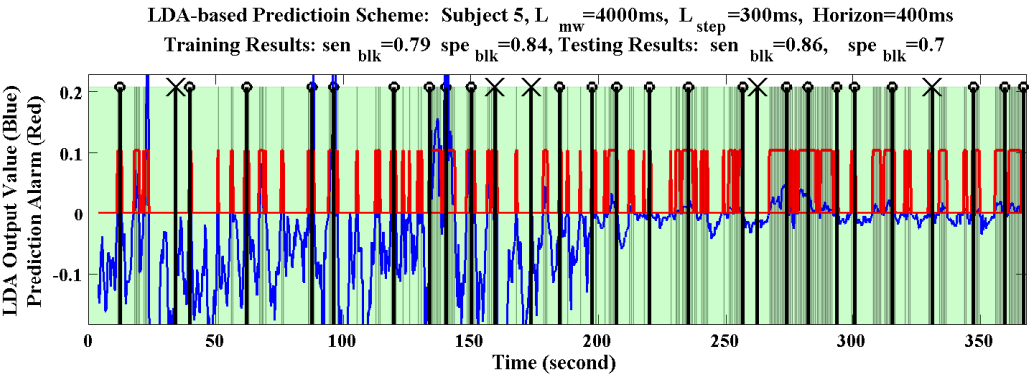


Figure 7.17: The prediction outcome of the LDA-based LDA-based prediction scheme for patient 5 using the prediction horizon of H=400 ms with L_{mw} = 4000 ms and L_{step}= 300 minute. The vertical black lines indicate the onset starting times of the occurred seizures. The blue line represents the prediction value of the LDA classifier. If the prediction value is higher than 0, a monitored pattern is classified as pre-seizure, a warning is triggered; otherwise, the pattern is classified as non-event. The LDA hyperplane is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

Table 7.6: The training and testing results of the adaptive-threshold-based ATP prediction scheme for the 24 subjects using the prediction horizon of 400 ms and the frequency band of 2-50 Hz.

	Settings			Training		Testing		Non-Update	
Sub	L_{mw} (100ms)	$L+step$ (100ms)	Horizon (100ms)	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}
1	50	1	4	0.77	0.95	0.75	0.88	0.00	0.97
2	40	1	4	0.85	0.91	0.69	0.89	0.08	0.98
3	50	1	4	0.86	0.88	0.86	0.75	0.00	0.99
4	50	1	4	0.88	0.86	0.81	0.69	0.00	1.00
5	50	1	4	0.71	0.91	0.79	0.89	0.00	0.99
6	30	1	4	0.80	0.94	0.79	0.82	0.00	1.00
7	40	1	4	0.79	0.93	0.72	0.83	0.06	0.98
8	50	1	4	0.81	0.94	0.87	0.87	0.07	0.97
9	30	1	4	0.75	0.97	0.80	0.94	0.00	0.99
10	30	1	4	0.76	0.92	0.94	0.84	0.00	1.00
11	30	1	4	0.93	0.85	0.85	0.73	0.00	0.97
12	30	1	4	0.84	0.87	0.74	0.78	0.11	0.97
13	30	1	4	0.87	0.88	0.86	0.81	0.07	0.98
14	40	1	4	0.76	0.92	0.85	0.80	0.00	0.99
15	40	1	4	0.75	0.90	0.82	0.85	0.00	0.99
16	30	1	4	0.89	0.82	0.65	0.77	0.00	1.00
17	40	1	4	0.88	0.96	0.71	0.95	0.00	1.00
18	40	1	4	0.71	0.86	0.93	0.85	0.00	0.97
19	30	1	4	0.79	0.91	0.56	0.90	0.00	0.99
20	40	1	4	0.94	0.78	0.89	0.76	0.06	0.98
21	50	2	4	0.82	0.74	0.50	0.87	0.27	0.86
22	40	1	4	0.85	0.98	0.83	0.90	0.00	1.00
23	50	1	4	0.79	0.87	0.77	0.73	0.15	0.94
24	40	1	4	0.89	0.96	0.94	0.90	0.06	1.00
Ave.				**0.82**	**0.90**	**0.79**	**0.83**	**0.04**	**0.98**
PA				**0.86**		**0.81**		**0.52**	

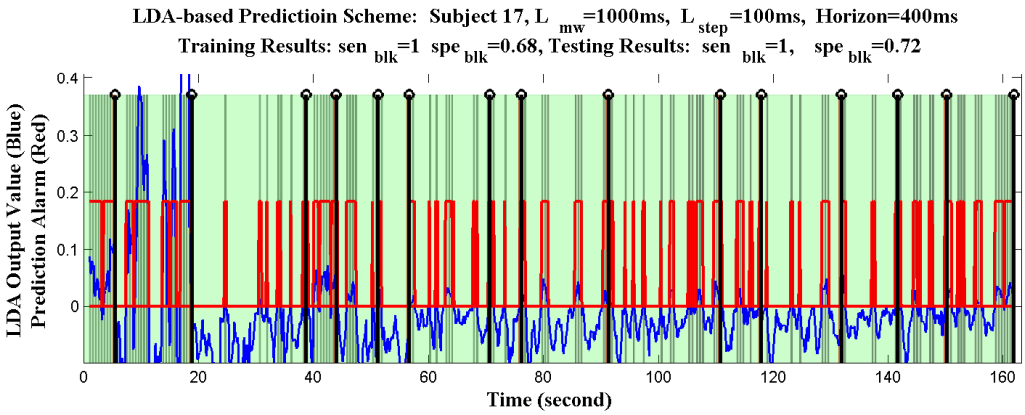


Figure 7.18: The prediction outcome of the LDA-based ALP prediction scheme for patient 17 using the prediction horizon of H= 400 ms with L_{mw} = 1000 ms and L_{step}= 100 ms. The vertical black lines indicate the timings of seizure onset. The piecewise horizontal line indicates the adaptive threshold, which is updated after each seizure onset. The red line represents the prediction alarms. The non-zero values in the red line indicate prediction alarms.

