공학박사 학위논문

시계열 데이터 패턴 분석을 위한 종단 심층 학습망 설계 방법론

End-to-End Deep Learning Design Methodologies for Pattern Recognition in Time Series

2019 년 2 월

서울대학교 대학원

전기컴퓨터공학부

황 보 선

End-to-End Deep Learning Design Methodologies for Pattern Recognition in Time Series

시계열 데이터 패턴 분석을 위한 종단 심층 학습망 설계 방법론

지도교수 장병탁

이 논문을 공학박사 학위논문으로 제출함

2019 년 2 월

서울대학교 대학원

전기컴퓨터공학부

황보선

황보선의 공학박사 학위논문을 인준함

2019 년 2 월

위원장	문병로	(인)
부위원장	장병탁	(인)
위원	윤성로	(인)
위원	박철수	(인)
위원	하정우	(인)

This thesis is dedicated to my beloved parents, family, friends and all people who have supported me.

Abstract

Pattern recognition within time series data became an important avenue of research in artificial intelligence following the paradigm shift of the fourth industrial revolution. A number of studies related to this have been conducted over the past few years, and research using deep learning techniques are becoming increasingly popular. Due to the nonstationary, nonlinear and noisy nature of time series data, it is essential to design an appropriate model to extract its significant features for pattern recognition.

This dissertation not only discusses the study of pattern recognition using various hand-crafted feature engineering techniques using physiological time series signals, but also suggests an end-to-end deep learning design methodology without any feature engineering. Time series signal can be classified into signals having periodic and non-periodic characteristics in the time domain. This thesis proposes two end-to-end deep learning design methodologies for pattern recognition of periodic and non-periodic signals.

The first proposed deep learning design methodology is Deep ECGNet. Deep ECGNet offers a design scheme for an end-to-end deep learning model using periodic characteristics of Electrocardiogram (ECG) signals. ECG, recorded from the electrophysiologic patterns of heart muscle during heartbeat, could be a promising candidate to provide a biomarker to estimate event-based stress level. Conventionally, the beat-to-beat alternations, heart rate variability (HRV), from ECG have been utilized to monitor the mental stress status as well as the mortality of cardiac patients. These HRV parameters have the disadvantage of having a 5-minute measurement period. In this thesis, human's stress states were estimated without special hand-crafted feature engineering using only 10-second interval data with the deep learning model. The design methodology of this model incorporates the periodic characteristics of the ECG signal into the model. The main parameters of 1D CNNs and RNNs reflecting the periodic characteristics of ECG were updated corresponding to the stress states. The experimental results proved that the proposed method yielded better performance than those of the existing HRV parameter extraction methods and spectrogram methods.

The second proposed methodology is an automatic end-to-end deep learning design methodology using Bayesian optimization for non-periodic signals. Electroencephalogram (EEG) is elicited from the central nervous system (CNS) to yield genuine emotional states, even at the unconscious level. Due to the low signal-to-noise ratio (SNR) of EEG signals, spectral analysis in frequency domain has been conventionally applied to EEG studies. As a general methodology, EEG signals are filtered into several frequency bands using Fourier or wavelet analyses and these band features are then fed into a classifier. This thesis proposes an end-to-end deep learning automatic design method using optimization techniques without this basic feature engineering. Bayesian optimization is a popular optimization technique for machine learning to optimize model hyperparameters. It is often used in optimization problems to evaluate expensive black box functions. In this thesis, we propose a method to perform whole model hyperparameters and structural optimization by using 1D CNNs and RNNs as basic deep learning models and Bayesian optimization. In this way, this thesis proposes the Deep EEGNet model as a method to discriminate human emotional states from EEG signals. Experimental results proved that the proposed method showed better performance than that of conventional method based on the conventional band power feature method.

In conclusion, this thesis has proposed several methodologies for time series pattern recognition problems from the feature engineering-based conventional methods to the end-to-end deep learning design methodologies with only raw time series signals. Experimental results showed that the proposed methodologies can be effectively applied to pattern recognition problems using time series data.

Keywords: Pattern Recognition, Time Series, End-to-End Deep Learning Networks, Hyper Parameters Optimization, Affective Computing
Student Number: 2014-30319

Contents

Chapter 1 Introduction 1

1.1 Pattern Recognition in Time Series 1

1.2 Major Problems in Conventional Approaches 7

1.3 The Proposed Approach and its Contribution 8

1.4 Thesis Organization 10

Chapter 2 Related Works 12

2.1 Pattern Recognition in Time Series using Conventional Methods 12

2.1.1 Time Domain Features 12

2.1.2 Frequency Domain Features 14

2.1.3 Signal Processing based on Multi-variate Empirical Mode Decomposition (MEMD) 15

2.1.4 Statistical Time Series Model (ARIMA) 18

2.2 Fundamental Deep Learning Algorithms 20

2.2.1 Convolutional Neural Networks (CNNs) 20

2.2.2 Recurrent Neural Networks (RNNs) 22

2.3 Hyper Parameters and Structural Optimization Techniques 24

2.3.1 Grid and Random Search Algorithms 24

2.3.2 Bayesian Optimization 25

2.3.3 Neural Architecture Search 28

2.4 Research Trends related to Time Series Data 29

2.4.1 Generative Model of Raw Audio Waveform 30

Chapter 3 Preliminary Researches: Patten Recognition in Time Series using Various Feature Extraction Methods 31

3.1 Conventional Methods using Time and Frequency Features: Motor Imagery Brain Response Classification 31

3.1.1 Introduction 31

3.1.2 Methods 32

3.1.3 Ensemble Classification Method (Stacking & AdaBoost)..32

3.1.4 Sensitivity Analysis 33

3.1.5 Classification Results 36

3.2 Statistical Feature Extraction Methods: ARIMA Model Based Feature Extraction Methodology 38

3.2.1 Introduction 38

3.2.2 ARIMA Model 38

3.2.3 Signal Processing 39

3.2.4 ARIMA Model Conformance Test 40

3.2.5 Experimental Results 40

3.2.6 Summary 43

3.3 Application on Specific Time Series Data: Human Stress States Recognition using Ultra-Short-Term ECG Spectral Feature 44

3.3.1 Introduction 44

3.3.2 Experiments 45

3.3.3 Classification Methods 49

3.3.4 Experimental Results 49

3.3.5 Summary 56

Chapter 4 Master Framework for Pattern Recognition in Time Series 57

4.1 The Concept of the Proposed Framework for Pattern Recognition in Time Series 57

4.1.1 Optimal Basic Deep Learning Models for the Proposed Framework 57

4.2 Two Categories for Pattern Recognition in Time Series Data 59

4.2.1 The Proposed Deep Learning Framework for Periodic Time Series Signals..59

4.2.2 The Proposed Deep Learning Framework for Non-periodic Time Series Signals...61

4.3 Expanded Models of the Proposed Master Framework for Pattern Recogntion in Time Series....................................... 63

Chapter 5 Deep Learning Model Design Methodology for Periodic Signals using Prior Knowledge: Deep ECGNet.. 65

5.1 Introduction 65

5.2 Materials and Methods 67

5.2.1 Subjects and Data Acquisition 67

5.2.2 Conventional ECG Analysis Methods 72

5.2.3 The Initial Setup of the Deep Learning Architecture..........75

5.2.4 The Deep ECGNet 78

5.3 Experimental Results 83

5.4 Summary 98

Chapter 6 Deep Learning Model Design Methodology for Non-periodic Time Series Signals using Optimization Techniques: Deep EEGNet 100

6.1 Introduction 100

6.2 Materials and Methods 104

6.2.1 Subjects and Data Acquisition 104

6.2.2 Conventional EEG Analysis Methods 106

6.2.3 Basic Deep Learning Units and Optimization Technique 108

6.2.4 Optimization for Deep EEGNet 109

6.2.5 Deep EEGNet Architectures using the EEG Channel Grouping Scheme 111

6.3 Experimental Results 113

6.4 Summary 124

Chapter 7 Concluding Remarks 126

7.1 Summary of Thesis and Contributions 126

7.2 Limitations of the Proposed Methods 128

7.3 Suggestions for Future Works 129

Bibliography 131

초 록 139

List of Tables

Table 3.1 Classification results accuracy comparison..................................37

Table 3.2 Ljung-Box test result..41

Table 3.3 Signal steadiness judgment result (ADF test) 41

Table 3.4 Experimental protocols 45

Table 3.5 Experimental procedures..46

Table 3.6 HRV feature descriptions 48

Table 3.7 Applied classification algorithms...49

Table 3.8 Performance comparisons of classification algorithms on experimental protocols for protocol experiments..........................53

Table 3.9 Performance comparisons of subjects on experimental protocols (Naïve Bayes)..55

Table 4.1 Examples of various hyperparameters and structres for model construction of non-periodic signals 61

