Proxy Relearning for Feature-Driven Pattern Recognition in High-Dimensional Imbalanced Time Series Data Sets

by

Wilfred Yau-Chuen Cho

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science Graduate Department of Electrical and Computer Engineering University of Toronto

© Copyright 2017 by Wilfred Yau-Chuen Cho
Abstract

Proxy Relearning for Feature-Driven Pattern Recognition in High-Dimensional Imbalanced Time Series Data Sets

Wilfred Yau-Chuen Cho
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2017

This thesis explores the possibility of feature-driven time series pattern recognition from both practical and theoretical perspectives for predictive modelling in a situation where data are imbalanced, minority class examples are scarce, the ratio of feature dimension to sample size is high, and the class labels provided might not be optimized for the application. These problems are common in learning patient-specific patterns in medical and health domains, where labels provided by medical experts might not fit the goal of predictive modelling. Extracting informative labels for supervised learning is a difficult and time-consuming task. A novel strategy is proposed to solve the problems mentioned above, which aims to reduce human effort by automatically finding the earliest pattern that a classifier can recognize. The proposed algorithm locates and learns similar patterns across training examples that maximize the difference between both classes. This method ensures precise learning and boosts the performance of classifier by reducing the number of false positives. The performance of the algorithm was evaluated based on the classification results and the anticipation responses on the data provided by EPILEPSIAE, a European Epilepsy Database. An average false positive of 0.0519 per hour was achieved using the proposed algorithm with a sensitivity of 0.79 in anticipating seizures.
A dedication to life and its stochasticity,  
that something seems inevitably erratic,  
can make someone feel genuinely unique.

Through the challenge of foretelling,  
brings us the gift of imagination,  
as a weapon in the gloom of certainty.

A purpose of being,  
the contemplation of ambiguity,  
until one reaches its destiny.
Acknowledgements

Thesis Committee:
Prof. Paul Yoo (Chair)
Prof. Konstantinos N. Plataniotis
Prof. Stark Draper

To live, not for the things we already know, but for the things we have yet to discover. The unknown grants us with bountiful gifts; the gift of faith, the gift of hope, and the gift of imagination. It is these gifts that give us a purpose. A reason to search and discover, to fight for what we believe in, to dream and plan and contemplate; to find meaning in the meaningless and to find destiny in the uncertainty. Our past is left behind and the present greedily demands our attention and yet we gaze, into the void where the future lies, and we know that it is there, though we cannot see it, just quite yet.

An acknowledgment to my loving Mom and Dad. There is no word to describe their continuing support and contribution to their son. Thank you, my dear sister, for the free food that fed my hungry stomach. A belly rub for my sweet little Kitty Soft Paws, who kept me company for a period of time.

Great thanks to my graduate adviser Prof. Roman Genov, for his immense patience for a stubborn student, his insightful suggestions on my blind side, his generosity for allowing me to pursue my ideas freely, his support throughout my research, his advice in the biomedical domain, and the opportunity to join his laboratory.

Many thanks to my other advisers: Dr. Taufik Valiante, Dr. David Groppe, and Prof. Jose Luis Perez Velazquez. Their inputs are the cornerstone of my thesis. A great appreciation to my thesis committee for their feedback and comments.

A shout-out to my fellow comrades: Ahmed, Asish, Dayeol, Gairik, Gerard, Javid, Maged, Navid, Nikola, Ryan, Reza, and Thai. The alone time we spent together and our conversations are most cherished.

Special thanks to Rebecca Jonsson from Sweden for turning my idea to the beautiful opening paragraph and her honest friendship.
# Table of Contents

Acknowledgements iv

Table of Contents vii

List of Tables viii

List of Figures ix

List of Abbreviations x

1 Introduction 1
   1.1 Motivation .................................................. 1
   1.2 Machine Learning Challenges .................................. 2
   1.3 Novelties List .................................................. 4
   1.4 Thesis Organization .......................................... 4

2 Background Methodology 6
   2.1 Machine Learning Approaches .............................. 6
      2.1.1 Ensemble Method ........................................ 6
      2.1.2 Divide-and-Conquer Algorithm ......................... 7
   2.2 Data Labelling ................................................ 7
   2.3 Classifiers ................................................ 8
      2.3.1 Support Vector Machine ................................ 8
      2.3.2 Logistic Regression .................................... 10
      2.3.3 Neural Network .......................................... 10
      2.3.4 Naive Bayes ............................................. 12

3 Implementation of The Alignment Algorithm and Classifier 14
   3.1 Algorithm Structure and Basis ........................... 14
   3.2 Learning Mechanics ......................................... 15
      3.2.1 Proxy Relearning Definition .......................... 15
      3.2.2 Proxy Relearning Approach ............................ 16
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.4 Application of Labels</td>
<td>46</td>
</tr>
<tr>
<td>6    Classification and Anticipation Horizon</td>
<td>49</td>
</tr>
<tr>
<td>6.1  Definitions</td>
<td>49</td>
</tr>
<tr>
<td>6.1.1 Training and Testing Conditions</td>
<td>49</td>
</tr>
<tr>
<td>6.2  Evaluation Problems and Methods</td>
<td>49</td>
</tr>
<tr>
<td>6.3  Choice of Gating Classifier</td>
<td>50</td>
</tr>
<tr>
<td>6.4  Cross-Validation and Testing Results</td>
<td>51</td>
</tr>
<tr>
<td>6.4.1 Optimization Parameters</td>
<td>51</td>
</tr>
<tr>
<td>6.4.2 Hold-Out Test Average and Best Results</td>
<td>51</td>
</tr>
<tr>
<td>6.5  Prediction and Detection Responses</td>
<td>56</td>
</tr>
<tr>
<td>6.5.1 Results and Types of Response</td>
<td>56</td>
</tr>
<tr>
<td>6.5.2 Learning Trade-off</td>
<td>57</td>
</tr>
<tr>
<td>7    Discussion</td>
<td>63</td>
</tr>
<tr>
<td>7.1  Outcome</td>
<td>63</td>
</tr>
<tr>
<td>7.2  Future Work</td>
<td>63</td>
</tr>
<tr>
<td>7.2.1 Feature Encoding</td>
<td>63</td>
</tr>
<tr>
<td>7.2.2 Long Term Attention Learning</td>
<td>64</td>
</tr>
<tr>
<td>7.2.3 Choice of Backbone Classifier</td>
<td>64</td>
</tr>
<tr>
<td>7.2.4 One-Class Learning</td>
<td>64</td>
</tr>
<tr>
<td>Bibliography</td>
<td>65</td>
</tr>
</tbody>
</table>
List of Tables

4.1 List of Patients ................................................................. 30
5.1 Learning and Leave-One-Out Cross-Validation Results .................. 45
6.1 Hold-Out Test Average Results .............................................. 53
6.2 Hold-Out Test Best Results .................................................. 54
List of Figures

1.1 Label Learning .............................................. 3
2.1 Support Vector Machine .................................. 9
2.2 Neural Network ............................................. 11
3.1 Algorithm Work Flow ..................................... 18
3.2 Classification Structure .................................. 23
4.1 Onset Comparison 1 ........................................ 27
4.2 Onset Comparison 2 ........................................ 28
4.3 Wavelet Phase Coherence Transformation .............. 33
4.4 Feature Engineering ....................................... 34
5.1 PAT635 Seizure 1 .......................................... 38
5.2 PAT635 Seizure 7 .......................................... 39
5.3 PAT970 Seizure 17 ......................................... 40
5.4 PAT590 Seizure 1 .......................................... 41
5.5 PAT590 Seizure 12 ......................................... 42
5.6 Leave-One-Out Cross-Validation ......................... 44
5.7 PAT818 Seizure 2 .......................................... 47
5.8 PAT818 Seizure 7 .......................................... 48
6.1 Leave-One-Out Cross-Validation with Hold-Out Test ........................................ 52
6.2 Box-plot of False Positive Rate and Sensitivity ........ 55
6.3 PAT1096 Seizure 8 Whole Event ......................... 57
6.4 PAT1096 Seizure 8 Response ............................... 58
6.5 PAT1096 Seizure 9 Whole Event ......................... 59
6.6 PAT1096 Seizure 9 Response ............................... 60
6.7 PAT442 Seizure 19 Response ............................... 61
6.8 PAT862 Seizure 6 Response ............................... 62
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEG</td>
<td>Electroencephalogram</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HR</td>
<td>Hour</td>
</tr>
<tr>
<td>LR</td>
<td>Logistic Regression</td>
</tr>
<tr>
<td>LOOCV</td>
<td>Leave-One-Out Cross-Validation</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>PLV</td>
<td>Phase Locking Value</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>S.F.</td>
<td>Sampling Frequency</td>
</tr>
<tr>
<td>SMOTE</td>
<td>Synthetic Minority Over-sampling Technique</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Building an adequate time series predictive pattern recognition model using imbalanced data set with ill-defined labels is almost an impossible task. Researchers often have to deal with a condition known as class imbalance due to the nature of that specific pattern (i.e. the class of interest for detection or prediction) occurs significantly less compared to classes of disinterest. In such scenario, reliance on the correctness of class label is extremely important. The precision of the labels are crucial to train a classifier and requirement of the preciseness is often down to the correct time sample. Human experts may have a difficult time to locate the corresponding timestamp due to the volatile nature of the data which could be difficult to understand or could not be completely understood. As we try to gain a better understanding of the nature of the data, researchers often use multiple complex feature engineering techniques to describe a single event. However, as we try to classify these patterns in increased layers of dimension, we encounter a problem known as the curse of dimensionality\[38\].

The process to find these precise labels for supervised learning are incredibly difficult, time-consuming and subjected to sole expert’s knowledge. Especially in the medical field\[31\], human experts may not be able to agree upon the placement of the labels, since the time series data were acquired at a high sampling rate and the temporal pattern of the data cannot be easily described. Even if the human experts can agree on the labels, it may not suit the application presuming that the prediction or detection of the pattern requires high complexity algorithm which computer hardware might not be able to support in a real-time manner. In a case where the application is bounded by computational or hardware constraints, a readjustment of the original labels is required for better operation-specific pattern recognition. Hence, to solve these troubling issues, a novel technique must be presented to find the application and operation suitable labels from previously labelled data for predictive pattern recognition.
1.2 Machine Learning Challenges

Traditional pattern recognition and prediction algorithms rely on the abundance of data. However, when minority class of the data is scarce and data are imbalanced, it could degrade the performance of the classifier[20]. Popular methods to resolve this kind of problems include One-Class learning[25] and Synthetic Minority Over-sampling Technique[18], where the former aims to train classifiers solely on abundant data from majority class and classify minority class samples as outliers, the latter aims to over-sample minority class by creating synthetic examples and undersample majority class to achieve a balance training set for better classification performance. For the purpose of detecting slight dynamic changes in order to predict events as early as possible, One-Class Support Vector Machine is not a suitable classifier since it fails to do so as it is insensitive to small changes. The difficulties in prediction and early detection of certain events are due to ill-defined patterns which may have a high variance through each recurrence. A good learning algorithm must be generalizable in high-dimensional space where training examples fail to provide information in those domains. As we increase the dimension and complexity of the features, performance of the classifier is still largely disappointing using SMOTE, due to its practice in resampling observed data. If the classes of the data are not easily separable, obtaining better labels is often more important than resampling.

Using mis-labelled training data or labels that are not optimized for application purposes has a huge impact on classification performance. A data set containing 20% mis-labelled data can reduce classification accuracy by 8% [12], not to mention the effects caused by extremely imbalanced data. A great model of classifier requires a perfect training set of patterns which are neatly clustered with good separation from different classes. However, grouping such abnormal time series patterns remains a challenge since the nature of the pattern might not be easily observed. Clustering methods are useful ways to understand the distribution of data, as well as the characteristics of the time series pattern. However, not only clustering in high-dimensional space is difficult[9], it is also impossible to create a practical, general, and realistic model of data behaviour if only given a handful of examples[14][33].