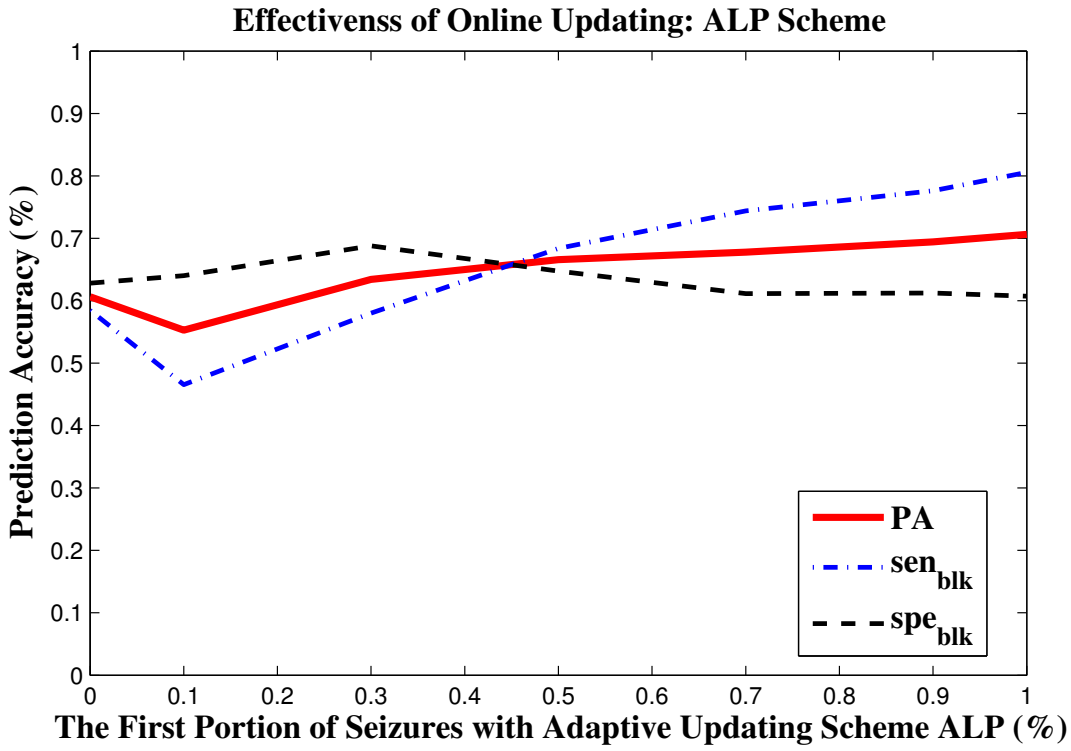


Figure 7.19: The effectiveness of the adaptive online updating scheme ALP using the EEG frequency band 2-50 Hz. The ALP scheme was performed on the EEG with the first portion of total events, and the obtained LDA classification hyperplane was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ALP scheme was actively performed. The point 0 indicates that the initial LDA classification hyperplane was unchanged throughout the prediction process; and the point 1 means that the threshold was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more events were used to train the ATP prediction scheme. The strong increase trend of prediction accuracy indicates that the adaptive updating scheme ALP is effective to increase online prediction performance over time.

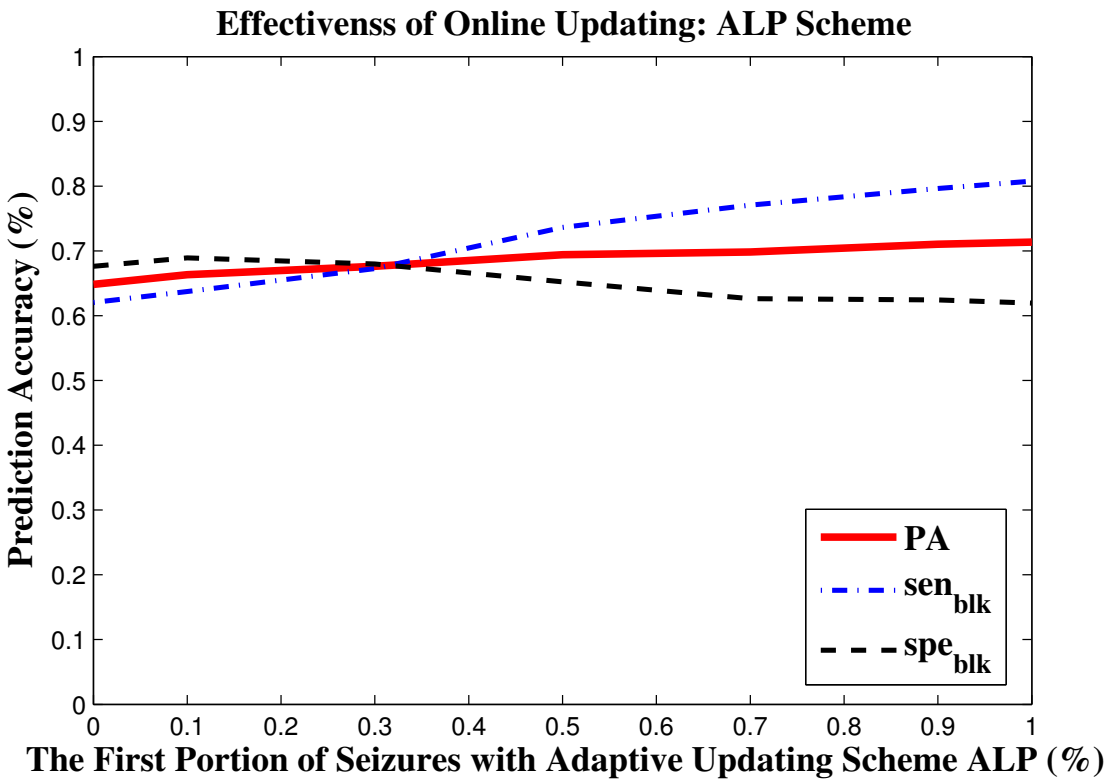


Figure 7.20: The effectiveness of the adaptive online updating scheme ALP using the EEG frequency band 8-13 Hz. The ALP scheme was performed on the EEG with the first portion of total events, and the obtained LDA classification hyperplane was kept unchanged in the remaining EEG recordings. The horizontal axis indicates the portion of seizures the ALP scheme was actively performed. The point 0 indicates that the initial LDA classification hyperplane was unchanged throughout the prediction process; and the point 1 means that the threshold was updated after each seizure onset. It shows clearly that the overall prediction accuracies increased as more events were used to train the ATP prediction scheme. The strong increase trend of prediction accuracy indicates that the adaptive updating scheme ALP is effective to increase online prediction performance over time.

Table 7.7: The training and testing results of the LDA-based prediction scheme for the 24 subjects using the prediction horizon of 400 ms and the frequency band of 8-13 Hz.

	Settings			Training		Testing		Non-Update	
Sub	L_{mw} (100ms)	$L + step$ (100ms)	Horizon (100ms)	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}	sen_{blk}	spe_{blk}
1	10	1	4	1.00	0.69	0.92	0.70	0.92	0.66
2	20	1	4	0.85	0.73	0.69	0.63	0.46	0.78
3	40	1	4	0.86	0.65	0.64	0.68	0.64	0.59
4	20	1	4	0.88	0.73	0.81	0.61	0.94	0.57
5	40	1	4	0.93	0.72	1.00	0.67	0.43	0.85
6	20	1	4	0.93	0.65	0.86	0.60	0.50	0.58
7	10	1	4	1.00	0.65	0.78	0.69	0.78	0.72
8	50	1	4	0.88	0.59	0.80	0.48	0.87	0.53
9	10	1	4	1.00	0.58	0.80	0.56	0.60	0.61
10	40	1	4	0.82	0.58	0.88	0.63	0.88	0.38
11	40	1	4	0.71	0.60	0.54	0.58	0.54	0.68
12	20	1	4	0.95	0.64	0.84	0.63	0.84	0.55
13	10	1	4	0.73	0.61	0.64	0.61	0.50	0.60
14	10	1	4	0.95	0.59	0.95	0.57	0.75	0.65
15	40	1	4	0.67	0.67	0.55	0.73	0.27	0.90
16	10	1	4	0.83	0.68	0.94	0.62	0.41	0.69
17	30	1	4	1.00	0.66	0.86	0.65	0.71	0.56
18	20	1	4	0.86	0.75	0.64	0.70	0.29	0.78
19	30	1	4	0.89	0.62	1.00	0.67	0.33	0.87
20	40	1	4	0.94	0.63	0.89	0.54	0.78	0.76
21	10	1	4	0.91	0.61	0.86	0.54	0.64	0.78
22	10	1	4	0.92	0.56	0.75	0.53	0.50	0.83
23	10	1	4	0.93	0.69	0.92	0.70	0.62	0.69
24	30	1	4	0.83	0.60	0.82	0.55	0.71	0.62
Ave.				0.89	0.65	0.81	0.62	0.62	0.68
PA				0.77		0.72		0.65	