Table 5.1 Procedure of the stressor tests. 68

Table 5.2 Experimental protocol descriptions for Case1 70

Table 5.3 The correlation coefficients between short 10-sec HRV parameters and standard 5-minute HRV parameters and their descriptions 73

Table 5.4 The performance comparison between the initial setup empirical deep learning model and conventional methods (HRV-1, HRV-4 and Spectrogram)...78

Table 5.5 Average heart rates (Heart beat duration).....................................80

Table 5.6 The usage of resources and runtimes according to model structures 83

Table 5.7 Conventional method results accuracies-HRV parameters............84

Table 5.8 Conventional method results accuracies-Sectrogram....................85

Table 5.9 Runtime comparison results according to stride sizes..................88

Table 6.1 A list of conventional shallow machine learning algorithms. 107
Table 6.2 Items to be optimized for the improvement of Deep EEGNet performance...110
Table 6.3 Accuracies using conventional methods – Case1........................114
Table 6.4 Accuracies using conventional methods – Case2........................115
Table 6.5 The optimal hyperparameters & model structure - Case1...........116
Table 6.6 Classification accuracies using Deep EEGNet - Case1...............117
Table 6.7 The optimal hyperparameters & model structure - Case2...........118
Table 6.8 Classification accuracies using Deep EEGNet – Case2..............119

List of Figures

Figure 1.1 Conventional pattern recognition flow in time series data............2
Figure 1.2 Pattern recognition flow using deep learning classifiers in time series data...3
Figure 1.3 Pattern recognition flow using transformed features and deep learning classifiers in time series data..4
Figure 1.4 Pattern recognition flow data using deep learning features and various classifiers in time series..5
Figure 1.5 Summary of typical frameworks for pattern recognintion in time series...6
Figure 1.6 Summary of major problems of conventional methods and contribution of the proposed methods..10
Figure 2.1 An example of EMD decomposition for an input signal.............16
Figure 2.2 Convolution and pooling operations of 1D CNN........................21
Figure 2.3 RNNs LSTM (Long-Short Term Memory) structure..................23
Figure 2.4 Illustration of the Bayesian optimization procedure...................27
Figure 2.5 Neural Architecture Search methods...29
Figure 3.1 Stacking ensemble classifier..33
Figure 3.2 Principal channels corresponding to correlation coefficient and mutual information..34
Figure 3.3 Mutual information of each channel: Channel band power was used as a simple feature of EEG signal......................................35
Figure 3.4 Pre-accuracy comparison for channel sensitivity: Using a single SVM classifier...36
Figure 3.5 Example of ARIMA model conformity result (acceleration sensor) 42
Figure 3.6 Example of ARIMA model conformity result (GSR sensor)......43
Figure 3.7 Spectral feature description...48

Figure 3.8 Classification result using HRV parameters 50
Figure 3.9 Classification result using HRV parameters for subjects 50
Figure 3.10 Classification result using 10-sec spectral features 51
Figure 3.11 Classification result using 10-sec spectral features for subjects .51
Figure 3.12 Summary of Algorithm Performance Comparison for protocol experiments 54
Figure 3.13 Protocol Comparison Result (Naïve Bayes) 54
Figure 3.14 Experimental Results on All Subjects...56
Figure 4.1 1D convolution and pooling operations 58
Figure 4.2 The proposed framework concept and the basic deep learning unit 59
Figure 4.3 Examples of appropriate convolution filter length settings.........60
Figure 4.4 Examples of appropriate pooling length settings 60
Figure 4.5 Examples of the proposed frameworks 62
Figure 4.6 Unimodal expansion of the proposed framework 63
Figure 4.7 Multimodal expansion of the proposed framework 64
Figure 4.8 Attention mechanism expansion of the proposed framwork.......64
Figure 5.1 ECG signal acquisitions 68
Figure 5.2 Example of the sequential visual stimuli of CWT across time..69
Figure 5.3 Examples of the visual stimuli to elicit 69
Figure 5.4 Conventional ECG anaysis methods 75
Figure 5.5 Initial setup empirical deep learning architecture 77
Figure 5.6 Raw ECG signal plots corresponding to segment time windows 79
Figure 5.7 The proposed optimal RCNN deep learning architecture: Deep ECGNet 81
Figure 5.8 The various deep learning architectures......................................82
Figure 5.9 The experimental results corresponding to various deep learning

architectures 86
Figure 5.10 The experimental results corresponding to stride sizes 88
Figure 5.11 The pooling length experimental results corresponding to various pooling sizes with fixed 500-point convolution filter length (For example, where ‘800conv800pooling’ means 800-point convolution filter length and 800-point pooling length.) 90
Figure 5.12 The pooling length experimental results corresponding to various pooling sizes with fixed convolution filter length(800-point)...91
Figure 5.13 The convolution filter length experimental results corresponding to various convolution filter sizes with fixed pooling length (800-point) 92
Figure 5.14 The ECG curve with its most common waveforms. Important intervals and points of measurement are depicted. ECG interpretation requres knowledged of these waves and intervals 93
Figure 5.15 The coverages of convolution operation within ECG heart beat duration (0.8 sec) with 0.4sec (a),(b),(c), 0.6 sec(d),(e),(f), 0.8sec (g),(h),(i) width convolution filters, respectively. (Where, the red line is the starting point of ECG P-wave, and the purple box is the coverage region of convolution filter.) 94
Figure 5.16 The captured PQRST waveforms in the convolution operatoin region with 800-pooling..95
Figure 5.17 Visualizations for the hidden layer representation exmples after convolution & activation function layer (ReLU) (Where, the red line is the starting point of ECG P-wave.)................................96
Figure 5.18 The final experimental results compared with conventional ECG analysis methods 97
Figure 5.19 The procedure of the proposed system to generate optimal features 98

Figure 6.1 Experimental prosesses corresponding to the emotional paradigms 105
Figure 6.2 Examples of the emotional video clips to elicit 105
Figure 6.3 Conventional EEG power bands analysis methods 108
Figure 6.4 EEG electrode positions & groups of 7-channel for Case1 and 32-channel for Case2 111
Figure 6.5 Proposed Deep EEGNet architectures 112
Figure 6.6 The comparison results for channel grouping for Case1 and Case 2 121
Figure 6.7 The comparison results for Case 1 . (VS Convenctional methods) 122
Figure 6.8 The comparison results for Case 2 (VS Conventional methods) 123

List of Abbreviations

ADF	Augmented Dicky-Fuller
ANS	Autonomous Nervous System
ARIMA	Autoregressive Integrated Moving Average
BCI	Brain-Computer Interface
BP	Blood Pressure
CNNs	Convolutional Neural Networks
CNS	Central Nervous System
CWT	Color and Word Test
DEAP	A Database for Emotion Analysis using Physiological Signals
DNN	Deep Neural Networks
ECG	Electrocardiogram
EEG	Enceptrophalogram
EMG	Electromyogram
FIR	Finite Impulse Response
GSR	Galvanic Skin Response
HF	High Frequency
HRV	Heart Rate Variability
IIR	Infinite Impulse Response
IMF	Intrinsic Mode Function
LF	Low Frequency
LSTM	Long Short-Term Memory
MEMD	Multivariate Empircal Mode Decomposition
MIST	Montrèal Imaging Stress Test
ML	Machine Learning
MLP	Multi-Layered Perceptrons
NAS	Neural Architecture Search
QDA	Quadratic Discriminant Analysis

RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RMSSD	Square root of the mean average of the sum of the square
RNNs	Recurrent Neural Networks
SAM	The Self-Assessment Manikins
SDNN	Standard Deviation of NN intervals
SNR	Signal to Noise Ratio
SVMs	Support Vector Machines
TF	Total Frequency
TTS	Text to Speech
VLF	Very Low Frequency

Chapter 1

Introduction

1.1 Pattern Recognition in Time Series

Pattern recognition mainly deals with the problem of recognizing a sensed object. In reality, this is actually an implementation problem of artificial intelligence using an engineering approach. Although pattern recognition can be defined in many ways, it is often defined as a field of artificial intelligence that deals with the problem that a computational mechanical device recognizes an object. The recognition task is generally categorized based on how the learning procedure determines the output category. In particular, pattern recognition with time series analysis is an important research theme in the sphere of Machine Learning (ML). A large amount of time series data is generated daily in fields from industry to science and biology. However, most classic data mining algorithms do not perform or scale well on time series data. The inherent structural characteristics of nonstationary, nonlinear and noisy time series data raise the challenge of making classical data mining algorithms inefficient for time series data. As a result, time series analysis has attracted an enormous amount of attention in the past two decades.

Pattern recognition in time series is usually divided into three categories: supervised learning, unsupervised learning and semi-supervised learning. In this thesis, we focus on supervised learning techniques commonly used for time series and novel pattern recognition techniques for physiological time series signals.