Popular data-driven approach to solve time series pattern recognition uses methods such as Recurrent Neural Networks (RNN) and Hidden Markov Model (HMM). Long Short-Term Memory (LSTM)[35], a practical RNN algorithm which aims to solve the long-term dependencies problem by capturing important events, had been proven in the field of machine learning with its excellence in speech recognition. However, LSTM and HMM often require a large amount of data to sample from or rely on a representable probability distribution. Hence, they are not practical choices for this thesis.

A better approach to build anticipation models relies on searching for bifurcation patterns that lead to the event of interest and precisely learn from these examples to train classifiers. However, manually finding these patterns could be a difficult and time-consuming task. Figure 1.1 illustrates the challenge of finding these patterns. Searching these patterns for training classifier could be very difficult because choices are often abundant, but the number of training
examples could be very little. Finding a pattern that can achieve generalization in predicting future events requires an autonomous pattern-finding mechanism.

![Figure 1.1: Label Learning](image)

An illustration of finding the best pattern that indicates the forthcoming of an event. Which duration and sample of patterns (i.e. A, B, or C and its corresponding window) could be most informative as regards the cause of the time series event?

The goal of this thesis is to implement a technique that addresses the following problems:

- **Obtaining Training Labels**: The cost of learning adequate labels could be high, due to the needs of well-compensated human experts. Moreover, human-labelled data might not be effective for the purpose of a given application. This causes a large amount of time being wasted in the readjustment of labels.

- **High-Dimensional Learning**: When the number of input sensors or feature dimension is relatively high comparing to that of training examples, dimension reduction at first might not be a wise approach especially in the process of learning sensitive differences between classes. To guarantee success, the perfection of feature engineering is needed, as well as an effective learning architect that can precisely learn the minor characteristics differences between classes of temporal pattern.

- **High-Dimensional Classification**: If the dimension is too high, classification could be dif-
ficult due to the curse of dimensionality, resulting in over-fitting since regions of feature dimension are not learnt properly. A classifier structure that can handle high-dimensional time series data is needed with a focus on suppressing false positive rate.

1.3 Novelties List

The proposed algorithm is embedded with novel concepts highlighted by:

- Feature-Driven Approach: Previously known labels might not be optimized for application and operation purposes. A proxy relearning algorithm introduced in this thesis maximizes the performance of classification given application constraints.

- Autonomous Label Learning: To minimize the cost of human experts, and time and effort in finding labels, the algorithm searches for the best timestamp labels across all training examples that maximize the differences between classes.

- Custom Anticipation Horizon: User can decide on building detective or predictive model, with a trade-off between the rate of false positives per hour and the duration of anticipation horizon to the event of interest, through setting different levels of learning parameters' threshold during automatic labelling.

- High Dimensional Learning and Classification: A classification architect aims to break down high-dimensional data into multiple lower dimensional subsets to ensure precision learning and classification, with a focus on recognizing multi-dimensional patterns.

- False Positives Suppression: The proposed classification structure is able to achieve an extremely low false positive on the test set, due to its ability to generalize by learning relevant and adequate class boundary.

1.4 Thesis Organization

This thesis focuses on two areas, which are labelling solution for the application and better machine learning structure.

First is to provide better labelling solution to locate time series pattern where the beginning of these patterns is difficult for experts to identify. One of the two motivations for this is that in the case of supervised learning, the label provided might not be optimized for the intended application, which could induce lower true positive rate and higher false positive rate during classification. The other motivation is to find the earliest bifurcation point in time where the classifier can predict a certain event. The reason is that given a wider time horizon to predict an event could potentially disturb or prevent it from happening through intervention. The focus is to adjust these labels to ensure patterns learnt are indeed separable. By using proxy
relearning defined labels, one can ensure classification is effective. Note that this thesis focuses on short-term anticipation, but not long-term forecasting.

The second area focuses on constructing a machine learning structure that aims to learn pattern in extreme imbalanced and high-dimensional data with minimum structural complexity. The motivation behind this is that most machine learning techniques are created to learn from a large balanced database, hence an alternative structure is required for high-dimensional imbalanced data sets. The other aspect is to minimize the complexity and allow us to implement the algorithm on real-world applications, where a constraint is imposed on the computational power.

The synopsis of the chapters are as follows:

- **Chapter 2: Background Methodology**
  This chapter provides an introduction to the underlying Machine Learning concepts used in this thesis.

- **Chapter 3: Implementation of The Alignment Algorithm and Classifier**
  This chapter provides the theoretical novel concept of proxy relearning, where users can make a decision on trading false positive rate for the anticipation horizon, thus fully explores the possibility between prediction and detection of events given application constraints.

- **Chapter 4: Testing of Methodology: Epilepsy**
  This chapter provides an introduction to epilepsy, related work in seizure anticipation, and an overview of choice of feature used to evaluate the algorithm.

- **Chapter 5: Learning Performance**
  This chapter illustrates the concept of predictability as a method to understand the relationship across training examples. Examples of seizure patterns that are predictable and unpredictable are displayed along with the algorithm defined labels.

- **Chapter 6: Classification and Anticipation Horizon**
  This chapter defines the training and testing condition, discusses the result of the unobserved test case. The anticipation horizon is also illustrated as well, showcasing the resourcefulness and the anticipation power of the algorithm.

- **Chapter 7: Discussion**
  This chapter concludes the performance of the algorithm and proposes other ideas to enhance the algorithm for future work.
Chapter 2

Background Methodology

In the field of classical statistics and pattern recognition, one often performs similarity measurements to understand the data using methods such as covariance, principal component analysis, and probability distribution. However, due to the nature of small data set with high dimensionality where the number of features is much greater than the number of observations, traditional methods simply cannot provide informative information\cite{3}\cite{23}. In this chapter, a few background methodologies are introduced to solve this problem.

2.1 Machine Learning Approaches

This section covers the basis of two types of machine learning approaches: ensemble method and Divide-and-Conquer algorithm. Both methods rely on the construction of multiple classifiers. The major difference is that ensemble methods focus on training multiple classifiers with manipulation of training method (e.g. methods of drawing training set), while Divide-and-Conquer algorithms focus on training classifier specialized in different subsets of input space.

2.1.1 Ensemble Method

Ensemble method uses the combination of multiple classifiers or simpler learners to make decisions based on the summation of weighted or unweighted classification from individual classifiers. Common techniques to train ensemble classifier include bagging, cross-validated committees, boosting algorithm, and stacking\cite{5}\cite{37}. Bagging and cross-validated committees share the concept of training classifier by dividing the training sets to train multiple classifiers, while the former randomly draws training examples from a smaller population of the original training set, the latter excludes part of the subset during training. AdaBoost, a type of boosting, aspires to learn from examples that are misclassified in order to improve the performance of classification by tuning weak learners in regions of misclassification. Stacking is a method that combines multiple classifiers’ decision and using that information with a combiner algorithm or a gating classifier to make a final decision.
The above methods enhance the performance of the predictive model by diminishing variance and bias during training. However, none of these methods addresses the concern of dimensionality.

2.1.2 Divide-and-Conquer Algorithm

Divide-and-Conquer algorithms in machine learning domain are methods that focus on covering different input regions with different learners. By splitting the problem into smaller sub-problems and solving the sub-problems independently, the solutions can be combined to solve the original problem. Examples of such algorithm are Classification and Regression Trees, Multivariate Adaptive Regression Splines, Iterative Dichotomiser 3, and Mixture of Experts[24]. A Mixture of Experts makes use of high-dimensional observations by dividing the input space into smaller sub-domain problems and infer informative decisions through a final gating network. Separating each dimension from vast observations can allow each expert to be a specialist of an individual domain. Divide-and-Conquer algorithms work well when the training set is small, particularly when the curse of dimensionality is a major concern.

2.2 Data Labelling

One of the major challenges of pattern recognition in time series domain is that the temporal patterns in the training set must be aligned to ensure the success of building a useful classifier. In the domain of supervised learning, one heavily relies on the human expert to label certain events correctly as ground truth with high precision. However, this poses a few problems realistically. As the pattern becomes more complex, human experts may find it difficult to agree upon the beginning and the ending of the pattern especially in the case of epileptic seizure[17][32]. Hence, labels are arbitrarily placed without standardization[36]. In a small data domain, one training example with a small labelling error could undermine the general performance of a classifier, especially inducing higher false positives. To make matters more complicated, multiple layers of feature engineering such as phase-phase coupling and amplitude-phase coupling, which aim to find patterns in a different domain, are completely different from the original observation. Hence, the original labels provided by the expert might not be suitable for these complex observations. The concept to identify if a pattern in training phase is indeed recognizable is an important step to build a practical classifier. However, in the domain of supervised learning, one assumes the data being labelled are indeed the absolute truth of the pattern or suitable for the application. Unfortunately, without properly understanding the reasons behind the placement of labels, blindly following the labels provided may not be the best method for building a predictive model.

One can conclude that supervised learning poses a major problem when dealing with small data set with inappropriate labelling due to its lack of flexibility in understanding and learning from noisy data [39]. It does not aim to explore the possibility in learning pattern given forward
or backward margin in the window of a pattern. Understanding the attribute difference between
data classes is more important than reliance on the supervised labels in building practical
predictive models. Unsupervised learning, such as expectation-maximization poses a powerful
concept to tackle situations mentioned above as it excels in learning from incomplete data[26].
The EM algorithm contains expectation step and maximization step. The former calculates the
likelihood of observations belonging to certain class or cluster, based on the current estimation
of the parameters, while the latter returns the best parameter that maximizes the likelihood of
the data belonging to that class or cluster. Using the basic concept of this algorithm, one can
apply this to estimate parameters of a distribution. This thesis proposes to apply the concept
of convergence to find the optimal timestamp to match time series pattern that belongs to the
class of interest.

2.3 Classifiers

2.3.1 Support Vector Machine

Another challenge in recognizing a pattern is that pattern captured may not be easily observable
nor separable between classes, especially working on a small data set. To identify which patterns
are indeed separable or recognizable remains a major challenge. Support Vector Machine is an
ideal methodology in separating classes as it aims to find the boundary between classes by
maximizing the margin between them[13].

Consider a classifier as a linear function \( f(x) \), it can be expressed as:

\[
y = f(x|w, b) = w^T x + b
\]  

(2.1)

where \( w \) is known as the weight vector, and \( b \) is known as the bias.

For binary classification, the input is bounded by \( x \in \mathbb{R}^d \), and the output is bounded by
\( y \in \{-1, 1\} \), representing the outcome of binary classification.

During training, the user is given a set of examples \( x_i \) and their corresponding labels \( y_i \) to
learn a proper set of weights for support vectors.

Optimization over weight is performed by:

\[
C \sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) + \min_{w \in \mathbb{R}^d} ||w||^2
\]

(2.2)

Where the loss function penalizes misclassifications, the regularization term maximizes the
margin. \( C \) is a regularization parameter that controls the degree of penalization, which also
controls the margin between classes. Small \( C \) gives a large margin, and large \( C \) gives a narrow
margin of the class boundary.
Hence, Support Vector Machine can be expressed as:

$$ f(x) = \sum_i a_i y_i (w_i^T x) + b $$

(2.3)

Where $a_i$ represents the learnt zero and non-zero support vectors.

There are several non-linear kernel tricks, including polynomial, and radial basis function (a.k.a. Gaussian kernel), which aim to find a hyperplane in a higher dimension to separate classes. Due to the nature of high-dimensional small data set, non-linear kernels are prone to cause over-fitting in learning[19]. Linear Support Vector Machine classifier, on the other hand, outperforms kernels that are more complex in solving classification problem given a small sample size of data.

Figure 2.1: Support Vector Machine
Figure taken from Support Vector Machines for Classification[13], an illustration of class boundary and margin.
2.3.2 Logistic Regression

The Logistic Regression model is one of the most commonly used technique to quickly identify class differences.