patients. In the driving EEG prediction problem, the ATP scheme generated a considerable better prediction performance than the ALP scheme. Using the best training settings, the overall testing prediction accuracies for the ATP scheme and ALP scheme are 82% and 72%, respectively, averaged over 24 subjects.

The big advantage of ALP scheme is that it employs statistics and optimization theory to obtain a decision boundary to classify pre-event and non-event patterns. The main drawback of this scheme is that the constructed pattern library is incrementally increasing. The computational load to train the LDA classifier online is thus incrementally increasing over time. The other drawback of ALP scheme is that its prediction performance may be seriously deteriorated by monitoring noises and outliers. The online noises around the decision boundary of LDA classifier may lead to many false predictions. While outliers of monitored patterns may deteriorate the quality of the trained LDA decision boundary.

The probabilistic ATP scheme constructs a pattern library based on pattern clusters

in discrete feature space. Thus a resulting advantage of the ATP scheme is that the size of the pattern library is limited. Since the total number of pattern clusters is a known number. In real-life applications, the pattern-cluster library can only take a very small space and is very computational efficient. For example in the seizure prediction case, the total number of stored pattern clusters is around a level of one thousand, and the number of identified pre-seizure pattern clusters is around a level of one hundred. Another significant advantage of the ATP scheme is that it is not sensitive to pattern noises and outliers in the online monitoring process. A prediction is only triggered if the monitored pattern cluster is an already identified as a pre-event pattern cluster in the pattern library. All other monitored patterns (including any pattern noises and outliers) cannot trigger any warning alarms. This makes the proposed probabilistic ATP prediction very attractive in real-life applications. However, one drawback of the ATP scheme is that much effort have to be taken on the discretization of each selected feature space. An appropriate discretization of the feature space is crucial in the APT prediction scheme.

In general, the proposed adaptive online prediction framework with two prediction-rule schemes generated very attractive prediction performances on the two challenging online prediction problems. The general structure of the online monitoring and prediction framework make it convenient to be applied to a wide range of prediction problems of complex time series events using a prediction horizon. The proposed framework is a fundamental contribution to the field of time series data mining. Especially, the proposed framework provide a useful analytical tool for multichannel nonstationary time series data.

Chapter 8

Conclusions and Future Research

8.1 Conclusions

This dissertation research made an extensive study on time series prediction problems. The time series feature extraction techniques and the start-of-the-art prediction models were reviewed. The serious drawbacks of the existing methods for complex non-stationary time series motivate the directions of this research. This research presents two novel adaptive online monitoring and prediction frameworks as well as a robust algorithm for time series feature extraction. The proposed approaches have made original and fundamental contributions to the fields of online monitoring and prediction of massive non-stationary noisy time series data.

Chapter 4 presents an adaptive prediction framework which was built based on the concept of reinforcement learning. The proposed framework is a baseline sample-based approach, which compares the query time series patterns with the patterns from a baseline with known class information (normal or abnormal). The search of the best matching patterns within the whole database can be achieved by employing a KNN method. The two baselines were updated online with a gradient-based reinforcement learning algorithm according to prediction feedbacks. If a prediction is wrong, it punishes the 'bad' baseline samples according to their contributions to this false prediction. If a prediction is correct, then the 'good' baseline samples are enhances according to their contributions to the right prediction. By doing so, the proposed framework is supposed to collect the more and more predictive baseline patterns over time. Using EEG recordings from five patients with epilepsy, we have demonstrated that the adaptive learning framework considerably improved the prediction performance of the system

based on the time block-based sensitivity/specificity and ROC analysis. However, like many other reinforcement learning problems, the proposed reinforcement learning system may require a large number of seizures to construct the most representative and informative baselines for each individual patient. However, the current available seizures for each patient were too few (7 to 23) to train the reinforcement learning system. We anticipate that the performance of the proposed reinforcement learning prediction system could be further improved with more EEG data and seizure onsets available for each patient.

Chapter 5 proposes a new online time series segmentation algorithm TSTD and SWTD. The current segmentation approaches highly rely on some data-specific decomposition strategies, which lead to a tedious parameter tuning procedure in practice. Another bottleneck problem of online segmentation algorithms is the high computational complexity. To tackle these problems, we present an online time series segmentation approach that is accurate, fast, and easily applicable to various time series with different scales. In particular, the proposed online segmentation framework has three important features. Firstly, it employs a data-independent decomposition strategy, which employs a scaled universal statistical threshold measure to control approximation accuracy directly regardless of data values. Secondly, it employs a novel two-stage top-down segmentation algorithm, which is capable of achieving a guaranteed approximation accuracy for various time series without a tedious threshold turning process. At last, it employs a closed-form online updating formulas and achieves a very low computing cost to process massive time series streams online. The complexity of processing a new incoming data point is only $O(1)$. It is very easy to setup the parameters of SWTD compared with many others that employ data-dependent threshold strategies. We employed only one parameter setting ($R_{kp}^{2*} = 0.95$ and $R_{on}^{2*} = -1$) in the numerical experiments of 24 real-world time series. The experimental results showed that the proposed online SWTD works very fast online while achieving a high approximation accuracy for all data sets with only one parameter setting. The proposed online segmentation approach SWTD has a great potential to work well for online monitoring and processing of highly nonstationary time series without a tedious parameter tuning

process. Based on this algorithm, one can represent massive time series by their key skeleton points, which are efficient to deal with in a very low dimensionality.