For supervised learning, or classification, a functional model is often used to map observed inputs to output categories. A sizeable amount of model construction techniques have been developed for this purpose [Duda00], including decision trees, rule induction, Bayesian networks, memory-based reasoning, Support Vector Machines (SVMs), and neural networks.

Traditionally, for pattern recognition of time series data, appropriate features are extracted using domain knowledge, and then shallow machine learning classifiers such as Nearest Neighbor, Decision Tree, Bayesian Networks, SVM and ensemble classifiers such as Random Forest and AdaBoost are used. (Figure 1.1)

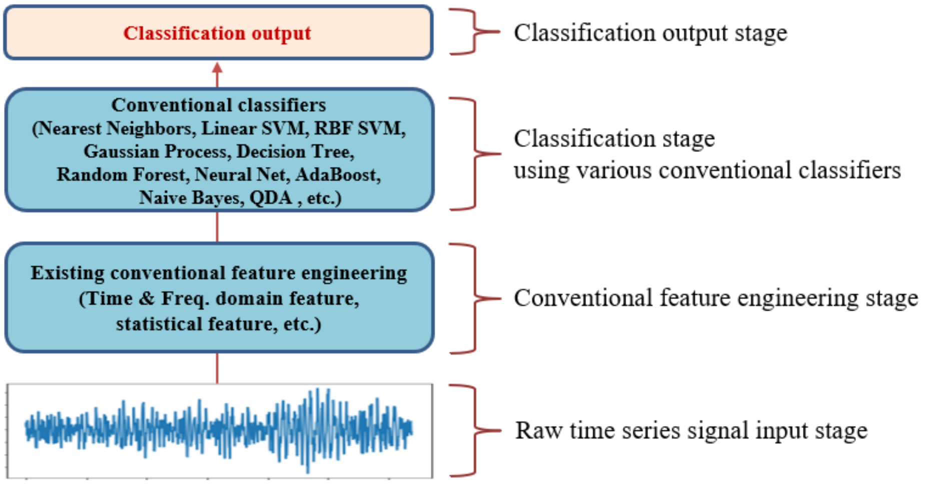

Classification output
Conventional classifiers
(Nearest Neighbors, Linear SVM, RBF SVM,
Gaussian Process, Decision Tree,
Random Forest, Neural Net, AdaBoost,
Naive Bayes, QDA , etc.)
Existing conventional feature engineering
(Time & Freq. domain feature,
statistical feature, etc.)
Classification output stage
Classification stage
using various conventional classifiers
Conventional feature engineering stage
Raw time series signal input stage

Figure 1.1 Conventional pattern recognition flow in time series data

In addition, pattern classification methods using deep learning techniques have been actively developed in recent years. Until recently, there were three major categories that utilize the deep learning model for time series pattern analysis. The first category is to extract existing features and use them as inputs to the deep learning classifiers. Turner et al. extracted the time domain features of multi-channel EEG time series data and performed the task of seizure detection using Deep Belief Network as a classifier [Turner2014]. This is an example of using a deep learning algorithm as a classifier after extracting features using existing feature engineering techniques. (Figure 1.2)

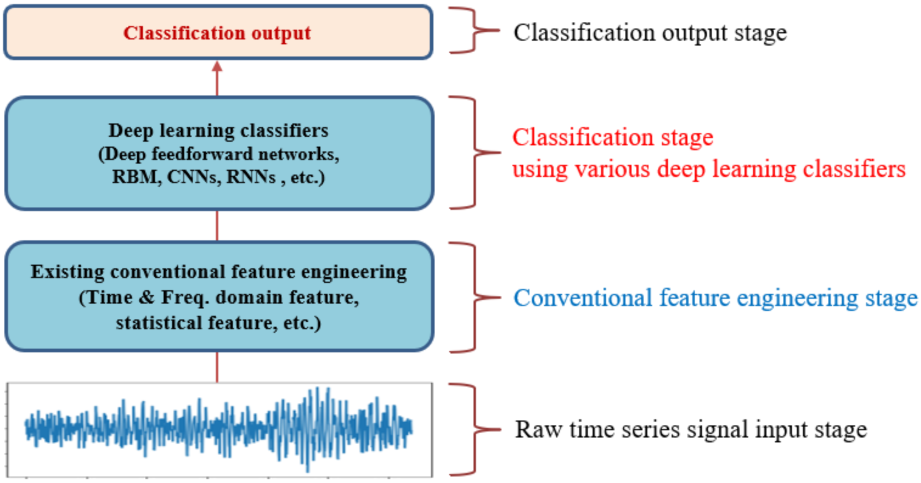

Classification output
Classification output stage
Deep learning classifiers
(Deep feedforward networks,
RBM, CNNs, RNNs , etc.)
Classification stage
using various deep learning classifiers
Existing conventional feature engineering
(Time & Freq. domain feature,
statistical feature, etc.)
Conventional feature engineering stage
Raw time series signal input stage

Figure 1.2 Pattern recognition flow using deep learning classifiers in time series data

The second category is to use the existing deep learning algorithm with good performance as a classifier. This is a method to convert the time series data into the appropriate type data according to the corresponding algorithm. Hatami et al. developed the time series classification method which uses Recurence Plots (RP) to transform time-series into 2D texture images and then takes advantage of the deep CNN classifier [Hatami2018]. This method is an example of data transformation preprocessing to convert 1D times series data into 2D type data to use a 2D CNN classifier which shows excellent performance in image classification. (Figure 1.3)

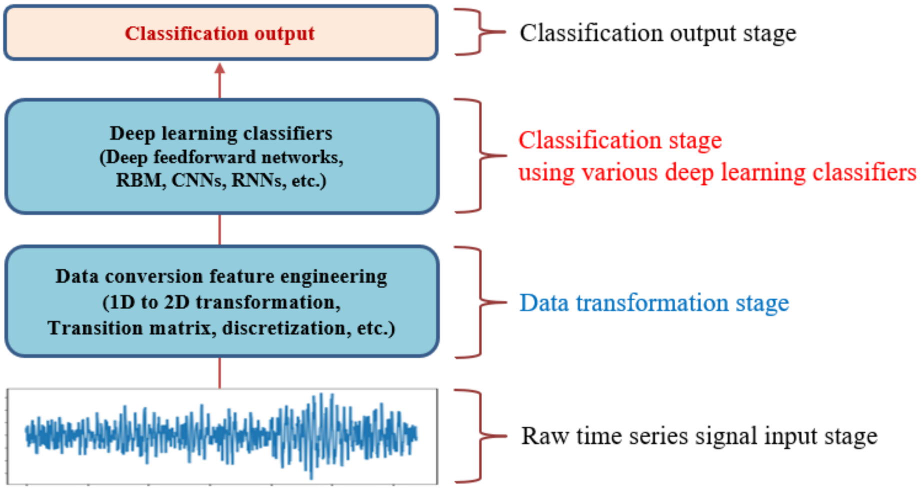

Classification output
Classification output stage
Deep learning classifiers
(Deep feedforward networks,
RBM, CNNs, RNNs, etc.)
Classification stage
using various deep learning classifiers
Data conversion feature engineering
(1D to 2D transformation,
Transition matrix, discretization, etc.)
Data transformation stage
Raw time series signal input stage

Figure 1.3 Pattern recognition flow using transformed features and deep learning classifiers in time series data

The third category is to use deep learning algorithms as a feature extractor. Mehdiyeva et al. used the deep learning model to generate features of time series data. They used LSTM Autoencoder to extract time series features and classify them as normal and abnormal states in the steel industry process using a deep feedforward network as a classifier [Mehdiyev2017].

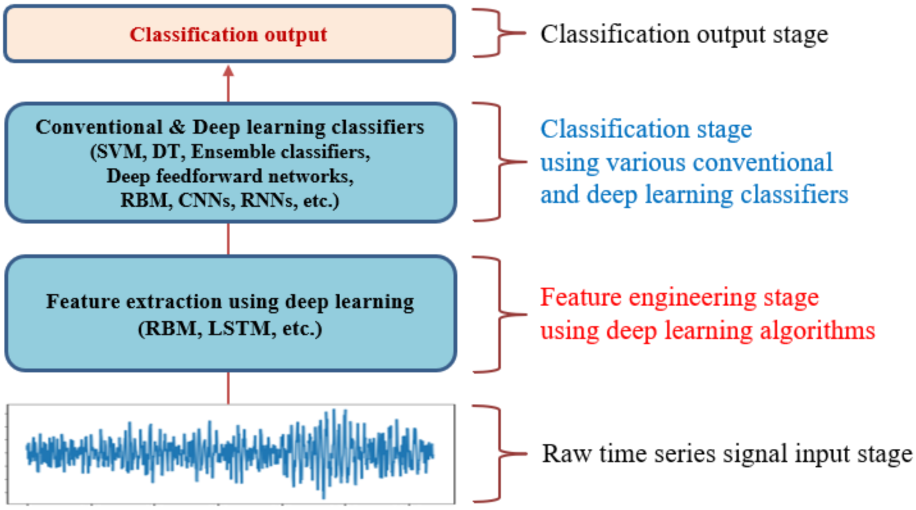

Classification output
Conventional & Deep learning classifiers
(SVM, DT, Ensemble classifiers,
Deep feedforward networks,
RBM, CNNs, RNNs, etc.)
Feature extraction using deep learning
(RBM, LSTM, etc.)
Classification output stage
Classification stage
using various conventional
and deep learning classifiers
Feature engineering stage
using deep learning algorithms
Raw time series signal input stage

Figure 1.4 Pattern recognition flow data using deep learning features and various classifiers in time series

The methods described so far have in common that they must undergo several steps such as feature engineerings to perform pattern recognition on time series data. The main focus of this thesis is to propose the new deep learning design methodology to overcome these common procedures that have to go through cumbersome steps to perform pattern recognition using time series data.