The Logistic Regression classifier can be defined as:

\[ f(x) = \sigma(w^T x + w_0) \quad y \in [0, 1] \quad (2.4) \]

where the sigmoid function is defined as:

\[ \sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.5) \]

For binary classification, the posterior can be modelled as:

\[ p(C = 0|x) = \sigma(w^T x + w_0) = \frac{1}{1 + e^{-w^T x - w_0}} \quad (2.6) \]

\[ p(C = 1|x) = 1 - p(C = 0|x) = \frac{e^{-w^T x - w_0}}{1 + e^{-w^T x - w_0}} \quad (2.7) \]

To learn model weights, one can maximize the conditional likelihood, or minimize the same by taking the logarithm of the conditional likelihood function with its sign reversed. Hence, the loss function could be derived as follow:

\[ l_{\log}(w) = - \sum_{i=1}^{N} t_i \log(1 - p(C = 0|x^i, w)) - \sum_{i=1}^{N} (1 - t_i) \log(p(C = 0|x^i, w)) \quad (2.8) \]

Through gradient descent, as expressed by the following equation, weights can be learnt by iteration till an optimum point is reached.

\[ w_j^{t+1} \leftarrow w_j^t - \lambda \frac{\partial l_{\log}(w)}{\partial w_j} \quad (2.9) \]

The learning rate \( \lambda \) determines the degree of weight adjustment towards the optimal weights. If \( \lambda \) is very large, convergence to the optimal point might not be possible due to overshooting. If \( \lambda \) is too small, many iterations may be required to reach the best values.

2.3.3 Neural Network

Neural Network is a non-linear discriminative classifier, that consists of an input layer, hidden layer, and an output layer.

The hidden units can be defined as:
Chapter 2. Background Methodology

Figure 2.2: Neural Network
An illustration of Fully Connected Neural Network with 4 inputs, 5 hidden layers and 1 output for binary classification.

\[ h_j(x) = \sigma(w_j + \sum_{i=1}^{D} x_i v_{ji}) \]  \hspace{1cm} (2.10)

where \( j \) represents the index of the hidden unit, \( i \) represents the index of input.

The output unit can be defined as:

\[ o_k(x) = \sigma(w_k + \sum_{j=1}^{J} h_j(x) w_{kj}) \]  \hspace{1cm} (2.11)

where \( k \) presents the index of an output unit.

Similar to Logistic Regression, the sigmoid function is a common activation function used, amongst hyperbolic tangent and rectified linear functions.

A forward pass from the input to the output units performs inference, while a backward pass performs learning. To optimize weights in the function, back-propagation computes gradients and updates weights accordingly in the multi-layer network through cycles of forward and
backward passing.

In this thesis, the use of the Neural Network follows the recommendation from the paper *Learning capability and storage capacity of two-hidden-layer feed-forward networks* [21], where the number of neurons in each of the two hidden layers were formulated using the following equations:

\[
L_1 = \sqrt{(m + 2)N + 2} \sqrt{\frac{N}{m + 2}}
\]

(2.12)

\[
L_2 = m \sqrt{\frac{N}{m + 2}}
\]

(2.13)

where \( L_1 \) represents the number of neurons in the first layer, \( L_2 \) represents the number of neurons in the second layer, \( m \) represents the number of output, and \( N \) represents the number of input.

### 2.3.4 Naive Bayes

Naive Bayes classifier is different from the classifiers mentioned above, as it is a generative approach to classify data.

The basis of Naive Bayes classifier begins with Baye’s Rule, which is defined as:

\[
\text{Posterior} = \frac{p(C|x)}{\text{Evidence}} = \frac{p(x|C)p(C)}{p(x)} = \frac{\text{Class Likelihood} \times \text{Prior}}{\text{Evidence}}
\]

(2.14)

whereby the interest is to find the probability of class given feature vector \( x \). The model can be expressed as:

\[
p(C|x_1, ..., x_i) \propto p(C) \prod_{i=1}^{n} p(x_i|C)
\]

(2.15)

Assuming the data are Gaussian distributed with a Bernoulli prior, one can learn the parameters by maximizing the log likelihood as expressed below:

\[
l(\phi, \mu_0, \mu_1, \Sigma) = - \log \prod_{n=1}^{N} p(x^n, C|\phi, \mu_0, \mu_1, \Sigma)
\]

(2.16)

where \( \phi, \mu_0 \) and \( \mu_1 \), and \( \Sigma \) represent the prior, mean of each class, and the covariance respectively. Once the parameters are learnt, one can perform inference by using the following
Chapter 2. Background Methodology

equation:

\[
p(C|x, \phi, \mu_0, \mu_1, \Sigma) = \frac{1}{1 + e^{-w(\phi, \mu_0, \mu_1, \Sigma)^T x}}
\] (2.17)
Chapter 3

Implementation of The Alignment Algorithm and Classifier

3.1 Algorithm Structure and Basis

In this section, the architect and the reasoning of the labelling algorithm, as well as the structural design of the classifier, are elaborated in details.

The purpose of aligning patterns across example is to find the optimal timestamp that can truly distinguish one class from the other. Optimal performance of the classifier relies on the optimal choice of timestamp label. However, precisely choosing these timestamps could be difficult for supervised learning, since we do not know which timestamp is optimal for pattern recognition until we perform classification, and these patterns are often hard to be described and identified. In general, aligning pattern across examples is a difficult and painful task. Proxy relearning technique proposes a general solution to find optimal labels by exploring time series patterns given a reference point across training examples until the point of convergence or mutual similarities are reached, where learning performance of the classifier is maximized.

For the purpose of building a predictive or detective temporal pattern recognition model, training examples with a frame of time series pattern defined by the onset and the offset of an event are expected. In most cases, climax and the ending of the class of interest are much more easily separated from the general norm. However, the beginning of an event such as seizures, earthquake and stock market crashes might not be easily distinguishable. These cases share similar characteristics, where the exact beginning is hard to pinpoint, the climax of the event is well understood. The goal of aligning training examples is to find the earliest point of bifurcation from each example that is indeed separable from the class of disinterest. One can use this propriety to capture abnormal patterns by updating the label in time forward manner through evaluation of the performance of the classifier as feedback.

To prevent complexity, linear time series classification is much preferred to reduce computational time and power. A Divide-and-Conquer classifier learns multiple discretized patterns
from different sensors through time and frequency, while a gating classifier (i.e. decision-making classifier) infers from the result of multi-pattern recognition classifiers through spatial features. Combining information from space and time can allow the classifier to learn a better pattern, thus reducing the false positive rate.

The proposed alignment algorithm focuses on two areas. First, it aims to find the earliest timestamp from each of the training examples that share bifurcation behaviour. Secondly, it aims to eliminate bad training examples that do not help the classifier to learn, which can boost the performance of the classifier. Keep in mind that not all patterns are deemed recognizable or separable, and learning from inappropriately labelled data could deteriorate classifier’s performance. Through these methods, abnormal patterns are observed, and better labels are placed across all examples.

The basis of this algorithm assumes training with a limited amount of examples containing inseparable data will increase false positive rate dramatically. However, doing this might lower true positive rate since inseparable datum does not necessarily mean it was mis-labelled. One solution to prevent learning misrepresentation from a multi-class event is to collect weeded-out examples that were previously determined to be bad training examples and use them to train a different classifier. The newly trained classifier can then be combined with the original classifier for classification. For better representation learning, one can also use the abundance of examples in class of majority and refresh the training set of majority class by using different sets of examples each time the classifier is being trained to avoid over-fitting.

3.2 Learning Mechanics

There are four major types of machine learning concepts. Supervised learning which relies on vast samples of labelled data to build a classification model. Unsupervised learning which understands the structure of unlabelled data for inference. Semi-supervised learning which uses a small set of labelled data and a large amount of unlabelled data to construct a classifier. Reinforcement learning which contains a reward function to maximize goal or good behaviour within the environment without the need of labelled training examples.

3.2.1 Proxy Relearning Definition

The newly proposed proxy relearning is a hybrid form of the concepts mentioned above, but could not be categorized by traditional convention. Proxy relearning mainly draws ideas from supervised learning, unsupervised learning, and reinforcement learning based on:

- Label Approximation: Data are completely labelled by human experts. However, using labels provided to train classifier fails the purpose of the application. Hence, application-specific labels are needed. Using old labels as a reference point and a constraint/reward function, better labels could be obtained through reinforcement learning. This concludes the term "proxy".
• Data Understanding: The underlying attribute difference between classes of data is unclear, but a class assumption is given for the data. Through adjustment of labels and clustering methods, hidden class attribute difference might surface. When an adequate grouping of data is obtained, the corresponding labels are optimized for the application. This idea touches the boundary of unsupervised learning since the goal is to obtain a set of extremely informative labels for the feature-driven application which was not previously provided.

• Relearning: Since new labels are obtained through iterations of classification and adjustment, labels are constantly redefined until convergence. This concludes the term "relearning".

• Application of Labels: Once optimized labels are obtained, a classifier could be built using supervised learning technique.

3.2.2 Proxy Relearning Approach

Different from the popular data-driven approach, the proposed algorithm adopts a feature-driven approach due to the lack of training examples from minority class of data, limited computational power or hardware implementation constraints. This method is more practical since the user can decide what type of feature to use and learn feature-specific patterns. That being said, it does require the users to design different types of feature in order for the algorithm to work. Hence, some basic scientific knowledge of the data must be understood before applying the algorithm.

To solve a problem, there are usually two common ways. One that focuses on application purpose with confinement and the other explores solutions disregarding constraints.

Application approach selects a portion of known space or data and uses that selective knowledge to solve the problem. This is a real-world problem-solving technique as we are limited by certain constraints. A lot of development is focused on solving a few particular cases given limited examples without understanding population behaviour. However, without fully understanding the problem or the sample space, a lot of unknown behaviours usually arises as developers were not expected to handle them.

Understanding approach usually exhausts the developers by trying different methods in order to gain insights. Relatively time-consuming, developers will understand what is possible or not by thoroughly going through different kind of scientific experiments, disregard of application constraints. However, given limitations, methods that were used to understand the data might not be fitted for application purposes.

The learning algorithm is designed to cope with the above situations, given application constraints to understand data. An illustration of the design is displayed in Figure 3.1. Multi-sensor recordings from minority class and majority class are fed in the feature engineering module to compute user-defined features that are relevant to separate both classes. The user should follow the constraints of the application (e.g. the number of sensors, computational
capability) when applying the learning algorithm. The outputs from feature engineering will be fed to a stack of linear Support Vector Machine classifiers. Each classifier learns solely and directly from the corresponding output of the feature engineering module. It aims to learn and explore temporal patterns, class boundary, and reliability of the feature. In the next stage, gating classifier explores the connectivity of the multi-binary outputs from the stack of linear SVM classifiers, which aims to understand classification patterns and relationship between the linear SVM classifiers. Through this design, the algorithm can determine the training examples' quality and their impact on classification.

### 3.2.3 Input and Pattern Definition

This alignment algorithm searches for continuous short-term pattern given information from features and/or sensors within a time frame unless user-defined features contain memory where information from the past is retained beyond the time frame.

Most classification application relies on information from multiple observations or sources at the same time to infer a decision. Theoretically, all feature patterns should have their own ideal time label of transition from the norm to the class of interest. However, to reduce learning complexity, all features in time domain share one label for this algorithm. This means not all sensors would transit from one class to another at the same time, while some sensors lead in transition, others remain in the previous state. This does not pose a learning problem, as long as some sensors exhibit transitional behaviour, the gating classifier would be able to distinguish the pattern based on the votes of the stack of linear SVM classifiers.

A learning problem occurs if there are inconsistencies across the timing of transition between sensors, which becomes difficult for the gating classifier to learn. To solve this scenario, a simple solution is to superimpose these observations into smaller dimensions and treat the sensors from the system as a whole. The user must understand the general nature of these patterns before applying the alignment algorithm.

### 3.2.4 Divide-and-Conquer Matrix Learning

The two-dimensional input matrix consists of a discrete time series of a feature vector. The dimension of feature vector differs from the size of input sensors because feature engineering often increases the size of the feature space. The dimension of the matrix is huge when compared to the limited amount of training examples due to data being acquired at a high sampling rate. Correlation analysis performed in this scenario is often under-sampled and fails to reach insightful observations in a high-dimensional small sample size data set[6][22][38].