Chapter 6 develops a general online monitoring and prediction framework for time series event. The proposed prediction framework employs a feature selection technique to select the event-related first-level characteristic features from raw time series. A two-sliding-window approach is proposed for online monitoring time series and temporal feature extraction. The first-level sliding window extracts the selected first-level characteristic features from raw time series; and the second-level sliding window extracts the temporal patterns of the first-level features. A pattern library is constructed to store the window-monitored time series patterns and some statistics of their occurrence history related to a target event (such as occurrence frequency in pre-event and non-event period, occurrence spectrum in different pre-event periods). Given a pattern library, we propose two different prediction schemes to construct online prediction rules. The first one is a probabilistic adaptive-threshold prediction (ATP) scheme which employs the concept of pattern cluster. A pattern-cluster library is constructed in discrete feature space. A probabilistic formula is proposed to estimate the pre-event likelihood of each stored pattern-cluster based statistics of its occurrence history. An optimized score threshold that maximizes the prediction performance over the monitoring history is identified to discriminate pre-event and non-event pattern clusters. The threshold is re-optimized after each occurrence of a target event. A big advantage of this approach is that the size of pattern-library is limited by the maximum number of pattern cluster. The drawback is that some efforts are needed to find the most appropriate discretization criterion in each feature space. The second proposed prediction approach is an adaptive LDA-based prediction (ALP) scheme, which performs online pattern-discovery and prediction in continuous feature space. In the ALP scheme, the pattern library only stores the monitored feature vectors and their class labels (pre-event or non-event). Then the pattern-discovery problem can be formulated as a typical binary classification problem. The popular binary classification technique LDA is employed to to construct an optimal hyperplane to classify the feature vectors of the two classes. The LDA hyperplane is retrained after each occurrence of a target event. The most

advantage of this scheme is that optimization and statistics theory can be employed to find the optimized hyperplane in continuous feature space. However, one drawback of this approach is that the size of pattern-library keeps on increasing over time. One solution to this problem is to use the recent monitored patterns, and discard the far away ones.

Chapter 7 applied the proposed online monitoring and prediction framework to two challenging real-world problems based on EEG time series data. With significant prediction results, the proposed adaptive prediction schemes ATP and ALP successfully demonstrated their superior prediction ability for online monitoring and prediction of massive non-stationary time series data. In the seizure prediction problem, the ALP prediction scheme achieved a little better prediction performance than the ATP prediction scheme. The overall averaged testing prediction accuracies for the ALP scheme and ATP scheme are 80% and 77%, respectively. In the driving EEG prediction problem, the ATP scheme generated a considerable better prediction performance than the ALP scheme. Using the best training settings, the overall averaged testing prediction accuracies for the ATP scheme and ALP scheme are 82% and 72%, respectively. We notice that one drawback of the ALP scheme is that its prediction performance may be seriously deteriorated due to online noises around the decision boundary which may lead to many false predictions. On the other hand, the ATP scheme is not sensitive to pattern noises and outliers. A prediction is only triggered if the monitored pattern cluster is an identified pre-event pattern cluster in the pattern library. All other monitored patterns including noises and outliers cannot trigger any warning alarm. This makes the proposed probabilistic ATP prediction very attractive in real-life applications.

In general, the proposed adaptive online prediction framework generated very attractive prediction performances on the two challenging online prediction problems. The general structure of the online monitoring and prediction framework enable it to be applicable to a wide range of prediction problems of complex time series events using a prediction horizon. The proposed framework is a fundamental contribution to the field of time series data mining, especially for the analysis of multichannel nonstationary and chaotic time series data.

8.2 Future Research

Future work can be proceed in the following directions:

- Apply the proposed online monitoring and prediction framework to many other time series prediction problems. Such as the online prediction of financial time series events. The temporal pattern analysis of financial time series has been demonstrated useful to predict some specific events, such as abrupt large rises or drops of stock prices. We will evaluate the prediction approach on prediction of financial time series event in the future.

- Investigate more on online incremental learning and prediction approaches. In the prediction framework, the prediction rule is retrained after each occurrence of a target event. In the retrain process of ALP scheme, the new decision boundary is obtained by training on the whole updated pattern library again, and did not use any information about the current information. In an online prediction model, it is desirable to achieve incremental online learning. That is we do not necessary to train the decision boundary from the very beginning. If new patterns are added into the pattern library, the new decision boundary is adjusted only use the information of the current boundary and the new pattern samples. Online SVM model is a promising direction to achieve this goal.

- Develop new feature selection algorithms. In this research, we employ the popular feature selection approach, the Pudil's floating search. Feature selection is an very important problem for pattern classification systems. How to select good features are very important to many data mining applications. However, we notice that the current feature selection algorithms are far from perfect. Based on the concept of minimum correlation and maximum relevance, we may proceed further to develop new feature selection algorithms which could extract the most important features embedded in a high dimensional space.

- Develop new time series distance measure based on the proposed time series segmentation algorithm TSTD. A possible application is to facilitate the calculation

of the current dynamic time warping (DTW) algorithms. A preliminary study of the skeleton-based dynamic time warping (SDTW) algorithm is discussed in the following subsection. More experiments are needed to evaluate the performance and limitations of SDTW.

References

[1] H.D.I. Abarbanel. *Analysis of observed chaotic data*. New York: Springer, 1996.

[2] N. Addison. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine, and finance*. Taylor & Francis, 2002.

[3] R. Agarwal and J. Gotman. Adaptive segmentation of electroencephalographic data using a nonlinear energy operator. *Proceedings of 1999 IEEE International Symposium on Circuits and Systems*, 4:199–202, 1999.

[4] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[5] A. Anari and J.W. Kolari. *Engineering Statistics Handbook*. Springer, 2009.

[6] C. W. Anderson and Z. Sijercic. Classication of EEG signals from four subjects during five mental tasks. *In Solving Engineering Problems with Neural Networks: Proceedings of the International Conference on Engineering Applications of Neural Networks*, 1996.

[7] Charles W. Anderson, Erik A. Stolz, and Sanyogita Shamsunder. Discriminating mental tasks using EEG Represented by AR models. In *In Proceedings of the 1995 IEEE Engineering in Medicine and Biology Annual Conference*, 1995.

[8] S. Arndt, G. Tyrrell, R.F. Woolson, M. Flaum, and N.C. Andreasen. Effects of errors in a multicenter medical study: Preventing misinterpreted data. *Journal of Psychiatric Research*, 28(5):447–459, 1994.

[9] R. Aschenbrenner-Scheibe, T. Maiwald, M. Winterhalder, H.U. Voss, J. Timmer, and A. Schulze-Bonhage. How well can epileptic seizures be predicted? An evaluation of a nonlinear method. *Brain*, 126:2616–2626, 2003.

[10] G.A. Barreto, R.A. Frota, and F.N.S. de Medeiros. On the classification of mental tasks: a performance comparison of neural and statistical approaches. *In Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*, 2004.

[11] Donald J. Berndt and James Clifford. Finding patterns in time series: a dynamic programming approach. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in knowledge discovery and data mining*, pages 229–248. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[12] H. Bhaskar, D.C. Hoyle, and S. Singh. Machine learning in bioinformatics: A brief survey and recommendations for practitioners. *Computers in Biology and Medicine*, 36:1104–1125, 2006.

[13] B. Blankertz, G. Curio, and K.R. Müller. Classifying single trial EEG: towards brain computer interfacing. *Advances in Neural Information Processing Systems*, 14(2):157–164, 2002.

[14] B.L. Bowerman and R.T. OConnell. *Forecasting and time series: an applied approach, 3rd ed.* Duxbury Press, Belmont, California, 1993.