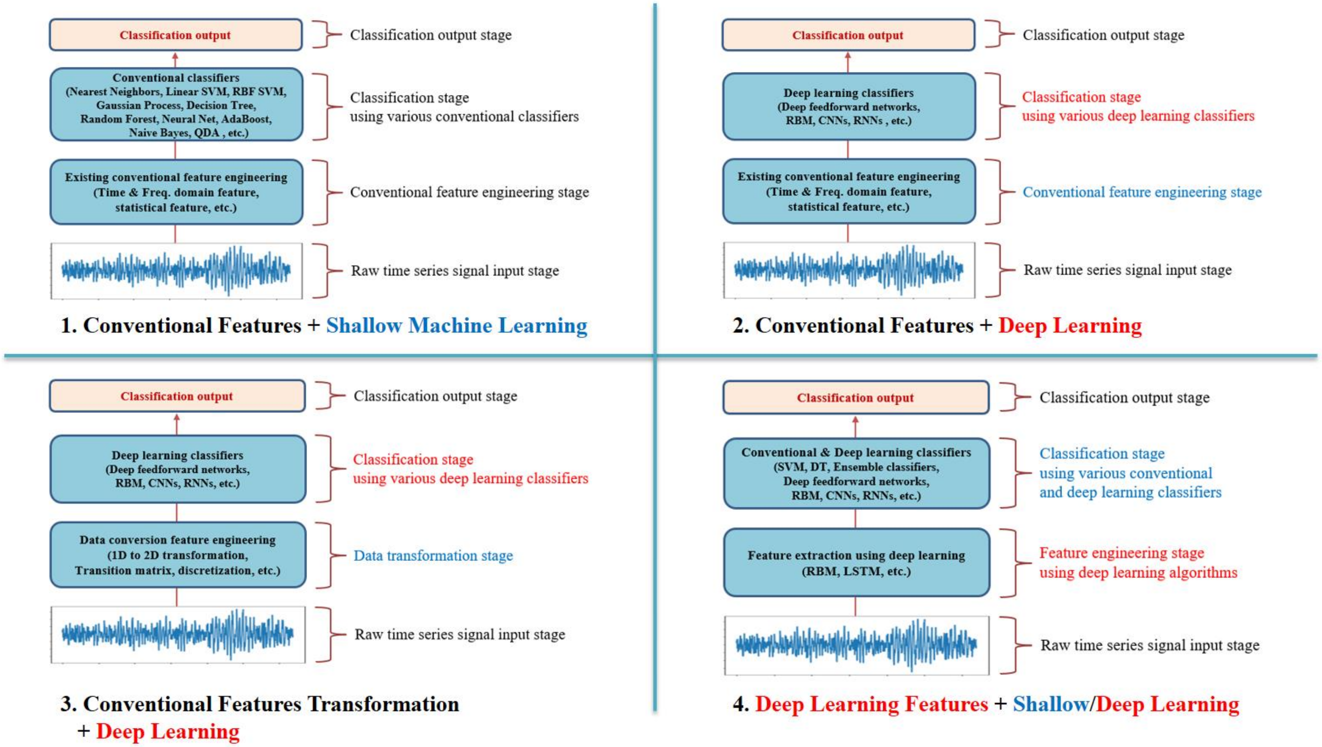

Classification output
Classification output stage
Conventional classifiers (Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, QDA , etc.)
Classification stage using various conventional classifiers
Existing conventional feature engineering (Time & Freq. domain feature, statistical feature, etc.)
Conventional feature engineering stage
Raw time series signal input stage
1. Conventional Features + Shallow Machine Learning
Classification output
Classification output stage
Deep learning classifiers (Deep feedforward networks, RBM, CNNs, RNNs , etc.)
Classification stage using various deep learning classifiers
Existing conventional feature engineering (Time & Freq. domain feature, statistical feature, etc.)
Conventional feature engineering stage
Raw time series signal input stage
2. Conventional Features + Deep Learning
Classification output
Classification output stage
Deep learning classifiers (Deep feedforward networks, RBM, CNNs, RNNs, etc.)
Classification stage using various deep learning classifiers
Data conversion feature engineering (1D to 2D transformation, Transition matrix, discretization, etc.)
Data transformation stage
Raw time series signal input stage
3. Conventional Features Transformation + Deep Learning
Classification output
Classification output stage
Conventional & Deep learning classifiers (SVM, DT, Ensemble classifiers, Deep feedforward networks, RBM, CNNs, RNNs, etc.)
Classification stage using various conventional and deep learning classifiers
Feature extraction using deep learning (RBM, LSTM, etc.)
Feature engineering stage using deep learning algorithms
Raw time series signal input stage
4. Deep Learning Features + Shallow/Deep Learning

Figure 1.5 Summary of typical frameworks for pattern recognition in time series

1.2 Major Problems in Conventional Approaches

This thesis discusses a methodology for designing end-to-end deep learning methodologies using raw signals that does not require special analysis or hand-crafted feature engineering of the data. In other words, the purpose of this study is to propose a deep learning design methodology to overcome several weaknesses of existing pattern recognition methods using time series data. This section discusses the major problems of existing time series analysis techniques, and the major problems of existing methods.

First, shallow machine learning algorithms require a process of extracting special feature vectors for learning. This process requires an additional feature engineering process in the entire design process and, in the case of time series data in particular, requires a significant degree of domain knowledge in order to extract the unique feature points of the signal. For example, suppose you develop a system that analyzes the EEG or ECG signals to determine whether the subject is experiencing health anomalies. In this case, pure machine learning researchers may require a great deal of knowledge about EEG or EEG to use existing algorithms.

Second, the methodology for feature engineering itself requires a lot of knowledge and time. The feature extraction methodology that has been studied so far is also very diverse and vast. For example, in the case of signal pattern recognition, many cases require signal processing theory and a mathematical background, and furthermore, it takes much time to acquire domain knowledge.

Third, the execution time of feature engineering itself can take up a large part of the overall system. For example, ARIMA, a statistical model, and MEMD, an empirical signal processing technique, are based on optimization and iteration, respectively. Therefore, when this process is regarded as the execution time of the feature engineering process, it can act as a bottleneck of the entire design period.

Fourth, the quality of the data is not always assured, even if the feature is extracted by a certain method. This is because, in the existing feature-based methodology, the feature extraction process is not simultaneously considered based on the final target class or label, and the whole process is largely divided into two procedures: feature extraction and classification. Therefore, it is not possible to say that the feature itself extracted by assuming a specific characteristic with a specific knowledge base is the optimal feature for improving the overall performance.

Finally, existing feature-based methods involve a very tedious process of selecting a classification algorithm that matches the extracted features. The performance of the selectable classifier that exhibits the best performance for the extracted feature itself must be accompanied by a performance experiment. Also, it cannot be said that the selected classifier is the optimal classifier that gives the best performance to the given data.

This thesis proposes end-to-end deep learning design methodologies that can perform pattern recognition from a time series signal to a single learning model, from generation to classification of features, as a way to overcome the disadvantages of existing feature engineering based pattern recognition methods.

1.3 The Proposed Approach and its Contribution

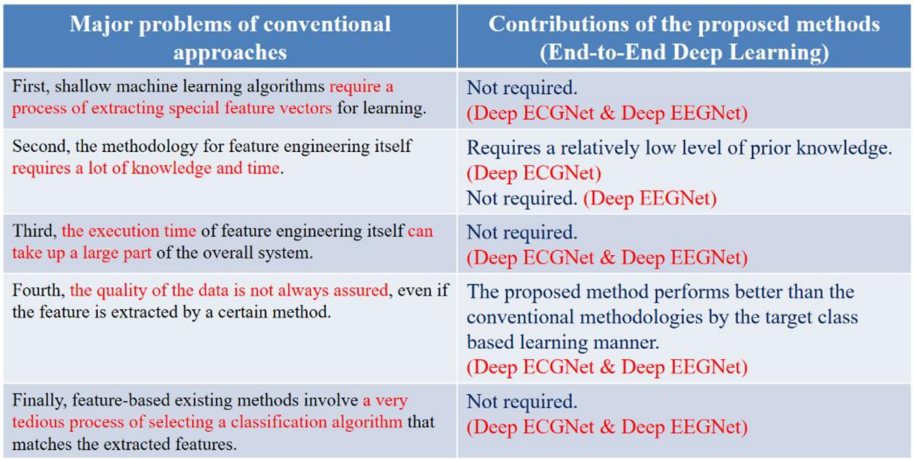
This thesis proposes end-to-end deep learning design methodologies for time series data. In the domain of machine learning, the scope of this thesis is supervised learning and the applied field is pattern recognition for physiological signal which is representative time series data. This thesis proposes two methodologies depending on signal characteristics. Unlike the existing methodologies based on feature engineering, the proposed method does not require a special hand-crafted feature engineering process and it performs better than the existing methodologies by the target class based learning manner.