Divide-and-Conquer algorithms are good methods for the purpose of searching informative data behaviour in a high-dimensional data set. Instead of training one classifier to handle a huge two-dimensional matrix, one can train multiple classifiers by breaking down the feature vector individually, transforming the two-dimensional matrix into multiple time series feature vectors, thereby reducing the input space for classifier greatly. However, while performing the
Figure 3.1: Algorithm Work Flow

above-mentioned matrix dissection, a risk would be involved in which the covariance within the two-dimensional matrix could be lost. Keep in mind that the covariance observed could be non-conclusive in the first place due to lack of samples. The relationship between features can still be constructed through gating classifier.
3.2.5 Z-Score

Since most of these patterns were measured in different scales and units, learning weights for the model could be ineffective. Unstandardized feature values have an impact on classifier optimization. By using the mean and standardization of the observations, one can apply z-score to standardize the input features. This can greatly reduce the training computation time as all input features are scaled properly and the feature values are in similar range, where class boundary can be easily quantified. However, using z-score instead of raw input might have resolution issues, due to the fact that z-score is a scaled value with the assumption of a Gaussian distribution. This thesis will not compare the classification result of using z-score or not, but acknowledges their impact[8] on the overall performance of pattern alignment.

3.2.6 Choice of Learner

There are different types of classifier one can apply to learn class boundary or class differences. Since Support Vector Machine finds a boundary that maximizes the feature space distance between two classes, it serves perfectly well for the purpose of learning data separability. There are three main types of kernel for Support Vector Machine: linear, polynomial, and Gaussian kernel. Although in normal condition, where data are balanced and training examples are abundant, polynomial and Gaussian kernel would outperform linear kernel due to its ability to craft better boundary. However, the decision boundary is also more prone to over-fitting if the training set is noisy or data is insufficient[19]. Moreover, optimization problem for nonlinear kernels is much more difficult to solve. It would require a lengthy computational time even with a system of high computational power. Due to this phenomenon, the choice of learner in the architect should be linear Support Vector Machine to avoid further complexity in the proxy relearning algorithm.

3.3 Algorithm for Proxy Relearning

This section covers the implementation and algorithm of proxy relearning, which focus on finding time series patterns across examples that precede the event of interest. The label-relearning mechanism is displayed in Algorithm 1 below.

3.3.1 Alignment Algorithm Parameters

Support Vector Machine performs classification based on score and its sign, where a score is a real number with no upper or lower bound. The score is calculated based on the Euclidean distance between input and learnt support vectors. The user can assess the confidence or the assertiveness of the classification based on the score. The further away the score is from zero means a higher certainty that it belongs to a class and score close to zero means maximum uncertainty in an unbiased setting. Due to the architect of the Divide-and-Conquer classifier,
Result: Alignment Convergence
initialization of timestamp using original labels;
perform training and validation operation;
while \( TP < \text{Threshold or } FP > \text{Threshold} \) do
  if \( timestamp < \text{ending of pattern} \) then
    compute class scores (group training and validation set together);
    find worst minority class example;
    if \( \text{confidence} < \text{threshold} \) then
      timestamp adjustment on worst case;
    else
      stop learning;
    end
  else
    stop learning worst case;
  end
obtain new training set from majority class;
perform training and validation operation;
end

Algorithm 1: Proxy Relearning

assessment of each linear SVM classifier’s prediction confidence could be difficult. Instead, the classifier’s output can be transformed into a bounded measurement between -1 and 1 by averaging the binary outputs of the stack of linear SVM classifiers, producing a normalized value of class score (generally denoted as \( S_c \)) for each sample.

Using this property, there are a few parameters to adjust for the confidence of learning. The parameters of confidence defined in the proxy relearning algorithm are the acceptable thresholds for sensitivity and false negative rate, learning margin between classes, the standard deviation of the class score, and worst case training score. These parameters control the trade-off between detection sensitivity, the predictive time horizon of abnormal activities, and the magnitude of false positive rate.

\[
\text{Sensitivity} = \frac{\text{Number of Minority Class Detected On Minority Validation Set}}{\text{Total Minority Validation Set}} \quad (3.1)
\]

\[
\text{False Positive Rate} = \frac{\text{Number of Minority Class Detected On Majority Validation Set}}{\text{Total Majority Validation Set}} \quad (3.2)
\]

\[
\text{Learning Margin } \quad LM = |\sum_{i=1}^{n} S_{m_i} / n - \sum_{i=1}^{N} S_{M_i} / N|, \quad LM \in \{0 \leq LM \leq 2\} \quad (3.3)
\]
Chapter 3. Implementation of The Alignment Algorithm and Classifier

Standard Deviation of Class Score \( SDS_c = \sqrt{\frac{\sum_{i=1}^{P} (S_{c_i} - \bar{S}_c)^2}{P-1}} \) (3.4)

Worst Case Training Score \( S_{m_W} = \min \{ S_{m_1}, \ldots, S_{m_n} \} \), \( S_{m_i} \in \{-1 \leq S_{m_i} \leq 1\} \) (3.5)

While in traditional supervised learning, where training set is used to learn weights, validation set is used to find appropriate parameter, and test set is used to evaluate performance, this methodology uses the validation set to evaluate if the adjusted labels are appropriate for modelling through observation of sensitivity and false positive rate.

Learning margin between classes and standard deviation of class score, where class score is denoted as \( S_m \) for minority class and \( S_M \) for majority class, are thresholds that aim to stabilize the model’s performance. In theory, the maximum margin between classes is 2 since the class label is either 1 or -1 for SVM classification. However, in practice, the user should understand that a harsh condition on learning margin might sacrifice the ability of minority class prediction.

3.3.2 Label Initialization

Given a set of minority class training examples, labels can be arbitrarily placed a few hours or a few minutes before the pattern of interest, depending on the user’s choice of prediction horizon. If the choice of data set provides timestamp labels, the user can also use it as a reference point for initialization.

3.3.3 Determination Phase (Forward Direction)

Determination phase aims to use low complexity method to find the maximum boundary between classes and measures the similarity and difference between training examples. Using Divide-and-Conquer classifier, each dimension or feature in sequential discrete time series is broken down and assigned to train corresponding Support Vector Machine classifier using linear kernel. The temporal feature series is classified and reduced to a binary output as if the data are being compressed. The output from each individual classifier is summed and divided by the number of votes to produce a bounded value between 1 and -1, producing a class score \( S_c \) for each example. A final linear Support Vector Machine classifier is trained on the resulting value as a gating classifier. Learning the importance of each expert is discouraged at this stage, due to the fact that the weight learnt at this stage might not be generalized enough for classification. A balanced set of training example from each class is encouraged to avoid skewing.
3.3.4 Evaluation Phase (Feedback Direction)

Although class score computed in the determination phase might give us an idea of similarities and differences amongst data, it could be misleading due to the nature that Support Vector Machine classifier is prone to search for any separable boundary between classes whether it is able to learn the true representation of the pattern or not. Hence, we cannot conclude the support vectors chosen are indeed generalized, or in other words, capable to represent or behave similarly on a greater scale using unobserved data. By separating minority examples into training and validation sets, we can evaluate the class score and learning margin $LM$ between both classes to determine the confidence of classification. If the margin is narrow, classifier fails to distinguish classes and would yield a bad performing classifier. The class scores of all the available training and validation examples are then computed from a separated model to understand the quality of the training examples.

3.3.5 Adjustment Phase (Reacquire Data)

Using the training example class scores that were computed in the stage of evaluation phase, one can assume the training example with the lowest class score from minority class is more similar or closely related to majority class. The algorithm assumes the poor performance of the classifier is due to bad training example, hence by adjusting the worst training case through iteration in time forward manner, it would allow the patterns in the training set to converge to a point of high similarity. If multiple training examples share the same minimum class score, those examples should be adjusted together to avoid bias. A randomized selection of training and validation set should be performed through each iteration for both majority and minority classes to avoid over-adjustment of labels.

3.4 Alignment Error

Since this algorithm aims to find the earliest possible abnormal pattern across minority class that deviates from majority class, there are a few theoretical constraints to be kept in mind. One of the problems is that the evaluation of abnormal patterns is based on the class score that could not define the characteristics of patterns. Since the class score calculated in determination phase is an approximation of similarities across examples, a harsh condition on the alignment parameters could cause overshooting in the readjustment of timestamp labels.

Another problem is that since this algorithm relies on features defined by user or application, the quality of the alignment is highly dependent on the quality of the features. Hence, poor choice or inappropriate use of features might result in poor alignment.
3.5 Classification Structure

The classifier shares the structure of the proxy relearning architecture excluding the module of learning constraints and label adjustment. The only difference is the methodology in processing the outputs from the stack of linear SVM classifiers for final inference, which will be outlined in Section 6.3. In plain words, the computation of determination phase for label learning is more or less the same as the classification process. Figure 3.2 illustrates the structure of the classifier.

Figure 3.2: Classification Structure
An illustration of the structure of the classifier. Inputs from multiple sensors pass through feature engineering module, and corresponding time series output is classified individually through a stack of linear SVM classifier. The voting results are then processed by the gating classifier for final inference.
Chapter 4

Testing of Methodology: Epilepsy

4.1 Epilepsy

Epilepsy is a disorder of the central nervous system with the characteristics of recurrent seizures\[4\]. Over 50 million people suffer from this neurological disorder and it has a huge impact on the daily life of these patients. Seizure, defined by the International League Against Epilepsy (ILAE), is "a transient occurrence of signs and/or symptoms due to abnormal excessive or synchronous neuronal activity in the brain"\[11\]. In short, a seizure is a general term to describe an abnormal electrographic activity where the patient might have behaviours of clinical or non-clinical symptoms.

Seizures are extremely difficult for the patients to predict\[34\], and often not preventable. The patient may suffer the loss of consciousness or motor control during an episode of seizure\[4\] and as a result, may lose control of the body and the mind. Due to the nature of this disorder, it has a great impact on the daily lives of individuals that suffer from epilepsy. Patients often have difficulty or are unable to perform independent tasks such as driving. A tremendous amount of effort is required for the patient to join the workforce as well. As a result, patients are likely to develop depression. Suicide rates are three times higher\[29\] when comparing patients suffering from epilepsy with the general public.

4.1.1 Definition of Seizures

According to *An Introduction to Epilepsy*\[4\] by Edward B. Bromfield (MD), Jose E. Cavazos (MD, Ph.D.), and Joseph I. Sirven (MD), a seizure can be characterized into two groups: clinical seizure and subclinical seizure. A clinical seizure is a result of an abnormal hypersynchronous discharge of a population of cortical neurons resulting in subjective symptom or objective signs while subclinical seizure appears only on electroencephalogram (EEG).
Clinical seizures can be classified into different subcategories, illustrated in the following list taken from An Introduction to Epilepsy[4]:

1. Partial Seizure (Focal Seizure, begins locally)
   
   (a) Simple Partial Seizure (Consciousness not impaired)
      
      i. with motor symptoms
      ii. with somatosensory or special sensory symptoms
      iii. with autonomic symptoms
      iv. with psychic symptoms
   
   (b) Complex Partial Seizure (Consciousness impaired)
      
      i. Begins as Simple Partial Seizures and progresses to impairment of consciousness
         A. without automatisms
         B. with automatisms
      ii. with impairment of consciousness at onset
         A. without automatisms
         B. with automatisms
      iii. Partial Seizure (Simple or Complex)

2. Generalized Seizure (Bilaterally Symmetric, without localized onset)
   
   (a) Absence Seizure
      
      i. True absence
      ii. Atypical absence
   
   (b) Myoclonic Seizure
   
   (c) Clonic Seizure
   
   (d) Tonic Seizure
   
   (e) Tonic-clonic Seizure
   
   (f) Atonic Seizure

3. Unclassified Seizure

Partial seizure, also known as focal seizure, is divided into simple and complex subgroups. Complex partial seizure differs from simple partial seizure mainly by the state of consciousness. A partial seizure can also progress to generalized seizure, including absence, myoclonic, clonic, tonic, tonic-clonic, and atonic subgroups. In short, a generalized seizure occurs with a widespread abnormal hypersynchronous discharge bilaterally (i.e. both left and right hemispheres of the brain), with the level of consciousness ranging from unimpaired to impaired, and may or may not include involuntary muscle movement.
4.1.2 Definition of Seizures in Thesis

Since the goal is to predict or detect seizure as early as possible and stimulate the brain to prevent seizing of the patient, the binary classification methodology proposed in this thesis only aims to predict clinical seizure. Recordings that contain seizure events defined by EPILEPSIAE are referred to as ictal segments in this thesis while other types of seizure, such as subclinical seizure, would be treated as inter-ictal (between seizures) activities.