[15] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis, Forecasting and Control.* Prentice Hall, 1994.

[16] K. Chakrabarti, E.J. Keogh, S. Mehrotra, and M.J. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems*, 27(2):188–228, 2002.

[17] K. Chan and A.W. Fu. Efficient time series matching by wavelets. In *Proceedings of 15th IEEE International Conference on Data Engineering*, pages 126–133, Sydney, Australia, 1999.

[18] W. Chaovalitwongse, Y.J. Fan, and R.C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research*, 56(6):1450–1460, 2008.

[19] W. Chaovalitwongse, L.D. Iasemidis, P.M. Pardalos, P.R. Carney, D.-S. Shiau, and J.C. Sackellares. Performance of a seizure warning algorithm based on the dynamics of intracranial eeg. *Epilepsy Research*, 64(3):93–113, 2005.

[20] W. Chaovalitwongse and P. Pardalos. On the time series support vector machine using dynamic time warping kernel for brain activity classification. *Cybernetics and Systems Analysis*, 44:125–138, 2008.

[21] W. Chaovalitwongse, P.M. Pardalos, L.D. Iasemidis, D.S. Shiau, and J.C. Sackellares. Dynamical approaches and multi-quadratic integer programming for seizure prediction. *Optimization Methods and Software*, 20(2-3):383–394, 2005.

[22] W. Chaovalitwongse, W. Suharitdamrong, C.C. Liu, and M.L. Anderson. Brain network analysis of seizure evolution. *Annales Zoologici Fennici*, 45(5):402–414, 2008.

[23] W.A. Chaovalitwongse, Y.J. Fan, and R.C. Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37:1005–1016, 2007.

[24] W.A. Chaovalitwongse, R.S. Pottenger, S. Wang, Y.J. Fan, and L.D. Iasemidis. The statistics of a practical seizure warning system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(5):977–988, 2011.

[25] L. Chisci, A. Mavino, G. Perferi, M. Sciandrone, C. Anile, G. Colicchio, and F. Fuggetta. Real-time epileptic seizure prediction using AR models and support

vector machines. *IEEE Transactions on Biomedical Engineering*, 57(5):1124–1132, 2010.

[26] B. Cox, T. Kislinger, and A. Emili. Integrating gene and protein expression data: pattern analysis and profile mining. *Methods*, 35(3):303–314, 2005.

[27] M. D'Alessandro, R. Esteller, G. Vachtsevanos, A. Hinson, J. Echauz, and B. Litt. Epileptic seizure prediction using hybrid feature selection over multiple intracranial EEG electrode contacts: a report of four patients. *Biomedical Engineering, IEEE Transactions on*, 50(5):603 –615, 2003.

[28] S. Dangel, P.F. Meier, H.R. Moser, S. Plibersek, and Y. Shen. Time series analysis of sleep EEG. *Computer assisted Physics*, pages 93–95, 1999.

[29] C. Davatzikos, K. Ruparel, Y. Fan, D.G. Shen, M. Acharyya, J.W. Loughead, R.C. Gur, and D.D. Langleben. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *NeuroImage*, 28(3):663–668, 2005.

[30] Z. Deng, F. Chung, and S. Wang. Clustering-inverse: A generalized model for pattern-based time series segmentation. *Journal of Intelligent Learning Systems and Applications*, 3(1):26–36, 2011.

[31] R.O. Duda, P.E. Hart, and D.G. Stork. *Unsupervised Learning and Clustering (2nd edition)*. Wiley, New York, 2001.

[32] M.A. Efroymson. *Mathematical Methods for Digital Computers*, chapter Multiple regression analysis, pages 191–203. New York: Wiley, 1960.

[33] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng. Learning CPG-based biped locomotion with a policy gradient method: application to a humanoid robot. *The International Journal of Robotics Research*, 27(2):213–228, 2008.

[34] J. Engel and T.A. Pedley. *Epilepsy: A Comprehensive Textbook*. Lippincott Williams & Wilkins, Philadelphia, PA, 1997.

[35] R. Esteller, J. Echauz, T. Cheng, B. Litt, and B. Pless. Line length: An efficient feature for seizure onset detection. *Proceedings of the 23rd International Conference of IEEE Engineering Medicine Biology Society*, 2:1707–1710, 2001.

[36] R. Esteller, J. Echauz, and T. Tcheng. Comparison of line length feature before and after brain electrical stimulation in epileptic patients. *Proceedings of the 26th International Conference of IEEE Engineering Medicine Biology Society*, pages 4710–4713, 2004.

[37] T.P. Exarchos, A.T. Tzallas, D.I. Fotiadis, S. Konitsiotis, and S. Giannopoulos. EEG transient event detection and classification using association rules. *IEEE Transactions on Information Technology in Biomedicine*, 10(3):451–457, 2006.

[38] M. Falkenstein, J. Hoormann, S. Christ, and J. Hohnsbein. ERP components on reaction errors and their functional significance: a tutorial. *Biological Psychology*, 51(2-3):87–107, 2000.

[39] S. Fazli, F. Popescu, M. Danoczy, B. Blankertz, K.R. Muller, and C. Grozea. Subject-independent mental state classification in single trials. *Neural networks*, 22(9):1305–1312, 2009.

[40] H. Feldwisch-Drentrup, B. Schelter, M. Jachan, J. Nawrath, J. Timmer, and A. Schulze-Bonhage. Joining the benefits: combining epileptic seizure prediction methods. *Epilepsia*, 51(8):1598–1606, 2010.

[41] Hao Feng and Chan Choong Wah. Online signature verification using a new extreme points warping technique. *Pattern Recognition Letters*, 24:2943–2951, 2003.

[42] Tony Finch. Incremental calculation of weighted mean and variance. Technical report, University of Cambridge, 2009.

[43] E. Fink, K.B. Pratt, and H.S. Gandhi. Indexing of time series by major minima and maxima. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, page 2332C2335, 2003.

[44] Tak Chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[45] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick. Online segmentation of time series based on polynomial least-squares approximations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):2232–2245, 2010.

[46] K. Fukunaga. *Statistical Pattern Recognition, seconde edition.* Academic Press, 1990.

[47] G.N. Garcia, T. Ebrahimi, and J.M. Vesin. Support vector EEG classification in the fourier and time-frequency correlation domains. *In Conference Proceedings of the First International IEEE EMBS Conference on Neural Engineering*, 2003.

[48] Andrew Britton Gardner. *A Novelty Detection Approach to Seizure Analysis from Intracranial EEG.* PhD thesis, Georgia Institute of Technology, 2004.

[49] E.S. Gardner. Exponential smoothing: the state of the art. *Journal of Forecasting*, 4(1):1–28, 1985.

[50] E.S. Gardner. Exponential smoothing: The state of the art–part ii. *International Journal of Forecasting*, 22(4):637–666, 2006.

[51] D. Garrett, D.A. Peterson, C.W. Anderson, and M.H. Thaut. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):141–144, 2003.