Many physiological signals have nonstationary, nonlinear, and noisy characteristics, which can be considered as cases with or without periodicity. This paper firstly proposes a deep learning design methodology for a stress state determination system using ECG signals with periodic characteristics of ECG signals.

In addition, a methodology for designing a system for determining the emotional states of a human being using EEG signals, which is a very noisy signal, is proposed in an end-to-end manner from an optimization viewpoint.

The proposed methodologies explain the structure of the model architecture and how to set various hyperparameters efficiently by analyzing the features of the signal itself, without any feature engineering processes using special domain knowledge. We also show that the proposed methodologies are very effective methods for recognizing time series data patterns.

The proposed methodologies are the first example to demonstrate that an efficient end-to-end deep learning model can be designed using only raw signals of ECG and EEG which are very popular time-series data representing various human conditions. In addition, the model constructed through the proposed methodology shows not only effective learnability but also better performance than those of the conventional methodologies based on the existing feature engineering. Fig. 1.6 illustrates the major problems of conventional approaches and the contributions of the proposed methods.



Major problems of conventional approaches	Contributions of the proposed methods (End-to-End Deep Learning)
First, shallow machine learning algorithms require a process of extracting special feature vectors for learning.	Not required. (Deep ECGNet & Deep EEGNet)
Second, the methodology for feature engineering itself requires a lot of knowledge and time.	Requires a relatively low level of prior knowledge. (Deep ECGNet) Not required. (Deep EEGNet)
Third, the execution time of feature engineering itself can take up a large part of the overall system.	Not required. (Deep ECGNet & Deep EEGNet)
Fourth, the quality of the data is not always assured, even if the feature is extracted by a certain method.	The proposed method performs better than the conventional methodologies by the target class based learning manner. (Deep ECGNet & Deep EEGNet)
Finally, feature-based existing methods involve a very tedious process of selecting a classification algorithm that matches the extracted features.	Not required. (Deep ECGNet & Deep EEGNet)

Figure 1.6 Summary of major problems of conventional methods and contributions of the proposed methods

1.4 Thesis Organization

The remained parts of this thesis are organized as follows.

Chapter 2 illustrates the related works of pattern recognition in time series. First, we introduce various hand-craft feature extraction methods related to time series data. Then, we review basic deep learning algorithms which are used in this research. Finally, we illustrate hyper parameter optimization techniques, specifically the Bayesian optimization technique.

Chapter 3 introduces preliminary research. They cover the works of feature engineering-based research that was done before the end-to-end deep learning design methodologies research. The main contents are discussed with respect to various physiological signal pattern recognition methodologies for time, frequency domain features and statistical feature extraction.

Chapter 4 explains the concept of the proposed master framework for pattern recognition in time series.

Chapter 5 and 6 introduce the different types of two end-to-end deep learning design methodologies.

Chapter 5 introduces the Deep ECGNet which was developed through various experiments and analysis using the periodic characteristic of ECG waveforms. We proposed the optimal recurrent and convolutional neural networks architecture, and also the optimal convolution filter length (related to the P, Q, R, S, and T wave durations of ECG) and pooling length (related to the heart beat period) based on the optimization experiments and analysis on the waveform characteristics of ECG signals. The experiments were also conducted with conventional methods using HRV parameters and frequency features as a benchmark test. Experiments were designed according to various experimental protocols to elicit stressful conditions.

Chapter 6 introduces the Deep EEGNet which is a novel deep learning design methodology based on the Bayesian optimization technique. The Deep EEGNet is an end-to-end deep learning framework developed without any feature engineering methods, which demonstrates the ability to recognize human affective states. The Deep EEGNet was constructed using the Bayesian optimization technique to modify the model hyperparameters and structures. Through this optimization process, they were tuned depending on the input data and output class labels. The experiments were also conducted with various conventional shallow machine learning methods to serve as benchmark tests using EEG band power features.

Finally, we summarize contributions, limitation of the thesis and future works in Chapter 7.

Chapter 2

Related Works

2.1 Pattern Recognition in Time Series using Conventional Methods

2.1.1 Time Domain Features

In order to extract features of time series data in time domain, several simple methods have been used, such as extracting statistical numerical values like mean value, standard deviation and so on. In addition to theses methods, JT Turner et al. proposed a method of extracting features of time series data by calculating numerical values about shape characteristics of signals in time domain. Each of the proposed features can be obtained as follows [Turner2014].

- Area: Area under the wave for the given time series. Computed as:

$$\mathrm{A} = \frac{1}{W}\sum_{i=0}^{W-1} x_i \tag{2.1}$$

- Normalized Decay: Chance corrected fraction of data that has a positive or negative derivative. I(x) is a Boolean indicator function, whose value is 1 when true, 0 when false.

$$\mathrm{D} = |\frac{1}{W-1}\sum_{i=0}^{W-2} I(x_{i+1} - x_i < 0) - .5\,|. \tag{2.2}$$

• Line Length: Summation of distance between all consecutive readings.

$$\ell = \sum_{i=1}^{W} -1|x_i - x_{i-1}|. \tag{2.3}$$

• Mean Energy: Mean energy of time interval.

$$\mathrm{E} = \frac{1}{W}\sum_{i=0}^{W-1}|x_i^2|. \tag{2.4}$$

• Average Peak Amplitude: Log base 10 of mean squared amplitude of the K peaks.

$$P_A = \log_{10}\left(\frac{1}{K}\sum_{i=0}^{K-1} x_{k(i)}^2\right). \tag{2.5}$$

• Average Valley Amplitude: Log base 10 of mean squared amplitude of the V valleys

$$V_A = \log_{10}\left(\frac{1}{V}\sum_{i=0}^{V-1} x_{v(i)}^2\right). \tag{2.6}$$

• Normalized Peak Number: Given K peaks, normalized peak number is the number of peaks normalized by the average difference between data readings.

$$N_P = K\left(\frac{1}{W-1}\sum_{i=0}^{W-2}|x_{i+1} - x_i|\right)^{-1} \tag{2.7}$$

• Peak Variation: Variation between peaks and valleys across time (measured in Hz), and electrical signal (measured in mV). In the case where the number of peaks is not equal to the number of valleys (if a time interval begins or ends during an increase or decrease in data, it is not recorded as a peak or valley), then the feature with the least features is used in comparisons between peaks and valleys. The mean (μ(PV)) and standard deviation (σ(P V)) of the indices are given by:

$$\mu(PV) = \frac{1}{K}\sum_{i=0}^{K-1} K_i - V_i$$

$$\sigma(PV) = \sqrt{\frac{1}{K-1}\sum_{i=0}^{K-1}(K_i - V_i - \mu(PV))^2} \tag{2.8}$$

The difference in readings is given by

$$\mu(x_{PV}) = \frac{1}{K}\sum_{i=0}^{K-1} x_{k(i)} - x_{v(i)},$$

$$\sigma(x_{PV}) = \sqrt{\frac{1}{K-1}\sum_{i=0}^{K-1}(x_{k(i)} - x_{v(i)} - \mu(x_{PV}))^2.} \tag{2.9}$$

The peak variation is calculated as

$$P_V = \frac{1}{\sigma(PV)\sigma(x_{PV})} \tag{2.10}$$

- Root Mean Square: The square root of the mean of the data points squared.

$$R_{MS} = \sqrt{\frac{1}{W}\sum_{i=0}^{W-1} x_i^2.} \tag{2.11}$$

JT Turner et al. used these nine features in time domain to perform seizure detection and evaluated the performance of Deep Belief Networks using them.

2.1.2 Frequency Domain Features

The features of time series data are often not exposed externally on the time axis due to their nonstationary, nonlinear and noisy characteristics. In this case, the features are extracted by looking at the data characteristics in the frequency domain. The Fourier transform and wavelet transform have been widely used to analyze frequency components of time series data. The Fourier transform of a function of time is itself a complex-valued function of frequency, whose absolute value represents the amount of that frequency present in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency. The Fourier transform is called the frequency domain representation of the original signal. For example, in the case of the EEG, the frequency band power converted in this way is often used [Ahirwal2012], and in the case of the ECG, the frequency domain parameter is used in the heart rate variability.