4.2 Motivation

Epilepsy affects about one percent of the total world population. Antiepileptic drug works for about 7 out of 10 patients\[16\], with the risk of having side effects such as tiredness, dizziness, and sleepiness. One-third of those patients are drug resistant and limited treatment can be provided for them. Patients can undergo an evaluation to determine if they are deemed for surgical resection. During this period of time, patients are implanted with intracranial electrodes for monitoring and epileptic source localization. Instead of surgical resection, a closed-loop system that stimulates the brain when the likelihood of a seizure event is high might be a treatment for these patients, based on a belief that stimulation close to the onset of a seizure is more effective in terminating the abnormal activity\[2\]. These data fit perfectly to test the method proposed as it aims to find the earliest detectable sign of a seizure, given a highly imbalanced data set where the nature of given labels is not well understood.

While there are a lot of successful seizure detection algorithms with high sensitivity and specificity, these algorithms rarely focus on investigating the full possibility of early detection or prediction of seizures. The proposed methodology emphasizes on learning labels for early seizure anticipation given user-defined features and constructs a classifier with a goal of achieving high sensitivity and extremely low false positive rate. By building a practical model to recognize patient-specific patterns using proxy relearning technique introduced in this thesis, the best timestamp label across each seizure examples could be found. Applying these labels obtained by the algorithm during training can construct a classifier with lower false positives rate and detect a seizure as early as possible. Based on the proxy relearning results, one can determine if the patient’s seizures are truly recurring patterns, meaning that their symptoms are indeed predictable and recognizable. A set of personalized tailored parameters can then be applied to the patient for long-term seizure prevention.

4.3 Past and Related Work

There are numerous papers related to the area of seizure detection and seizure prediction. However, due to the different choice of data sets, there are only a handful of papers that could relate to this thesis.

Using data provided by EPILEPSIAE with a masked patient identifier, an early seizure
A detection algorithm proposed by Cristian Donons et al. [7] tested eight patients and was able to achieve a best mean false positive rate at 0.33/h with a mean sensitivity of 86.27 percent using Random Forest classification on intracranial EEG. Note that the data used for computation may or may not be the same in Cristian Donnons et al.’s paper and this thesis.

Comparisons of ictal events are shown in Figures 4.1 and 4.2, showing the difference in definition of onset within the same database.

These results are often difficult to compare, as the timestamp of onset and offset assigned are ambiguous in the mathematical sense due to lack of standardization of labelling. The difficulty of obtaining labels for prediction is the lack of cross-disciplinary understanding in the fields of medicine, engineering, and computer science. The purpose of marking onset and offset of a seizure in a clinical setting is not to help computer scientist predict or detect a seizure, but to allow clinicians to locate seizure onset zones, where neurosurgeons can decide if the portion of the brain deemed seizure onset zone is suitable for surgical resection [10]. Due to the difference in objectives, clinicians are more focused on labelling the correct channels where a seizure occur, rather than finding the precise moment when a seizure starts. As the margin of error in pinpointing the start of a seizure increases by a matter of seconds to a few seconds, comparing different seizure anticipation algorithms’ detection latency across non-uniform patient data sets is meaningless.

While sensitivity and specificity are good methods to evaluate the performance of seizure prediction or detection algorithm, it remains a lot of questions about these algorithms before they can be implemented on an implantable device. For example, classification rate, which is frequently ignored in the literature, is one of the most important metrics to compare the speed of classification of different algorithms. A classifier that performs classification every second would require a higher threshold in specificity than a classifier that performs classification every 5 seconds to perform the same in terms of the false positive per hour. A classifier with a slower classification rate (e.g. 5 seconds) would have a greater window where classification could not be performed. A seizure could begin during that time window and intervention of seizure could be too late. In another example of non-quantitative assessment such as false positives handler, which aims to lower the rate of classification using a cool down period (i.e. system stops predicting seizures for an amount of time) or output smoothing (i.e. prediction is made after an
Chapter 4. Testing of Methodology: Epilepsy

An invasive electrode recording at GC2 obtained from patient 970. Note the difference between the original definition of seizure onset from Figure 4.1. Also note the similarity between the original onset defined in Figure 4.1 and the algorithm defined onset displayed in this figure.

Figure 4.2: Onset Comparison 2

Validation and test methodology also hold a crucial role in the justification of results. Steps of validation and testing approaches must be carefully outlined for comparison, including the amount of trained examples required to build the model and the quantity of hold-out test samples. Unfortunately, in this domain, information provided is lacking from previous work.

Algorithms used for prediction or detection could be highly complex, where the computational power or run-time requirement to execute the methodology is often disregarded. For certain applications, these algorithms might not be suited for implantable device[28] which aims to help patients suffering from epilepsy.

That being said, electrodes placement and medication factor are often ignored as well. Poor
placement of electrode could affect the quality of the data. A difference in seizure frequency is shown before and after medication withdrawn[1], due to the fact that medication is able to suppress certain neural logical activity. Hence, to compare results from different algorithms, they must be presented with identical data from the same patients.

Very often, seizure prediction methods propose and function on a selection of patient data could not be replicated. Citing from the paper Seizure Prediction: Methods by Paul R. Carney et al.[30], "the richness of the datasets has meant that these techniques have had limited success in predicting seizures. These limitations may in part stem from our lack of understanding about the mechanism leading to seizures. In many cases, the initial success of a particular measure has been difficult to replicate because the first set of trials was the victim of overtraining. Thus far, no measure has been able to reliably and repeatedly predict seizures with a high level of specificity and sensitivity."

In general and according to the paper Seizure Prediction for Therapeutic Devices: A Review[28] by Kais Gadhoumi et al., "as performance requirements may vary between applications, it becomes difficult to define a gold standard for the required seizure prediction performance. Consequently, it is generally difficult to judge or compare the performance of currently published algorithms, assuming their design and statistical validity."

4.4 Data Set

The proposed method was tested on epilepsy data purchased from EPILEPSIAE, a European Epilepsy Database. The data set contains 30 patients’ intracranial electroencephalography recordings. The data set contains a mean of 19.5 seizures and standard deviation of 17.59 seizures per patient. The duration of recordings has a mean of 209.75 hours with a standard deviation of 96.68 hours. The average number of monopolar electrodes implanted on the patient is 74.6 with a standard deviation of 27. The information of seizure onset zone of each seizure was given as a guideline. Since most seizures in the database have a duration from half a minute to a few minutes, this gives a ratio roughly close to 1:1000 of minority class versus majority class. To only learn the pre-ictal or early-ictal pattern excluding the ictal pattern, the ratio can increase to 1:4000. Hence, this imbalance ratio and lack of occurrence of the class of interest are ideal to test the algorithm.

Out of 30 patients from the purchased data set, 13 were selected in this study to test the methodology proposed. The clinical records of the patients that were used in this study are illustrated in Table 4.1. The data were obtained during pre-surgical evaluation to determine if the patient is suitable for surgical resection. In general, these patients suffer from partial (focal) seizures, and on average only 12 seizures were recorded during their stay.
### Data Set Information

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1096</td>
<td>9</td>
<td>22</td>
<td>Complex Partial</td>
<td>1024</td>
<td>25</td>
<td>18</td>
<td>32</td>
</tr>
<tr>
<td>970</td>
<td>19</td>
<td>43</td>
<td>Simple Partial</td>
<td>256</td>
<td>10</td>
<td>44</td>
<td>64</td>
</tr>
<tr>
<td>635</td>
<td>21</td>
<td>11</td>
<td>Secondarily Generalized</td>
<td>1024</td>
<td>29</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>273</td>
<td>7</td>
<td>7</td>
<td>Simple Partial</td>
<td>256</td>
<td>0</td>
<td>14</td>
<td>32</td>
</tr>
<tr>
<td>264</td>
<td>8</td>
<td>1</td>
<td>Simple Partial</td>
<td>512</td>
<td>10</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>862</td>
<td>9</td>
<td>0</td>
<td>Simple Partial</td>
<td>256</td>
<td>0</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>1125</td>
<td>14</td>
<td>22</td>
<td>Complex Partial</td>
<td>1024</td>
<td>14</td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>1077</td>
<td>9</td>
<td>6</td>
<td>Simple Partial</td>
<td>1024</td>
<td>121</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>818</td>
<td>9</td>
<td>3</td>
<td>Simple Partial</td>
<td>1024</td>
<td>0</td>
<td>14</td>
<td>32</td>
</tr>
<tr>
<td>13089</td>
<td>8</td>
<td>10</td>
<td>Complex Partial</td>
<td>1024</td>
<td>74</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>253</td>
<td>7</td>
<td>71</td>
<td>Complex Partial</td>
<td>512</td>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>442</td>
<td>22</td>
<td>3</td>
<td>Simple/Complex Partial</td>
<td>1024</td>
<td>20</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>958</td>
<td>16</td>
<td>317</td>
<td>Complex Partial</td>
<td>1024</td>
<td>0</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>Total</td>
<td>158</td>
<td>516</td>
<td>NA</td>
<td>NA</td>
<td>333</td>
<td>316</td>
<td>272</td>
</tr>
<tr>
<td>Average</td>
<td>11.5</td>
<td>39.7</td>
<td>NA</td>
<td>NA</td>
<td>25.6</td>
<td>24.3</td>
<td>20.9</td>
</tr>
</tbody>
</table>

| Table 4.1: List of Patients

List of patients used in this thesis. On average, patient only has 12 seizures for training and testing set with input dimension of 71. This high dimensional imbalanced data set creates difficulty in predictive modelling.

### 4.5 Feature Engineering

Based on a rodent trial on closed-loop seizure abortion[15], the main choice of feature for seizure anticipation is wavelet phase coherence, also known as Phase Locking Value (PLV), which measures the degree of synchrony between two electrode recordings. Mono-polar recordings are used for computing wavelet phase coherence using wavelet transformation. Energy information is used as a feature for wavelet phase coherence as well.

Use of wavelet transformation is recommended because of its better time-frequency resolution when compared with other methods such as Hilbert transform. However, there is no statistical support that analyzing neuronal synchrony with wavelet transform could yield significant benefits [27]. Hence, theoretically, the choice of methodology for frequency and synchrony analyses should not affect the predictive modelling.

The data are processed in a simulation fashion where data are processed second by second within an observation frame of 10 seconds with an extended horizon needed to balance out the wavelet end event.
4.5.1 Wavelet Transform

The wavelet transform is computed by the convolution of mother wavelet with the input signal. There are multiple choices of mother wavelet to choose from. In this experiment, the choice of Morlet wavelet is used. The wavelet coefficients as a function of time and frequency are defined as [27]:

\[ W_x(\tau, f) = \int_{-\infty}^{\infty} x(u) \Psi_{\tau,f}^*(u) du \]  

where the Morlet wavelet is defined as:

\[ \Psi_{\tau,f}(u) = \sqrt{f} \exp(i2\pi f(u - \tau)) \exp\left(-\frac{(u - \tau)^2}{2\sigma^2}\right) \]

The number of cycles of the wavelet, that controls the frequency resolution of the analysis is controlled by the width of the frequency interval for which phase is measured. It is defined as:

\[ nco = 6f\sigma \]

Five wavelet cycles were chosen for computation in this thesis. Through wavelet transformation, a frequency spectrum of the signal can be obtained, providing energy information from different bands of frequency.

4.5.2 Wavelet Phase Coherence

To measure the wavelet phase coherence between two electrodes, one can calculate the phase difference of two sources by the following formula:

\[ \Delta \phi_{i,j} = \text{angle}\left( \frac{w_i w_j^*}{|w_i||w_j|} \right) \]

where \( i \) and \( j \) are denoted as the two sources, \( * \) is denoted as complex conjugate and \( w \) is denoted as the wavelet transform of the signal.