[52] S. Geisser. *Predictive Inference.* New York: Chapman and Hall, 1993.

[53] G. Getz, H. Gal, I. Kela, D.A. Notterman, and E. Domany. Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics*, 19:1079–1089, 2003.

[54] L.B. Good, S. Sabesan, S.T. Marsh, K. Tsakalis, D.M. Treiman, and L.D. Iasemidis. Nonlinear dynamics of seizure prediction in a rodent model of epilepsy. *Nonlinear Dynamics Psychol Life Science*, 14(4):411–434, 2010.

[55] S.M. Haas, M.G. Frei, and I. Osorio. Strategies for adapting automated seizure detection algorithms. *Medical Engineering & Physics*, 29(8):895–909, 2007.

[56] T.C. Handy. *Event-Related Potentials: A Methods Handbook*. The MIT Press, Cambridge, MA, 2004.

[57] E.J. Hannan. The estimation of the order of an arma process. *Annals of Statistics*, 8(5):1071–1081, 1980.

[58] E.J. Hannan and B.G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society*, B(41):190–195, 1979.

[59] A.C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1991.

[60] K. Holden, D.A. Peel, and J. L. Thompson. *Economic Forecasting: An Introduction*, chapter Introduction to Forecasting Mehtods. Cambridge University Press, 1991.

[61] R.D. Holowczak and B.S. Donefer. Growth and challenges in the equity options market. Technical report, Zicklin School of Business, Baruch College, City University of New York, 2008.

[62] J. Hunter and N. McIntosh. *Knowledge-based event detection in complex time series data*, chapter Artificial Intelligence in Medicine, pages 271–280. Springer, 1999.

[63] L.D. Iasemidis. *On the dynamics of the human brain in temporal lobe epilepsy*. PhD thesis, University of Michigan, Ann Arbor, 1991.

[64] L.D. Iasemidis, J.C. Principe, J.M. Czaplewski, R.L. Gilman, S.N. Roper, and J.C. Sackellares. Spatiotemporal transition to epileptic seizures: A nonlinear dynamical analysis of scalp and intracranial EEG recordings. In *Spatiotemporal Models in Biological and Artificial Systems*, pages 81–88. Amsterdam: IOS Press, 1997.

[65] L.D. Iasemidis and J.C. Sackellares. Long time scale spatio-temporal patterns of entrainment in preictal ECoG data in human temporal lobe epilepsy. *Epilepsia*, 31:621, 1990.

[66] L.D. Iasemidis, D.S. Shiau, W. Chaovalitwongse, J.C. Sackellares, P.M. Pardalos, J.C. Principe, P.R. Carney, A. Prasad, B. Veeramani, and K. Tsakalis. Adaptive epileptic seizure prediction system. *IEEE Transactions on Biomedical Engineering*, 50(5):616–627, 2003.

[67] L.D. Iasemidis, D.S. Shiau, P.M. Pardalos, W. Chaovalitwongse, K. Narayanana, A. Prasada, K. Tsakalis, P.R. Carney, and J.C. Sackellares. Long-term prospective online real-time seizure prediction. *Clinical Neurophysiology*, 116:532–544, 2005.

[68] L.D. Iasemidis, H.P. Zaveri, J.C. Sackellares, and W.J. Williams. Linear and nonlinear modeling of ECoG in temporal lobe epilepsy. *25th Annual Rocky Mountain Bioengineering Symposium*, 24:187–193, 1988.

[69] L.D. Iasemidis, H.P. Zaveri, J.C. Sackellares, W.J. Williams, and T.W. Hood. Nonlinear dynamics of electrocorticographic data. *Journal of Clinical Neurophysiology*, 5:339, 1988.

[70] J.F. Kaiser. On a simple algorithm to calculate the energy of a signal. *Proceedings of 1990 International Conference of Acoustis, Speech, Signal Processing*, 1:381–384, 1990.

[71] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter. BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engeneering*, 51(6):1073–1076, 2004.

[72] M. Kawado, S. Hinotsu, Y. Matsuyama, T. Yamaguchi, S. Hashimoto, and Y. Ohashi. A comparison of error detection rates between the reading aloud method and the double data entry method. *Controlled Clinical Trials*, 24(5):560–569, 2003.

[73] E. Keogh. A fast and robust method for pattern matching in time series databases. In *proceedings of 9th International Conference on Tools with Artificial Intelligence*, 1997.

[74] E. Keogh, S. Chu, D. Hart, and M. Pazzani. *Segmenting Time Series: A Survey and Novel Approach*, chapter Data Mining in Time Series Databases, pages 1–21. World Scientific Publishing, 2nd ed. edition, 2003.

[75] E. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. In *proceedings of Third International Conference on Knowledge Discovery and Data Mining*, Newport Beach, California, 1997.

[76] E. Keogh, Q. Zhu, B. Hu, Hao. Y., X. Xi, L. Wei, and C.A. Ratanamahatana. The ucr time series classification/clustering homepage: www.cs.ucr.edu/∼eamonn/time_series_data/, 2011.

[77] E.J. Keogh and M.J. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *proceedings of AAAI Workshop on Predicting the Future: AI Approaches to Time-Series Analysis*, Madison, Wisconsin, 1998.

[78] J. Khan, J.S. Wei, M. Ringnér, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P.S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679, 2001.

[79] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, volume 2. Addison-Wesley, Boston, 3rd edition edition, 1998.

[80] L.H. Koopmans. *The Spectral Analysis of Time Series*. Academic Press, 1995.

[81] T. Koskela. *Neural Network Methods In Analysing And Modelling Time Varying Processes*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 2003.

[82] A. Koski, M. Juhola, and M. Meriste. Syntactic recognition of ecg signals by attributed finite automata. *Pattern Recognition*, 28(12):1927–1940, 1995.

[83] S. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica Journal*, 31:249–268, 2007.

[84] T.N. Lal, T. Hinterberger, G. Widman, M. Schröer, J. Hill, W. Rosenstiel, C.E. Elger, B. Schökopf, and N. Birbaumer. *Advances in Neural Information Processing Systems*, volume 17, chapter Methods towards invasive human brain computer interfaces, pages 737–744. MIT Press, 2005.

[85] H.H. Lange, J.P. Lieb, J. Engel, and P.H. Crandall. Temporo-spatial patterns of preictal spike activity in human temporal lobe epilepsy. *Electroencephalography and Clinical Neurophysiology*, 56:543–555, 1983.

[86] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Mining of concurent text and time series. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 37–44, 2000.

[87] Y. Lee and C.K. Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19:1132–1139, 2003.

[88] K. Lehnertz, R. Andrzejak, J. Arnhold, T. Kreuz, F. Morman, C. Rieke, G. Widman, and C. E. Elger. Nonlinear EEG analysis in epilepsy: its possible use for interictal focus localization, seizure anticipation and prevention. *Journal of Clinical Neurophysiology*, 18:209–222, 2001.

[89] K. Lehnertz and C. Elger. Can epileptic seizures be predicted? evidence from nonlinear time series analysis of brain electrical activity. *Physics Review Letters*, 80:5019–5022, 1998.