Also, we can use the feature by expanding the values of frequency domain according to time with 2D, which is called a spectrogram. This spectrogram can be used as input features for deep learning algorithm such as CNNs [Dennis2011].

2.1.3 Signal Processing based on Multi-Variate Empirical Mode Decomposition (MEMD)

Empirical mode decomposition (EMD) is suitable for non-linear and non-static process analysis because it makes no assumptions about the data [Park2013]. EMD is a fully data-based technique that decomposes a signal into amplitude modulation (AM) and frequency modulation (FM) components, known as an Intrinsic Mode Function (IMF), to reflect the natural oscillations of the signal. The whole procedures of EMD are explained in Algorithm 1.

Algorithm 1. The standard EMD algorithm

```
1. Input signal, $v(t)$.
2. Let $\tilde{v}(t) = v(t)$.
3. Identify all local maxima and minima of $\tilde{v}(t)$.
4. Find a lower “envelope,” $e_l(t)$ that interpolates all local minima.
5. Find an upper “envelope,” $e_u(t)$ that interpolates all local maxima.
6. Calculate the local mean, $\tilde{m}(t) = (e_l(t) + e_u(t))/2$.
7. Subtract the local mean from $\tilde{v}(t), c_i(t) = \tilde{v}(t) - \tilde{m}(t)$ ($i$ is an order of IMF).
8. Let $\tilde{v}(t) = c_i(t)$ and go to step 2; repeat until $c_i(t)$ becomes an IMF.
```

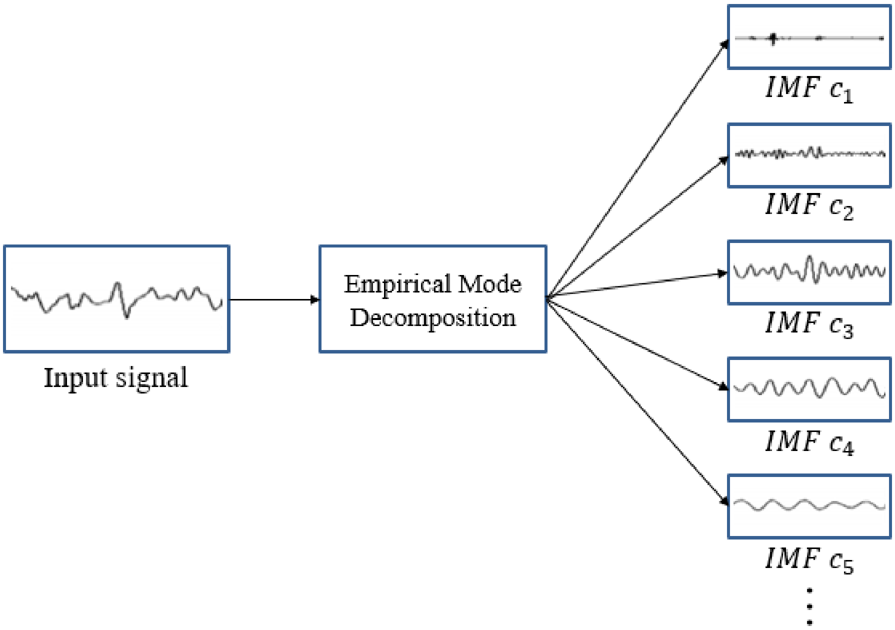


Figure 2.1 An example of EMD decomposition for an input signal.

$c_1(t)$, the first IMF, is subtracted from the original signal, v(t), resulting in a new signal r(t). The same procedure is iteratively conducted until it shows no more oscillations. This is a kind of sifting process controlled by a specific stopping criterion. The original signal v(t) is then

$$\mathrm{v(t)} = \sum_{i=0}^{M} c_i(t) + r(t) \tag{2.12}$$

where $c_i(t)$, i=1,…,M are the IMFs and r(t) is the remaining residue. Figure 2.1 illustrates a concept diagram of EMD decomposition procedure for an input signal. The narrowband nature of the IMFs satisfies the conditions under which the Hilbert transform can be applied [Huang1998].

$$\mathrm{H}\big(C_i(t)\big) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{c_i(t')}{t-t'} dt' \tag{2.13}$$

The localized time-frequency components can be obtained to apply this, where symbol *P* is the Cauchy principal value. The analytic signal is then obtained as

$$\mathrm{v(t)} = \sum_{i=0}^{M} \Big(c_i(t) + jH\big(c_i(t)\big)\Big) = \sum_{i=1}^{M} a_i(t) e^{j\theta_i(t)} \tag{2.14}$$

and is described by its amplitude and phase functions, $a_i(\mathrm{t})$ and $\theta_i(t)$. The instantaneous frequency, $w_i(t) = (d\theta_i(t))/(dt)$ can be produced by differentiating the $\theta_i(t)$ function.

As a natural and generic extension of the standard EMD, Rehman and Mandic [Rehman2010] proposed the multivariate EMD. While standard EMD estimates the local mean using the average of upper and lower envelopes, the MEMD is able to consider n-dimensional signals using by the multiple n-dimensional envelopes, which are generated by projecting the signal along different directions in n-variate spaces. The detailed procedure of MEMD is described in Algorithm 2.

Algorithm 2. The multivariate EMD algorithm (MEMD)

1. Choose a suitable point set for sampling on an (n-1) sphere.

2. Calculate a projection, denoted by $\{p^{\theta_k}(t)\}_{t=1}^{T}$, of the input signal $\{\mathrm{v}(t)\}_{t=1}^{T}$ along the direction vector x^{θ_k}, for all k (the whole set of direction vectors), giving $\{p^{\theta_k}(t)\}_{t=1}^{K}$ as the set of projections.

3. Find the time instants $t_j^{\theta_k}$corresponding to the maxima of the set of projected signals $\{p^{\theta_k}(t)\}_{t=1}^{K}$.

4. Interpolate $\left[t_j^{\theta_k}, \mathrm{v}\left(t_j^{\theta_k}\right)\right]$ to obtain multivariate envelope curves $\{e^{\theta_k}(t)\}_{t=1}^{K}$.

5. For a set of K direction vectors, the mean $\mathrm{m(t)}$of the envelope curves is calculated as $\mathrm{m(t)} = (1)/(K)\sum_{k=1}^{K} e^{\theta_k}(t)$.

6. Extract the "detail" $c_i(t)$ using $c_i(t) = v(t) - m(t)$ (*i* is an order of IMF). If the "detail" $c_i(t)$ fulfills the stoppage criterion for a multivariate IMF, apply the above procedure to $v(t) - c_i(t)$, otherwise apply it to $c_i(t)$.

2.1.4 Statistical Time Series Model (ARIMA)

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model in statistics and particularly in time series analysis. Both models are appropriate for time series data, either to better understand the data or to predict future values in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity. In ARIMA models, an initial differencing step can be applied one or more times to eliminate the non-stationarity of time series data. [Shumway2000]

The AR part of ARIMA means that the variable of interest regresses back to its own delayed variable (i.e., the prior value). The MA part indicates that the regression error is actually a linear combination of error items that occurred multiple times in the past. I (for "integrated") indicates that the data value has been replaced by the difference from the previous value. This differential process may have been performed more than once. The purpose of each of these functions is to fit the model to the data as much as possible.

Non-seasonal ARIMA models are generally denoted ARIMA(*p*,*d*,*q*) where parameters *p*, *d*, and *q* are non-negative integers, *p* is the order of the AR model, *d* is the degree of differentiation and *q* is the order of the MA model. Seasonal ARIMA models are usually denoted ARIMA(*p*,*d*,*q*)(*P*,*D*,*Q*)$_m$, where *m* refers to the number of periods in each season, and the uppercase *P*,*D*,*Q* refer to the autoregressive, differentiation, and moving average terms for the seasonal part of the ARIMA model.