The difference over time of the phase difference, also known as phase coherence is defined by:

\[ d\phi_{i,j}(t) = |\Delta \phi_{i,j}(t) - \Delta \phi_{i,j}(t - 1)| \]

The phase coherence across the reference channel and all other channels are then summed and divided by the number of possible combinations. The instantaneous local source coherence
can then be computed as:

\[ S_l = \frac{1}{M-1} \sum_{i \neq l}^M d\phi_{i,j} \] (4.6)

where \( M \) is denoted as the number of channels and \( l \) is denoted as the reference channel.

An illustration of wavelet phase coherence is displayed in Figure 4.3.

### 4.5.3 Crafting of Features and Downsampling

An illustration of feature engineering is displayed in Figure 4.4.

Wavelet transform was computed in the frequency range of 2 Hz to 100Hz with a window of ten seconds before the proxy relearning defined onset plus a few seconds of forwarding buffer window to remove the wavelet end effects at the beginning of the signal. Computation is executed every second, hence each subsequent feature tensor is comprised of 90 percent of the previous sample. Averaged over a range of frequency bands, either with a set of selective non-overlapping bandwidth based on patient’s pattern or using delta, theta, alpha, beta and gamma bands as a guideline, a set of frequency-averaged wavelet phase coherence bands was computed based on the sampling rate. The resulting values are once again averaged over a non-overlapping half-second window. Energy information from each electrode was also processed in a similar manner. The resulting feature tensor is transformed into z-scores for scaling and normalization purposes. In conclusion, the features used for classification include patterns in temporal and spatial information using energy and phase synchrony as measurements.

### 4.5.4 Input Dimension

The following equations illustrate the size of the input dimension of the Divide-and-Conquer Classifiers.

Feature Vector Dimension = \( FNB \)  

\[ \text{Time Series Vector Dimension} = \frac{T}{R} \] (4.8)

The number of dimensions is the product of the number of features \( F \), the total number of electrodes used for computing features \( N \) and the total number of frequency band \( B \). The length of the dimension is determined by the time resolution \( R \) and the duration of signal \( T \).
(a) An invasive electrode recording from the right hippo-campus (HR3) of patient 970.

(b) The Wavelet Phase Coherence computed using HR3 as reference point. The colour bar on the right hand side of the figure shows the index of synchrony, with index of 1 representing high instantaneous phase synchrony, while index of 0 representing out of synchrony.

Figure 4.3: Wavelet Phase Coherence Transformation
Input signals are first transformed into wavelet domain, and then phase coherence and energy information are extracted from selective bands. A z-score transformation is applied before downsampling. The resulting feature matrix is then fed to classifier module.
4.6 Input and Feature Elimination

4.6.1 Electrode Selection

The database provides the origin of each seizure, but once again they may not be the optimal choice for seizure detection. Although we can use methods such as Logistics Regression or by selecting the best individual Support Vector Machine classifier to determine the most informative sensor, the high computational intensity is not ideal due to the fact that choice of features increases exponentially with the number of inputs. Pre-screening each patients’ seizure activities and observing each channel individually is a much quicker way to obtain an idea of the patient’s pre-ictal behaviour. A recommendation of the choice of electrodes is by selecting the channels with most dynamic changes.

4.6.2 Dimension Reduction

During the training phase, dimension reduction is possible by removing some features based on the classification result of the training set. If the training set cannot be classified correctly, it is likely those features are not able to learn correct boundary. One can set a threshold of accuracy for each feature and weed out certain features if they do not reach it. However, dimension reduction is not recommended if the nature of the pattern is not well understood. Until true positive and false positive reach a certain threshold, some features or dimensions with lower weight can be dropped for computational efficiency. This means dimension reduction is not recommended for learning but encouraged for classification.
Chapter 5

Learning Performance

5.1 Predictability

Most researchers focus mostly on three areas, the resulting values of sensitivity, specificity, and false positive rate. By definition sensitivity, specificity and false positive rate are expressed as:

\[
\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.1}
\]

\[
\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \tag{5.2}
\]

\[
\text{False Positives Rate} = \frac{\text{False Positives}}{\text{True Negatives} + \text{False Positives}} = 1 - \text{Specificity} \tag{5.3}
\]

While these metrics are essential to evaluating the classifier’s or algorithm’s performance, the concept of predictability is often ignored. Most importantly, researchers should re-evaluate the nature of labels, and challenge the correctness before further work is performed.

Predictability differs from the concept of classification, by focusing on understanding the cause of certainty in order to predict events. While classification focuses on the correctness of inference by probability or feature space separation with the use of absolute ground truth, predictability focuses on understanding the reason of event recurrence by evaluating past examples. Only after understanding the predictability of events, and similarity across examples, then one can apply this knowledge and focus on classification. This two-step process can ensure better true positives and false positives due to the fact that not all abnormal events are indeed similar or recurring.

One may challenge the thought of learning by predictability since the metrics of predictabil-
ity are not easy to define and greatly depend on the user’s definition. However, there is simply no other better way. Learning the norm of the data, and classify outliers as the class of interest might provide an easy and direct implementation to solve this problem, but this implementation sacrifices small dynamic changes which could be used to predict upcoming events. Most importantly, not all outliers are the causes of abnormal activity since the pattern one might try to predict is only a subset of outliers. In other words, classification by abnormality is not a great practice due to the fact that abnormal activities can still occupy a vast feature space, making localization of boundaries difficult. There are different types of abnormal patterns while specific patterns are bounded and localized.

In this section, the predictability of seizures in patients suffering from epilepsy will be discussed.

5.1.1 Predictable and Unpredictable Patterns

For a few patients, ictal activities can be characterized as a recurring event in time and space domain. However, not all ictal events are deemed predictable. For example, patient 635 from the database has 12 seizure recordings and only 7 of the seizure recordings share similar characteristics. Figure 5.1 is an example of how predictable seizure looks like, while Figures 5.2 and 5.3 are examples of unpredictable seizures. Note that the characteristics of these two examples are extremely different, and the learning algorithm deemed the unpredictable pattern more closely to inter-ictal segments. Due to the fact that learning directly from inseparable data would cause higher false positive rate, the learning algorithm first filters out bad training examples and locates the optimal timestamp for learning to ensure precise learning.

5.1.2 Inconsistent Patterns

In the case where patient’s epileptic behavior is inconsistent, multi-dimension learning fails to acquire patient-specific symptoms, either due to a change in seizure origin, a shift in the ictal frequency range or patterns behave differently. An example of this exhibits in patient 590, given in Figures 5.4 and 5.5. In Figure 5.4, a seizure originated from the HL, BLC and BLB region, and a different seizure, in Figure 5.5, originated from BRA and HR region.

Due to the nature of binary classification, if certain patterns occur only once or twice, a classification model cannot possibly be built due to the lack of class representation in feature space. A better approach to this problem is to use abnormal aggregation, where all information is compressed into lower dimension and classifications are performed on the general index of abnormality patterns. If an adequate amount of multiple class ictal behaviours are provided, users can divide them into different classes using the same learning algorithm, and precisely learn different seizure precursors.

Since this thesis only focuses on binary classification at this stage, patients with inconsistent patterns such as patient 590’s recordings could not work with current algorithm.
A recurring pattern was captured during learning phase, and retained as valid training example for seizure anticipation.
Figure 5.2: PAT635 Seizure 7
A non-recurring pattern labelled as seizure in the database. It was deemed as an invalid training example by the algorithm for seizure anticipation.
Figure 5.3: PAT970 Seizure 17
A non-recurring pattern labelled as seizure in the database. It was deemed as an invalid training example by the algorithm for seizure anticipation.
First occurrence of seizure activity captured. Note the seizure origins were electrodes HL2-10, TLA1, BLB1-2, and BLC1-4.
Twelfth occurrence of seizure activity captured. Note the seizure origins changed from electrodes HL2-10, TLA1, BLB1-2, and BLC1-4 at first occurrence to electrodes HR1-10, TRA2-4, and BRA1-4. Also note that GB8 had huge artifacts due to bad electrode contact.
5.2 Seizure Learning

5.2.1 Recognizable Seizures

To determine which seizures in the data set are deemed recognizable, all available seizures are grouped together as training and validation set. Following the procedures in Section 3.3, the pattern alignment test should stop adjusting labels if all examples in the set are deemed recognizable. If adjustment of labels passed the end of the seizure offset defined by EPILEPSIAE, the example is deemed unrecognizable or user has a choice of selecting different types of features and repeats the test again. Bad training examples should be discarded since they do not fit the application requirement. Keep in mind that parameters of proxy relearning will have an impact on the determination of recognizable and unrecognizable seizures.

5.2.2 Leave-One-Out Cross-Validation

Once enough seizures are obtained, one could perform a leave-one-out cross-validation to ensure these training examples are indeed useful for creating a classification model. An illustration of LOOCV is displayed in Figure 5.6. In this phase, seizures are treated as time-independent events since subsequent seizures were used to classify past seizures.

The reason to use leave-one-out cross-validation is due to the fact that training examples are extremely limited, and k-fold cross-validation is nearly impossible to perform. However, it is useful to examine the labels learned are indeed useful or not. LOOCV is an unbiased cross-validation method since it utilizes N-1 training examples in each validation stage, where N is the total number of training examples. LOOCV iterations N times for hyper-parameter evaluation.

Since there is an abundance of inter-ictal data, LOOCV cannot be applied for each iteration due to computational limitation. Instead, a selected pool of inter-ictal data was randomly chosen as training and validation during each iteration of LOOCV.

Table 5.1 illustrates the learning and cross-validation results. An average of 8.9 seizures out of 11.5 seizures per patient was deemed recognizable. Note that the results of Table 5.1 cannot be used as valid model testing since no test set or unobserved data was used at this stage, and determination of hyper-parameter could not be made without using all available data. This is a test of the reliability of the learning and serves as a methodology for investigating whether data is truly separable. It is not a test for evaluation classification performance. In practice, the user should only apply this step if the classifier is unable to perform generalization on the test set and use this as a guideline to investigate the cause of misclassification.
Figure 5.6: Leave-One-Out Cross-Validation
A cross-validation method to evaluate the labelling and model performance. Each example in the training set is used to validate the performance of the model given a set of hyper-parameters.
Learning and Leave-One-Out Cross-Validation Results

<table>
<thead>
<tr>
<th>Patient</th>
<th>Total Seizures</th>
<th>Recognizable Seizures</th>
<th>Inter-ictal Hour Tested</th>
<th>Inter-ictal Hours Sampled</th>
<th>Sens.</th>
<th>FP/Hr</th>
<th>Best Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1096</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>16</td>
<td>1</td>
<td>4.04E-02</td>
<td>SVM</td>
</tr>
<tr>
<td>970</td>
<td>19</td>
<td>14</td>
<td>11</td>
<td>17</td>
<td>0.74</td>
<td>1.43E-01</td>
<td>NN</td>
</tr>
<tr>
<td>635</td>
<td>12</td>
<td>7</td>
<td>12</td>
<td>18</td>
<td>0.58</td>
<td>1.20E-01</td>
<td>NN</td>
</tr>
<tr>
<td>273</td>
<td>7</td>
<td>7</td>
<td>14</td>
<td>18</td>
<td>1</td>
<td>2.76E-01</td>
<td>NN</td>
</tr>
<tr>
<td>264</td>
<td>8</td>
<td>5</td>
<td>14</td>
<td>24</td>
<td>0.625</td>
<td>6.67E-02</td>
<td>NB</td>
</tr>
<tr>
<td>862</td>
<td>9</td>
<td>4</td>
<td>13</td>
<td>17</td>
<td>0.44</td>
<td>7.69E-02</td>
<td>SVM/LR/NB</td>
</tr>
<tr>
<td>1125</td>
<td>14</td>
<td>14</td>
<td>13</td>
<td>16</td>
<td>1</td>
<td>2.97E-01</td>
<td>NN</td>
</tr>
<tr>
<td>1077</td>
<td>9</td>
<td>8</td>
<td>14</td>
<td>19</td>
<td>0.89</td>
<td>2.68E-01</td>
<td>NN</td>
</tr>
<tr>
<td>818</td>
<td>9</td>
<td>6</td>
<td>14</td>
<td>19</td>
<td>0.67</td>
<td>1.79E-01</td>
<td>NN</td>
</tr>
<tr>
<td>13089</td>
<td>8</td>
<td>6</td>
<td>19</td>
<td>24</td>
<td>0.75</td>
<td>0.00E-00</td>
<td>NN</td>
</tr>
<tr>
<td>253</td>
<td>7</td>
<td>5</td>
<td>13</td>
<td>17</td>
<td>0.71</td>
<td>2.00E-01</td>
<td>SVM</td>
</tr>
<tr>
<td>442</td>
<td>22</td>
<td>21</td>
<td>16</td>
<td>23</td>
<td>0.95</td>
<td>4.47E-01</td>
<td>NN</td>
</tr>
<tr>
<td>958</td>
<td>16</td>
<td>10</td>
<td>17</td>
<td>24</td>
<td>0.625</td>
<td>1.35E-01</td>
<td>NB</td>
</tr>
<tr>
<td>Total</td>
<td>158</td>
<td>125</td>
<td>181</td>
<td>252</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Average</td>
<td>11.5</td>
<td>8.9</td>
<td>13.9</td>
<td>18.9</td>
<td>0.7911</td>
<td>1.74E-01</td>
<td>NN</td>
</tr>
</tbody>
</table>