[90] K. Lehnertz and B. Litt. The first international collaborative workshop on seizure prediction: summary and data description. *Clinical Neurophysiology*, 116(3):493–505, 2005.

[91] D. Lemire. A better alternative to piecewise linear time series segmentation. In *Proceedings of the 7th SIAM International Conference on Data Mining(SDM 2007)*, pages 545–550, Minneapolis, Minnesota, USA, 2007.

[92] Hailin Li, Chonghui Guo, and Wangren Qiu. Similarity measure based on piecewise linear approximation and derivative dynamic time warping for time series mining. *Expert Systems with Applications*, 38(12):14732–14743, 2011.

[93] L. Li, H. Tang, Z. Wu, J. Gong, M. Gruidl, J. Zou, M. Tockman, and R. Clark. Data mining techniques for cancer detection using serum proteomic profiling. *Artificial Intelligence in Medicine*, 32:71–83, 2004.

[94] C.-J. Lin and C. Wu. Detecting typing errors in a numerical typing task with linear discriminant analysis of single trial EEG. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 53(10):595–599, 2009.

[95] J. Lin, E. Keogh, Lonardi, S., and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Workshop on Research Issues in Data Mining and Knowledge Discovery, 8th ACM SIGMOD*, San Diego, CA, Jun 13 2003.

[96] B. Litt, R. Esteller, J. Echauz, M. D'Alessandro, R. Shor, T. Henry, P. Pennell, C. Epstein, R. Bakay, and M. Dichter. Epileptic seizures may begin hours in advance of clinical onset: a report of five patients. *Neuron*, 30:51–64, 2001.

[97] Xiaoyan Liu, Zhenjiang Lin, and Huaiqing Wang. Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering*, 20(12):1616 –1626, 2008.

[98] S. Lu, C. Guan, and H. Zhang. Subject-independent brain computer interface through boosting. In *The 19th International Conference on Pattern Recognition*, pages 1–4, Tampa, FL, Dec. 2008.

[99] O.L. Mangasarian and E.W. Wild. Proximal support vector machine classifiers. In *Proceedings of Knowledge Discovery and Data Mining*, pages 77–86, 2001.

[100] J. Martin-Guerrero, F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Marti, and N. Jiménez-Torres. A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients. *Expert Systems with Applications*, 36(6):9737–9742, 2009.

[101] J.J. McKee, N.E. Evans, and F.J. Owens. Efficient implementation of the Fan/SAPA-2 algorithm using fixed point arithmetic. *Automedica*, 16:109–117, 1994.

[102] S. Mika. *Kernel Fisher Discriminants*. PhD Thesis, Department of Computer Scicence, University of Technology, Berlin, Germany, 2002.

[103] A. Moran, I. Bar-Gad, H. Bergman, and Z. Israel. Real-time refinement of subthalamic nucleus targeting using bayesian decision-making on the root mean square measure. *Movement disorders*, 21(9):1425–1431, 2006.

[104] F. Mormann, R.G. Andrzejak, C.E. Elger, and K. Lehnertz. Seizure prediction: The long and winding road. *Brain*, 130(2):314–333, 2007.

[105] F. Mormann, R.G. Andrzejak, T. Kreuz, C. Rieke, P. David, C.E. Elger, and K. Lehnertz. Automated detection of a preseizure state based on a decrease in synchronization in intracranial electroencephalogram recordings from epilepsy patients. *Physical Review E*, 67:021912, 2003.

[106] F. Mormann, T. Kreuz, C. Rieke, R. Andrzejak, A. Kraskov, P. David, C. Elger, and K. Lehnertz. On the predictability of epileptic seizures. *Journal of Clinical Neurophysiology*, 116(3):569–587, 2006.

[107] K. Natarajan, R. Acharya U, F. Alias, T. Tiboleng, and S.K. Puthusserypady. Nonlinear analysis of EEG signals at different mental states. *Biomedical engineering Online*, 3:7, 2004.

[108] S. Nieuwenhuis, K. Ridderinkhof, J. Blom, G. Band, and A. Kok. Error-related brain potentials are differentially related to awareness of response errors: evidence from an antisaccade task. *Psychophysiology*, 38(5):752–760, 2001.

[109] H. Ocak. Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. *Signal Processing*, 88(7):1858–1867, 2008.

[110] D. Olsen, R. Lesser, J. Harris, R.Webber, and J. Cristion. Automatic detection of seizures using electroencephalographic signals. *U.S. Patent 5311876*, 1994.

[111] Ivan Osorio, Mark G. Frei, and Steven B. Wilkinson. Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset. *Epilepsia*, 39(6):615–627, 1998.

[112] T. Palpanas, M. Vlachos, E. Keogh, and D. Gunopulos. Streaming time series summarization using user-defined amnesic functions. *Knowledge and Data Engineering, IEEE Transactions on*, 20(7):992–1006, july 2008.

[113] S. Park, D. Lee, and W.W. Chu. Fast retrieval of similar subsequences in long sequence databases. In *Proceedings of the 3rd IEEE Knowledge and Data Engineering Exchange Workshop*, pages 60–67, 1999.

[114] Sanghyun Park, Sang-Wook Kim, and Wesley W. Chu. Segment-based approach for subsequence searches in sequence databases. In *Proceedings of the 2001 ACM symposium on Applied computing*, pages 248–252, 2001.

[115] L.C. Parra, C.D. Spence, A.D. Gerson, and P. Sajda. Response error correction-a demonstration of improved human-machine performance using real-time eeg monitoring. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):173–177, 2003.

[116] T. Pavlidis and S. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, 1(8):860–870, 1974.

[117] H.B.I. Persson, K.A. Alberts, B.Y. Farahmand, and T. Tomson. Risk of extremity fractures in adult outpatients with epilepsy. *Epilepsia*, 43(7):768–772, 2002.

[118] R. Pindyck and D. Rubinfeld. *Econometric Models and Economic Forecasts*. McGraw-Hill/Irwin, 1997.

[119] R.T. Pivik, R.J. Broughton, R. Coppola, R.J. Davidson, N. Fox, and M.R Nuwer. Guidelines for the recording and quantitative analysis of electroencephalographic activity in research contexts. *Psychophysiology*, 30(6):547–558, 1993.

[120] K. Polat and S. G*üe*s. Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform. *Applied Mathematics and Computation*, 187(2):1017–1026, 2007.

[121] H. Potter. Anatomy of the brain. http://faculty.ucc.edu/biology-potter/TheBrain/, 2006.

[122] R.J. Povinelli and Xin Feng. A new temporal pattern identification method for characterization and prediction of complex time series events. *Knowledge and Data Engineering, IEEE Transactions on*, 15(2):339–352, march-april 2003.

[123] B. Pratt and E. Fink. Search for patterns in compressed time series. *International Journal of Image and Graphics*, 2(1):89–106, 2002.

[124] P.W.Ferrez and J.d.R. Millán. Error-related eeg potentials generated during simulated brain-computer interaction. *IEEE Transactions on Biomedical Engineering*, 55:923–929, 2008.