When two out of the three terms are zeros, the model may be referred to by the non-zero parameter, dropping "AR", "I" or "MA" from the acronym describing the model. For example, ARIMA (1,0,0) is AR(1), ARIMA(0,1,0) is I(1), and ARIMA(0,0,1) is MA(1). ARIMA models can be estimated following the Box–Jenkins approach. [Shumway2000]

Given a time series of data X_t where t is an integer index and the X_t are real numbers, an $\mathrm{ARMA}(\mathrm{p}', \mathrm{q})$ model is given by

$$X_t - a_1 X_{t-1} - \cdots - a_{p'} X_{t-p'} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{2.15}$$

or equivalently by

$$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) X_t = \left(1 - \sum_{i-1}^{q} \theta_i L^i\right) \varepsilon_t \tag{2.16}$$

where L is the lag operator, the α_i are the parameters of the autoregressive part of the model, the θ_i are the parameters of the moving average part and the ε_t are error terms. The error terms ε_t are generally assumed to be independent, identically distributed variables sampled from a normal distribution with a zero mean. Assume now that the polynomial$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right)$ has a unit root (a factor $(1 - L)$) of multiplicity d. Then it can be rewritten as:

$$\left(1-\sum_{i-1}^{p'}\alpha_i L^i\right)=\left(1-\sum_{i-1}^{p'-d}\emptyset_i L^i\right)(1-L)^d \qquad (2.17)$$

An ARIMA(p,d,q) process expresses this polynomial factorization property with p=$p'-d$, and is given by:

$$\left(1-\sum_{i-1}^{p}\emptyset_i L^i\right)(1-L)^d X_t=\left(1+\sum_{i-1}^{q}\theta_i L^i\right)\varepsilon_t \qquad (2.18)$$

and thus can be thought as a particular case of an ARMA($p+d$, q) process having the autoregressive polynomial with d unit roots. (For this reason, no ARIMA model with $d > 0$ is wide sense stationary.) The above can be generalized as follows.

$$\left(1-\sum_{i=1}^{p}\emptyset_i L^i\right)(1-L)^d X_t=\delta+\left(1+\sum_{i-1}^{q}\theta_i L^i\right)\varepsilon_t \qquad (2.19)$$

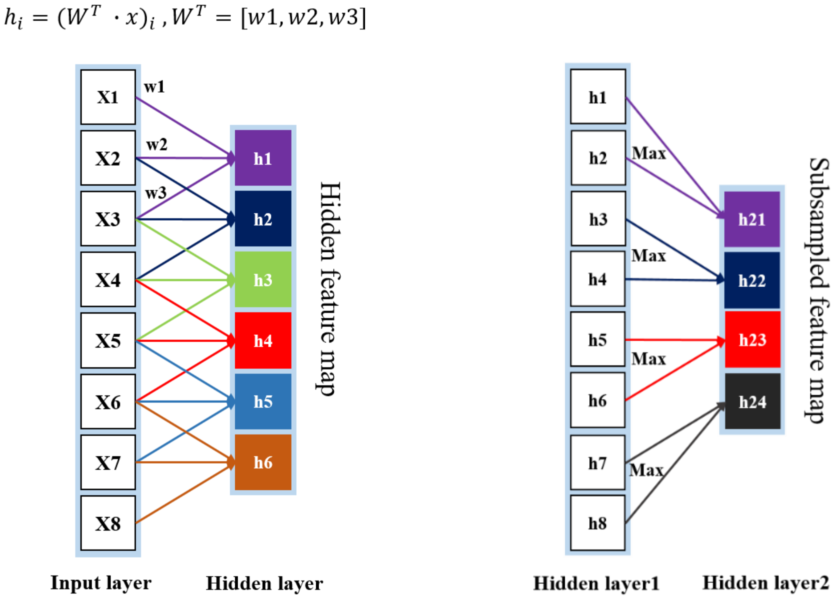
This defines an ARIMA(p,d,q) process with drift $\frac{\delta}{1-\sum\emptyset_i}$. [Shumway2000]

2.2 Fundamental Deep Learning Algorithms

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) were used as a basic unit to make a hidden representation of EEG signals in the whole learning architecture. The CNNs model is one of the most popular deep learning models and has shown outstanding performance in the field of computer vision, which was developed for processing visual information inspired by the human optic nerve structure [LeCun2015][LeCun2010]. CNNs have two main operational layers. The first layer is a convolution layer, while the second layer is a pooling layer providing the advantage of reducing overall model complexity. CNNs have proven to be able to recognize digits with only a few errors, even if placement, scale and rotation of the digits were changed [LeCun2010]. In this study, 1-Dimensional Convolutional Neural Networks (1D CNNs) were used to generate the hidden representations of a univariate time series of EEG signals. Hwang *et al*. have already proven that 1D CNNs are the most effective model for physiological signal analysis [Hwang2018].

The operation procedure and details of the convolution layer are illustrated in Figure 2.2(a), where the 1D filter is implemented in each channel group. During the learning process of CNNs, the convolution filter coefficients are determined by an optimization learning process using error backpropagation according to input signals and output class labels.



(a) Convolution operations

(b) Pooling operations

Figure 2.2 Convolution and pooling operations of 1D CNN

Figure 2.2(b) illustrates the structure of the pooling layer after the convolution operation. Since the convolution operation produces similar outputs across time, subsampling by taking the maximum or average value in a particular window size makes it possible to reduce the feature dimension. In addition, the patterns of data could be well recognized owing to this subsampling, choosing only one value within the pooling operation window, despite data variations such as placement, scale and rotation of data [LeCun2010].

2.2.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are neural network architectures in which hidden nodes are connected to directional edges to form a circulating structure. Long-Short Term Memory (LSTM), a special kind of RNNs, were developed to overcome the vanishing gradient problem of RNNs for long sequential data, and have a structure that adds cell-states to the hidden states of RNNs [Hochreiter1997]. LSTM has shown outstanding performance in sequential information processing such as natural language processing and time series data [Lipton2015].

As shown in Figure 2.3, LSTM models have three layers - forget, input and output. The forget layer determines whether to forget received data from the previous data flow. The input layer is a step to determine whether new information is stored in a cell state. Finally, the output layer decides what to output from the cell state [Hochreiter1997]. In this study, LSTM was used to create features containing sequential information using hidden representation generated from the CNNs layer.

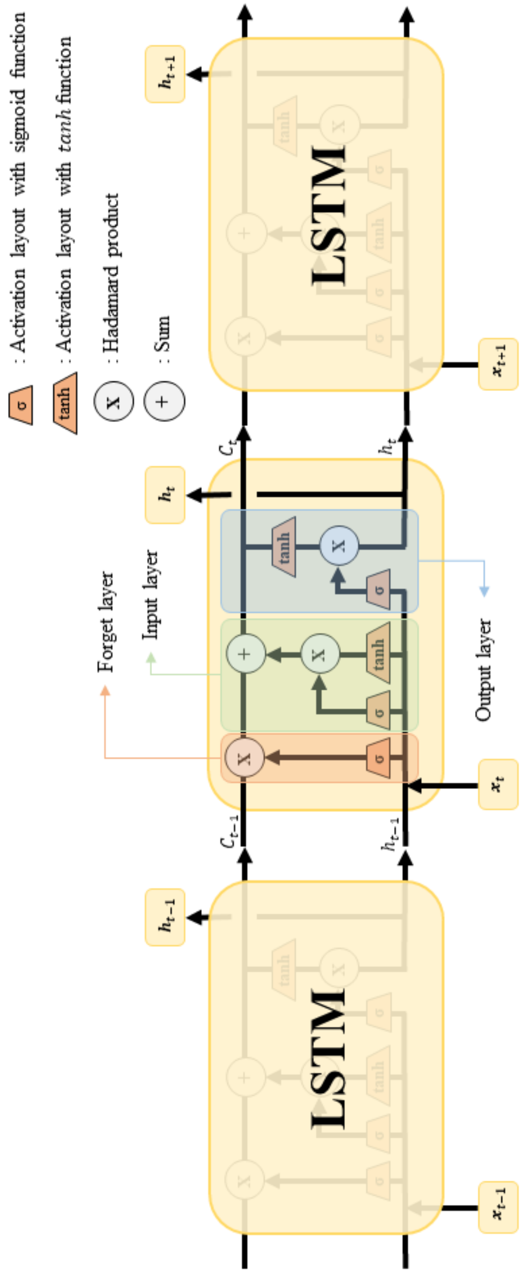


Figure 2.3 RNNs LSTM (Long-Short Term Memory) structure

Furthermore, L1/L2 Regularization, Dropout and Batch Normalization techniques were also applied to improve generalization abilities of the proposed model [Ng2004][Srivastava2014][Ioffe2015].

2.3 Hyper Parameters and Structural Optimization Techniques

When building a deep learning model, decisions of model structures and hyperparameters can be crucial factors in producing the best performance. For example, the number of layers in neural networks, the learning rate, the momentum value, types of activation functions and many other factors should be carefully chosen in order to ensure optimal performance. In the past, choosing those parameters (i.e. ‘tuning’) was the most difficult and time-consuming procedure in designing deep learning architectures.

2.3.1 Grid and Random Search Algorithms

The Grid Search and the Random Search algorithms are typically applied to find the hyperparameters. The Grid Search is a method used to select the parameters yielding the best cross-validation result among all the combinations of the parameters. Although this method can produce the best results in the whole given space, since it is a brute force method, it has the notable drawback of taking too long to search for the optimal solution.

Hwang *et al*. studied an optimal deep learning model for the recognition of human stress states based on the characteristics of ECG time-series patterns using the above mentioned Grid Search approach. This approach was able to find the optimal filter length and pooling length of 1D CNNs [Hwang2018]. Although this method yielded optimal results, this result was possible because of the use of prior knowledge using the characteristics of the ECG signal.