Table 5.1: Learning and Leave-One-Out Cross-Validation Results
An average of 8.9 seizures out of 11.5 seizures per patient were deemed recognizable. These seizures are retained as training and testing set. An leave-one-out cross-validation was performed on all available seizures to estimate the model performance.
5.3 Label Learning

5.3.1 Training Set Timestamp Labels

After the determination of recognizable seizures, a set of data containing those seizures and ictal segments should be separated into training and testing sets. Depending on the number of seizures provided, the user should retain 50% to 80% of the ictal examples as training sets, with a minimum of three examples, and the remaining as testing sets. Inter-ictal segments can be selected randomly as training and testing sets. To begin label learning, one can initialize the labels with the original timestamps provided by EPILEPSIAE and follow the procedures outlined in Section 3.3. The user should also control the rate of adjustment carefully. A smaller rate of adjustment would require longer computational time while a greater rate of adjustment might cause over-adjustment of labels.

5.3.2 Recommendation of Learning Parameters

A few parameters to adjust learning were mentioned in Section 3.3.1. It includes thresholds for sensitivity, false positive rate, learning margin and worst case training score. A recommendation of sensitivity setting of 0.9 to 1, false positive rate of 0.005 to 0.01, learning margin of 0.85 to 1, and the worst case of training score of 0.75 is used for learning labels in this thesis. Readjustment rate of 0.25 second per iteration is recommended to prevent over-adjustment.

5.3.3 Result of Label Learning

Figures 5.7 and 5.8 illustrate the result of proxy relearning on patient 818, where the newly defined label (vertical line in blue) is chosen to be the optimal point for learning for each example. Note that it differs from the label provided by EPILEPSIAE (vertical line in green). Using the feature information at this timestamp to train classifier increases classifier’s sensitivity, and reduces false positive rate greatly when compared to original labels for training.

5.3.4 Application of Labels

Once labels are obtained, oversampling around the labels can provide more training examples at the risk of over-fitting the classifier. Since the labels obtained by the proposed algorithm is a general approximation of best learning area, the user can use it as a reference to learn pattern up to a few seconds before the timestamp defined by the label to widen the anticipation horizon. Similarly, to lower false positives rate, the user can focus on learning pattern a few seconds after the timestamp label.

To avoid over-fitting, one can increase the training example of the majority class. Hence, the user must experiment with different rates of oversampling with different ratio of the majority and minority class samples in training set. By understanding the effects of these parameters, one can create a generalized classifier.
Figure 5.7: PAT818 Seizure 2
An illustration to compare original onset to the algorithm defined onset which was optimized for application purposes.
Figure 5.8: PAT818 Seizure 7
An illustration to compare original onset to the algorithm defined onset which was optimized for application purposes.
Chapter 6

Classification and Anticipation Horizon

6.1 Definitions

6.1.1 Training and Testing Conditions

To determine if seizures are recognizable or predictable, all available examples were examined by the learning algorithm. The examples that were deemed recognizable were split into training and testing sets. New timestamp labels were obtained for each recognizable training examples following procedures outlined in Chapter 5.

Based on the newly defined labels, minimum of four feature tensors examples with a length of 10 seconds is computed from the ictal class. Hours of inter-ictal segments were chosen randomly, and for each hour 3600 feature tensors with a length of 10 seconds were computed. These feature tensors are later separated into training and testing sets. These feature tensors were computed using MATLAB 2015.

Once the feature tensors were calculated, training and classification are performed following the guidelines in Chapter 3 using python based scikit-learn machine learning packages. A Divide-and-Conquer classifier built with linear Support Vector Machine classifiers and a gating classifier was created. The outputs of the stack of linear SVM classifiers are fed into a gating classifier with the choice of Logistic Regression, Support Vector Machine, Neural Network, and Naive Bayes to observe the difference between each methodology.

A leave-one-out cross-validation was performed to find the optimal parameters for the Divide-and-Conquer classifier. Once certain performance thresholds are reached on the validation set, the corresponding hyper-parameters were retained. Unobserved samples from both classes’ test set were used to evaluate the classifier performance.

6.2 Evaluation Problems and Methods

Unlike typical machine learning analytic, where the model classifies and assigns corresponding class label on a sample per sample basis, prediction model only concerns whether subsequent
samples would evolve into the event of interest. The predictive model could not evaluate true positive rate per time sample, but can only do so by evaluation per event. In order words, as long as one alarm out of all samples within the anticipation window is made before the event of interest, classification of minority event is deemed correct. However, the evaluation method for false positive rate remains the same, as time samples must be evaluated in a continuous manner to anticipate the event of interest to ensure acute prediction.

Methods of evaluation suggested in this thesis are sensitivity, false positive rate, and its seizure anticipation horizon to seizure convulsion.

There are two definitions of sensitivity used in this thesis, learning sensitivity and testing sensitivity. Learning sensitivity measures the anticipation of a seizure made during the frame of an epileptic event as defined by the EPILEPSIAE, whereas testing sensitivity measures the anticipation of a seizure defined by the learning algorithm. In Table 5.1, the sensitivity represents recognizable seizures out of the total seizures provided in the data set. In Tables 6.1 and 6.2, the sensitivity represents the anticipation performance out of the hold-out seizures. To avoid the argument of the validity of seizures in the database, and since all recognizable seizures were classified correctly, the conclusive sensitivity used in this thesis is the learning sensitivity from Table 5.1 at 0.7911, which includes anticipation of unrecognizable seizures such as those shown in Figures 5.2 and 5.3.

\[
\text{Sensitivity} = \frac{\text{Number of Seizures Anticipated per Seizure Events}}{\text{Total Number of Seizure Events}} \quad (6.1)
\]

The definition of the false positive rate used in this thesis is the number of seizures detected outside the frame of epileptic event defined by the EPILEPSIAE per sample, divided by the number of samples of inter-ictal data tested. Since false positive rate and specificity are interchangeable, and false positive per hour is a general metric to evaluate the performance of seizure anticipation classifier, this thesis uses false positive per hour for measurement.

False positive per hour is defined by the amount of incorrectly classified samples divided by the total number of samples being classified. One hour of data contains 3600 samples. Hence, to achieve a false positive per hour of 1, the error rate should not be higher than 2.78E-04.

\[
\text{False Positive Rate} = \frac{\text{Number of Seizures Detected from Inter-ictal Samples}}{\text{Total Number of Test Sample from Inter-ictal Period}} \quad (6.2)
\]

6.3 Choice of Gating Classifier

Gating mechanism in the classifier structure plays an important role. Not only it determines the sensitivity and specificity, the anticipation horizon and response of the classification are characterized as well. There are three ways for the gating classifier to infer from the results of the stack of linear SVM classifiers: a blinded vote, weighted vote, and inference vote. A blinded
vote method disregards the connection between each individual expert and makes decisions based on the average outcome. This type of gating classifier aims to learn a threshold to make a classification, which might not be ideal for complex patterns as it does not understand the importance of output from each linear SVM classifiers and the connection between the outputs. However, by reducing the complexity of recognition, it is less prone to over-fitting. The learning algorithm in Chapters 3 and 5 uses this method to explore labels to avoid over-fitting and biasing. Weighted vote method learns the reliability or the importance of the vote and assigns a higher or lower weight on each individual output of linear SVM classifiers. Some outputs’ weight could be significantly higher than the others, stating the importance of a particular set of features. Features with lower weights could be disregarded or dropped to speed up computational time. However, the model will have a risk of under-performance on test set if there is a lack of training examples, or the training examples are not a good representation of the class. Support Vector Machine and Logistic Regression model belong to this group. Inference vote method infers connections between the outputs of the stack of linear SVM classifiers by understanding their relationship with each other, learning specific sensors or feature behavioural patterns instead of just assigning weights. The Naive Bayes gating classifier learns and makes decisions on the joint probability of the outputs of the stack of linear SVM classifiers, while Neural Network learns the activation function, or connection, between each set of inputs from the outputs of the stack of linear SVM classifiers in subsequent hidden and output layers. Note that both blinded vote method and weighted vote method are discriminative models, while inference vote method could be a generative model or discriminative model.

6.4 Cross-Validation and Testing Results

6.4.1 Optimization Parameters

Amongst hundreds of parameters one can tune, a few noticeable parameters have major effects on the outcome of classifications. The parameters consist of the hyper-parameter for the stack of linear SVM classifiers, the hyper-parameter for the gating classifier, and the training set ratio between minority class and majority class samples. For the sake of reducing computational complexity, one must treat the hyper-parameter for each linear Support Vector Machine within the Divide-and-Conquer classifier the same. A grid search is recommended to search through these parameters to achieve the best result.

6.4.2 Hold-Out Test Average and Best Results

This section outlines the final test of the algorithm, including evaluation of label correctness, parameter stability, and classification capability. An illustration of validation and testing is displayed in Figure 6.1. Depending on the total number of recognizable seizures provided, a set of seizure recordings, in chronological order, were used as training and validation set for LOOCV, while the remaining seizures were left aside as test set. This mimics the real-world
situation, which only unseen future events are being evaluated. A pool of full hour inter-ictal segments was randomly split into training and testing sets. Hence, test data for both inter-ictal and ictal events are completely separated from unsupervised labeling and model parameter selection. Model parameters are determined using LOOCV, and the parameters are applied to the testing model.

Due to the fact that optimization for parameter exists on a non-convex plane, where multiple local optimal points may exist, a grid search was performed in the search for sets of suitable hyper-parameters with acceptable false positive rate and sensitivity. This means that there might be multiple acceptable sets of parameters. Hence, evaluation on the test set using these parameters can allow us to understand the reliability of the classification architect.

Once a certain classification performance is reached during cross-validation, the parameters are retained and applied on the test set. Test results are displayed in Tables 6.1 and 6.2. In Table 6.1, a few sets of parameters were deemed acceptable during the LOOCV stage, where sensitivity is maximized and false positive rate is minimized. These parameters were applied to the classifier to test unobserved ictal and inter-ictal data and the results of these parameters were averaged. Table 6.2 highlights the best performance achieved out of all sets of parameters computed. The main difference occurs at the false positive rate, where on average and best scenario, 0.39 and 0.0519 were achieved respectively.