[125] R.Q. Quiroga, J. Arnhold, K. Lehnertz, and P. Grassberger. Kulback-leibler and renormalized entropies: applications to electroencephalograms of epilepsy patients. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 62:8380–8386, 2000.

[126] M. Le Van Quyen, V. Navarro, M. Baulac, B. Renault, and J. Martinerie. Anticipation of epileptic seizures from standard EEG recordings. *The Lancet*, 361(9361):970–971, 2003.

[127] M.L.V. Quyen, J. Soss, V. Navarro, R. Robertson, M. Chavez, M. Baulac, and J. Martinerie. Preictal state identification by synchronization changes in long-term intracranial EEG recordings. *Clinical Neurophysiology*, 116:559–568, 2005.

[128] P. Rajdev, M.P. Ward, J. Rickus, R. Worth, and P.P. Irazoqui. Real-time seizure prediction from local field potentials using an adaptive wiener algorithm. *Computers in biology and medicine*, 40(1):97–108, 2010.

[129] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado. *Artificial Neural Networks: Biological Inspirations ICANN 2005*, volume 3696, chapter Ensemble of SVMs for Improving Brain Computer Interface P300 Speller Performances, pages 45–50. Springer Berlin / Heidelberg, 2005.

[130] Ira J.M.D. Rampil and Michael J.D.V.M. Laster. No correlation between quantitative electroencephalographic measurements and movement response to noxious stimuli during isoflurane anesthesia in rats. *Anesthesiology*, 77:920–925, 1992.

[131] P.E. Rapp, T. Bashore, J. Martinerie, A. Albano, I. Zimmerman, and A. Mess. Dynamics of brain electrical activity. *Brain Topography*, 2:99–118, 1989.

[132] S. Richter, D. Aberdeen, and J. Yu. Natural actor-critic for road traffic optimisation. In *Advances in neural information processing systems*, pages 1169–1176, Cambridge, MA, 2007. MIT Press.

[133] O.A. Rosso, M.T. Martin, A. Figliola, K. Keller, and A. Plastino. EEG analysis using wavelet-based information tools. *Journal of Neuroscience Methods*, 153(2):163–182, 2006.

[134] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

[135] J.C. Sackellares, D.S. Shiau, J.C. Principe, Mark C.K. Yang, L.K. Dance, W. Suharitdamrong, W. Chaovalitwongse, P.M. Pardalos, and L.D. Iasemidis. Predictability analysis for an automated seizure prediction algorithm. *Journal of Clinical Neurophysiology*, 23(6):509–520, 2006.

[136] Y. Salant, I. Gath, and O. Henriksen. Prediction of epileptic seizures from two-channel EEG. *Medical and Biological Engineering and Computing*, 36:549–556, 1998.

[137] T.A. Salthouse. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin*, 99(3):303–319, 1986.

[138] S. Scholtus. Algorithms for correcting some obvious inconsistencies and rounding errors in business survey data. *Statistics Netherlands*, Discussion Paper 08015, 2008.

[139] S. Scholtus. Automatic correction of simple typing errors in numerical data with balance edits. *Statistics Netherlands*, Discussion Paper 09046, 2009.

[140] G.E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[141] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. New York: John Wiley & Sons, 1989.

[142] P. Senin. Dynamic time warping algorithm review. Technical report, Information and Computer Science Departament University of Hawaii, Honolulu, 2008.

[143] A. Sfetsos and C. Siriopoulos. Time series forecasting with a hybrid clustering scheme and pattern recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):399–405, 2004.

[144] H. Shatkay and S.B. Zdonik. Approximate queries and representations for large data sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545, feb-1 Mar 1996.

[145] K.Q. Shen, X.P. Li, C.J. Ong, S.Y. Shao, and Einar P.V. Wilder-Smith. EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimate. *Clinical Neurophysiology*, 119(7):1524–1533, 2008.

[146] J. Shukla. Predictability in the midst of chaos: A scientific basis for climate forecasting. *Science*, 282(5389):728–731, 1998.

[147] C. Silva, I.R. Pimentel, A. Andrade, J.P. Foreid, and E. Ducla-Soares. Correlation dimension maps of EEG from epileptic absences. *Brain Topography*, 11:201–209, 1999.

[148] D.E. Snyder, J.E., D.B. Grimes, and B. Litt. The statistics of a practical seizure warning system. *Journal of Neural Engineering*, 5(4):392, 2008.

[149] V. Srinivasan and C. Eswaran. Approximate entropy-based epileptic EEG detection using artificial neural networks. *IEEE Transactions on Information Technology in Biomedicine*, 11(3):288–295, 2007.

[150] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–147, 1974.

[151] A. Subasi. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32:1084–1093, 2007.

[152] X. Sun, M.E. Orlowska, and X. Li. Finding temporal features of event-oriented patterns. *Proceedings of the 9th Paci*

c-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05), pages 778–784, 2005.

[153] X. Sun, M.E. Orlowska, and X. Zhou. Finding event-oriented patterns in long temporal sequences. *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'03)*, pages 16–26, 2003.

[154] R. Sutton and A. Barto. *Reinforcement learning: An Introduction*. MIT Press, 1998.

[155] S. Trewin. An invisible keyguard. In *Assets '02: Proceedings of the fifth international ACM conference on Assistive technologies*, pages 143–149, New York, NY, USA, 2002. ACM.

[156] J.A. Vastano and E.J. Kostelich. Comparison of algorithms for determining Lyapunov exponents from experimental data. In *International conference on dimensions and entropies in chaotic systems*, pages 100–107, Pecos River, NM, USA, 1985.

[157] S.S. Viglione and G.O. Walsh. Epileptic seizure prediction. *Electroencephalography and Clinical Neurophysiology*, 39:435–436, 1975.

[158] Shouyi Wang and Wanpracha Art Chaovalitwongse. *Evaluating and Comparing Forecasting Models*. John Wiley & Sons, Inc., 2010.

[159] G.H. Weiss. *Aspects and Applications of the Random Walk*. North Holland Press, Amsterdam, 1994.

[160] S. Wong, G.H. Baltuch, J.L. Jaggi, and S.F. Danish. Functional localization and visualization of the subthalamic nucleus from microelectrode recordings acquired during DBS surgery with unsupervised machine learning. *Journal of Neural Engineering*, 6:026006, 2009.

[161] C. Wu and Y. Liu. Queuing network modeling of transcription typing. *ACM Transactions on Computer-Human Interaction*, 15(1):1–45, 2008.

[162] W. Xu, C. Guan, C.E. Siong, S. Ranganatha, M. Thulasidas, and J. Wu. High accuracy classification of EEG signal. *In 17th International Conference on Pattern Recognition*, pages 391–394, 2004.

[163] D.M. Young. Data inaccuracy in the global transportation network. Master thesis, Wright-Patterson Air Force Base, Air Force Institute of Technology, OH, USA, 1996.

[164] G. Zhang, B.E. Patuwo, and M.Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998.