As we can see from the above results, the false positive per hour is roughly 7.5 times higher.
### Table 6.1: Hold-Out Test Average Results

<table>
<thead>
<tr>
<th>Patient</th>
<th>Recognizable Seizures</th>
<th>Hold-Out Seizures</th>
<th>Inter-ictal Hour Train/Valid</th>
<th>Inter-ictal Test Hours</th>
<th>Sens.</th>
<th>FP/Hr</th>
<th>Best Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1096</td>
<td>9</td>
<td>2</td>
<td>7</td>
<td>9</td>
<td>2/2</td>
<td>0.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>970</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>7</td>
<td>7/7</td>
<td>3.50E-01</td>
<td>NN</td>
</tr>
<tr>
<td>635</td>
<td>16</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8/8</td>
<td>9.29E-01</td>
<td>SVM</td>
</tr>
<tr>
<td>273</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>3/3</td>
<td>2.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>264</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>14</td>
<td>2/2</td>
<td>4.93E-02</td>
<td>SVM</td>
</tr>
<tr>
<td>862</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>1/1</td>
<td>0.00E-00</td>
<td>NN</td>
</tr>
<tr>
<td>1125</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>7/7</td>
<td>0.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>1077</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>11</td>
<td>4/4</td>
<td>6.06E-02</td>
<td>NN</td>
</tr>
<tr>
<td>818</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>11</td>
<td>3/3</td>
<td>0.00E-00</td>
<td>NB</td>
</tr>
<tr>
<td>13089</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>15</td>
<td>3/3</td>
<td>1.15E-00</td>
<td>NN</td>
</tr>
<tr>
<td>253</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>2/2</td>
<td>1.15E-01</td>
<td>SVM</td>
</tr>
<tr>
<td>442</td>
<td>21</td>
<td>12</td>
<td>14</td>
<td>9</td>
<td>12/12</td>
<td>4.44E-01</td>
<td>SVM</td>
</tr>
<tr>
<td>958</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>15</td>
<td>5/5</td>
<td>1.53E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>Total</td>
<td>125</td>
<td>59</td>
<td>125</td>
<td>130</td>
<td>59/59</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Average</td>
<td>9.6</td>
<td>4.5</td>
<td>9.6</td>
<td>10</td>
<td>1</td>
<td>0.39</td>
<td>SVM</td>
</tr>
</tbody>
</table>

Since optimization of hyper-parameters was performed using grid search, multiple sets of parameter might be available. This table illustrates the average performance using different sets of hyper-parameters that passed the threshold during cross-validation.

than best case scenario. Most noticeably, some patients can achieve an average of zero false positive per hour while one achieves 2 false positives per hour. The difference can be related to the fact that different patient has different patterns, and one may be more difficult to learn than the other. Hence, some variance is acceptable. Another reason is that the optimal detection horizon to the seizure convulsions set by the developer during label learning is different from patient to patient, where a learning trade-off against false positive rate must exist for the difference (see Section 6.5.2).

To understand the reliability of the parameters selection, box-plots were used as an evaluation method. Figure 6.2 illustrates the average false positive per hour, the top 5 best false positive rates, and the sensitivity. Both average false positive per hour and the top 5 best false positive rates share similar quartiles, where most of the outcome fall between 0 FP/Hr to 0.5 FP/Hr, confirming the reliability of the algorithm.
### Table 6.2: Hold-Out Test Best Results

Since optimization of hyper-parameters was performed using grid search, multiple sets of parameter might be available. This table illustrates the best performance of all sets of hyper-parameters that passed the threshold during cross-validation.

<table>
<thead>
<tr>
<th>Patient</th>
<th>Recognizable Seizures</th>
<th>Hold-Out Seizures</th>
<th>Interictal Hour Train/Valid</th>
<th>Interictal Test Hours</th>
<th>Sens.</th>
<th>FP/Hr</th>
<th>Best Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1096</td>
<td>9</td>
<td>2</td>
<td>7</td>
<td>9</td>
<td>2/2</td>
<td>0.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>970</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>7</td>
<td>7/7</td>
<td>2.50E-01</td>
<td>NN</td>
</tr>
<tr>
<td>635</td>
<td>16</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8/8</td>
<td>0.00E-00</td>
<td>NB</td>
</tr>
<tr>
<td>273</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>3/3</td>
<td>0.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>264</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>14</td>
<td>2/2</td>
<td>0.00E-00</td>
<td>SVM</td>
</tr>
<tr>
<td>862</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>1/1</td>
<td>0.00E-00</td>
<td>SVM/LR/NB</td>
</tr>
<tr>
<td>1125</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>7/7</td>
<td>0.00E-00</td>
<td>SVM/NB</td>
</tr>
<tr>
<td>1077</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>11</td>
<td>4/4</td>
<td>0.00E-00</td>
<td>SVM/NN/NB</td>
</tr>
<tr>
<td>818</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>11</td>
<td>3/3</td>
<td>0.00E-00</td>
<td>NB</td>
</tr>
<tr>
<td>13089</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>15</td>
<td>3/3</td>
<td>2.67E-01</td>
<td>NB</td>
</tr>
<tr>
<td>253</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>2/2</td>
<td>0.00E-00</td>
<td>SVM/NN/NB</td>
</tr>
<tr>
<td>442</td>
<td>21</td>
<td>12</td>
<td>14</td>
<td>9</td>
<td>12/12</td>
<td>1.11E-01</td>
<td>SVM</td>
</tr>
<tr>
<td>958</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>15</td>
<td>5/5</td>
<td>0.00E-00</td>
<td>SVM/NN/NB</td>
</tr>
<tr>
<td>Total</td>
<td>125</td>
<td>59</td>
<td>125</td>
<td>130</td>
<td>59/59</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Average</td>
<td>9.6</td>
<td>4.5</td>
<td>9.6</td>
<td>10</td>
<td>1</td>
<td>5.19E-02</td>
<td>SVM</td>
</tr>
</tbody>
</table>
Chapter 6. Classification and Anticipation Horizon

(a) Average False Positive Per Hour, refer to Table 6.1

(b) Top 5 Best False Positive Per Hour

(c) Sensitivity, refer to Table 5.1

Figure 6.2: Box-plot of False Positive Rate and Sensitivity

An illustration of patient’s false positive per hour using the proposed classification structure. Figure 6.2(a) shows the best FP/Hr for the 13 patients. It shows that two samples with FP/Hr higher than 1 are outliers. Figure 6.2(b) shows the top 5 FP/Hr from each patient, supporting the resilience of the algorithm proposed in handling false positives. Figure 6.2(c) shows the true sensitivity of detection of seizures including samples from recognizable and unrecognizable seizures from each of the 13 patients.
6.5 Prediction and Detection Responses

6.5.1 Results and Types of Response

Figures 6.4 and 6.6 illustrate the classification results during early ictal stages, indicated by the upward step response from the subplot, while Figures 6.3 and 6.5 represent the whole seizure event, where the green vertical line represents the onset defined by EPILEPSIAE. Only the electrodes used for computation were displayed in Figures 6.4 and 6.6 in order to reduce computation intensity. Hence, only a handful of electrodes were selected for seizure anticipation.

Note that not all responses from the gating classifier behave similarly due to different hyperparameters. Since the Support Vector Machine was able to perform the best on the test set, responses from other gating classifiers can be taken with a grain of salt. In general, Naive Bayes classifier is able to anticipate a seizure convulsion the earliest, followed by Support Vector Machine and Logistic Regression.

Another observation could be made is that the Support Vector Machine and Logistic Regression classifier only focus on learning the pre-convulsion patterns, and do not learn the ictal patterns (Figure 6.4). It was able to craft a boundary in high-dimensional feature space and only contained pre-ictal patterns by learning from proxy relearning defined labels. The precision of learning is concise to the pre-ictal state, excluding patterns during ictal state. This concise learning of pre-ictal state allows the classifier to only learn class boundary of feature patterns that lead to an ictal state. Through limiting the class boundary of the minority class, misclassification could be greatly reduced.
A full display of seizure event from patient 1096. Response of seizure anticipation alarm is demonstrated in Figure 6.4.

### 6.5.2 Learning Trade-off

It is quite obvious to users that in order to reduce the number of false positives, one must sacrifice prediction or detection latency. The learning trade-off can be observed when comparing the detection proximity to seizure convulsion. Patient 442’s (Figure 6.7) learning parameters are able to trigger the alarm almost 10 seconds ahead, while patient 862’s (Figure 6.8) learning parameters are only able to trigger the alarm at the onset of seizure convulsion. When referring to the classification results from the hold-out test average results (Table 6.1), one can observe
Figure 6.4: PAT1096 Seizure 8 Response
Display of first alarm of seizure anticipation from different gating classifiers. An anticipation window of 30 seconds before seizure convulsion is possible.

Patient 442 has a false alarm per hour of 0.444, while patient 862 has a false alarm per hour of 0.

The beauty of this algorithm and architecture is that it allows users to fully explore what is possible, in terms of the prediction/detection horizon or event anticipation, given constraints on the level of sensitivity and false positive rate. One can implement a predictive model or detective model by trading latency versus false positive or computational intensity, giving the users full flexibility in classifier modelling.
Chapter 6. Classification and Anticipation Horizon

Figure 6.5: PAT1096 Seizure 9 Whole Event
A full display of seizure event from patient 1096. Response of seizure anticipation alarm is demonstrated in Figure 6.6.
Figure 6.6: PAT1096 Seizure 9 Response
Display of first alarm of seizure anticipation from different gating classifiers. An anticipation window of 30 seconds before seizure convulsion is possible.
Figure 6.7: PAT442 Seizure 19 Response
A greater seizure anticipation window with a trade-off of higher false positive per hour (0.444 FP/Hr).
Figure 6.8: PAT862 Seizure 6 Response
A lower false positive per hour (0 FP/Hr) with a trade-off of smaller seizure anticipation window.
Chapter 7

Discussion

7.1 Outcome

This thesis provides a solid approach in short-term anticipation of seizure events with a learning mechanism that automatically labels the earliest bifurcation point for precise learning. The false positive per hour achieved at hold-out test scenario on epilepsy data set is at 0.39 on average and at best at 0.052. The proposed methodology highlights the importance of application-based concise learning, where labels are obtained through feature-driven proxy relearning, and used as a reference point to train classifiers. The designed classifier structure can learn information through time, space and feature relationships by dissecting high-dimensional space. With a fully modulated design, users can choose different thresholds of learning parameters for prediction or detection modelling, without reinventing a different architect.

7.2 Future Work

7.2.1 Feature Encoding

The importance of feature encoding, meaning a focus on data compression and informative learning, is one of the concepts this thesis has yet to explore. One of the major problems with the proposed method is its reliance on the proper use of features. If the choice of feature fails to recognize or separate patterns from different classes, the performance of the classifier will be extremely disappointing. Although the Divide-and-Conquer classifier module is able to compress input to a single binary value while retaining the simple representation of the data, it fails to understand a greater depth of the data, as it is limited by user-defined features. A better way to understand data dynamics and the class boundary is needed to fully utilize this method.
7.2.2 Long Term Attention Learning

One of the shortcomings of this technique is that it ignores possible informative patterns that were not captured in the user-defined time window. An alert that gives a longer time horizon might give users or the application more time to respond to certain events. However, this will change the fundamental basis of this thesis, where short-term prediction focuses on learning sensitive dynamic changes before an event through understanding the beginning of bifurcation, long-term attention learning does not only learn continuous patterns but also emphasizes on the past discretized events that contribute to the increase or decrease the event of interest’s likelihood. Although long-term attention learning could provide a warning or forecast the occurrence of certain events, the disadvantage is that it cannot provide the absolute timing when the event will happen. The hybrid version between long-term attention learning and short-term time series pattern recognition could provide the most informative prediction.

7.2.3 Choice of Backbone Classifier

One of the area this thesis did not explore is the choice of Divide-and-Conquer sub-classifiers. Since Support Vector Machine excels in separating classes of data with small attribute differences, it was proposed to use linear Support Vector Machine classifier as the backbone classifier for Divide-and-Conquer classifier. However, other classifiers might work as great or even better, such as Logistic Regression or Random Forest classifier.

7.2.4 One-Class Learning

While this thesis explores the possibility of binary class learning, it is not to discourage the use of One-Class learning, due to its ease of use and simplicity. One-Class learning does excel in a situation when predictability of the event is low. It encourages classification by identifying abnormality or outliers, but not through recognition of specific patterns. Hence, false positive may arise as a result.
Bibliography