The Random Search is an alternative approach to increase the speed of searching for optimal parameters. Instead of searching all the grids, Random Search observes some parameters chosen by chance, and selects the best parameters among them. Bengio *et al*. demonstrated that high-dimensional hyper-parameters could be optimized using random search with better performance compared to those using Grid Search [Bergstra2012]. However, this method has the disadvantage of leaving the results entirely to chance using no intelligent methods.

2.3.2 Bayesian Optimization

Unlike ECG signals, EEG signals are known to be nonstationary, nonlinear signals with low signal-to-noise ratio that do not exhibit any specific patterns [Park2014]. Therefore, it is difficult to find an optimal model using either the Grid Search or the Random Search method due to the complexity of the large search space.

For these noisy EEG signals, the Bayesian optimization technique could be the most appropriate candidate to find the best hyper-parameters since the Bayesian optimization is known to be suitable for finding the optimal points of a ‘black box’ objective function which requires a notoriously hard, time-consuming and expensive process [Snoek2012].

Bayesian optimization is formulated as below:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}) \tag{2.20}$$

where x is a bounded domain, and *f(*$\mathbf{x}$*)* is a black box function whose shape is not known.

Bayesian optimization assumes that *f(*$\mathbf{x}$*)* has a Gaussian process as a prior, and its posterior is estimated to derive the function *f(*$\mathbf{x}$*)*. Additionally, the next evaluation point of *f(*$\mathbf{x}$*)* can be decided by calculating $\underset{\mathbf{x}}{\operatorname{argmax}}\, \alpha(\mathbf{x}|D)$, where

$\alpha(\mathbf{x}|D)$ is the acquisition function which highlights where the distribution is at its largest. We are then able to select the next evaluation point of *f(*x*)*.

There are two major decisions in the process of Bayesian optimization. One of which is the selection of a Gaussian covariance kernel function for Gaussian prior over functions. The other is to decide an optimal acquisition function allowing the determination of the next evaluation point during the optimization process. Previous research proposed '*Matern5/2 kernel*' as an optimal kernel function and '*Expected Improvement*' as an empirically optimal acquisition function [20]. Gaussian Covariance Kernel (*Matern5/2*) is defined as follows:

$$K_{M52}(\mathbf{x},\mathbf{x}') = \theta_0\left(1+\sqrt{5r^2(\mathbf{x},\mathbf{x}')}+\tfrac{5}{3}r^2(\mathbf{x},\mathbf{x}')\right)\exp\left\{-\sqrt{5r^2(\mathbf{x},\mathbf{x}')}\right\},$$

$$r^2(\mathbf{x},\mathbf{x}') = \textstyle\sum_{d=1}^{D}(x_d - x'_d)^2/\theta_d^2. \qquad (2.21)$$

where θ_0 (the covariance amplitude) and θ_d (the D length scales $\theta_{1:D}$) are the Gaussian process hyperparameters.

The acquisition function (*Expected Improvement*) is defined as follows:

$$\alpha_{EI}(\mathbf{x};\{\mathbf{x}_n,y_n\},\theta) = \sigma(\mathbf{x};\{\mathbf{x}_n,y_n\},\theta)\left(\gamma(\mathbf{x})\Phi\big(\gamma(\mathbf{x})\big)+\mathrm{N}(\gamma(\mathbf{x});0,1)\right),$$

$$\gamma(\mathbf{x}) = \frac{f(\mathbf{x}_{best})-\mu(\mathbf{x};\{\mathbf{x}_n,y_n\},\theta)}{\sigma(\mathbf{x};\{\mathbf{x}_n,y_n\},\theta)},\ \mathbf{x}_{best}=\underset{\mathbf{x}_n}{\operatorname{argmin}} f(\mathbf{x}_n). \qquad (2.22)$$

where $\sigma^2(\mathbf{x};\{\mathbf{x}_n,y_n\},\theta)$ is the predictive variance function and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

Figure 2.4 shows the concept of the Bayesian optimization process. Based on the next evaluation point of the acquisition function, the time for the optimization process could be reduced.

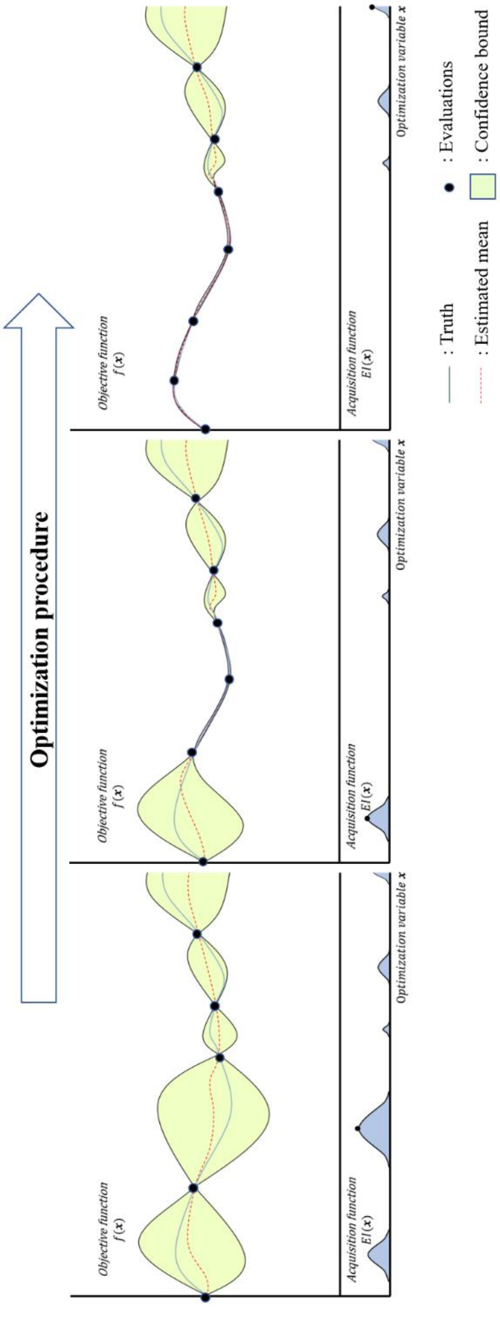


Figure 2.4 Illustration of the Bayesian optimization procedure

2.3.3 Neural Architecture Search

The success of deep learning in perceptual work is largely due to the automation of the feature engineering process. Hierarchical feature extractors are thoroughly learned from data in an end-to-end fashion, an opposed to by manual design. However, this success has increased the demand for architectural engineering, which requires an increasingly complex neural network structure. Neural Architecture Search (NAS), a process for automating architecture discovery, is a natural next step in automating machine learning. NAS can be viewed as a subfield of AutoML and greatly overlaps with hyperparameter optimization and meta-learning. There are three categories for NAS by Elsken et al. which are search space, search strategy, and performance estimation strategy [Elsken2018].

- Search Space: The search space, in principle, defines what architecture can be represented. Consolidating prior knowledge of the attributes that are appropriate for a task allows for the reduction of the search space, thus simplifying the search. However, this also leads to a human bias that can prevent us from finding new architectural building blocks that go beyond current human knowledge.
- Search Strategy: The search strategy explains how to explore the search space. It is desirable to quickly find a good performing architecture, but early convergence should be avoided in the area of suboptimal architectures, including encompassing the classical exploration-exploitation trade-off.

- Performance Estimation Strategy: The goal of the NAS is to find an architecture that achieves high predictive performance for generally invisible data. Performance Estimation refers to the process of evaluating this performance. The simplest option is to perform standard training and validation of the architecture for data, but unfortunately this limits the number of architectures that can be explored because it is computationally expensive. Therefore, many recent studies have focused on developing methods to reduce the cost of these performance estimations.

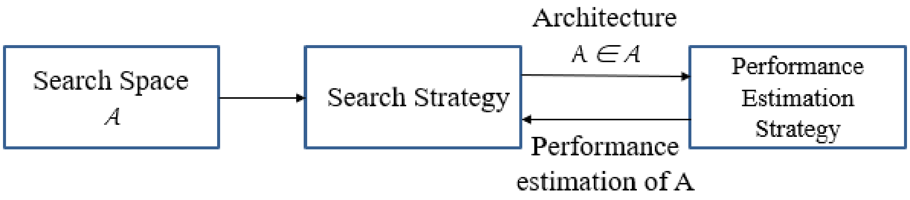


Figure 2.5 Neural Architecture Search methods

2.4 Research Trends related to Time Series Data

Recent research trends in time series data include research on pattern recognition as well as the generation and synthesis of time series data. Text to speech (TTS) is a research area of artificial intelligence that deals with the problem of generating speech waveforms from texts. A study called WaveNet, which was released by Google DeepMind in 2016, has received a great deal of attention [Oord2016].

2.4.1 Generative Model of Raw Audio Waveform

WaveNet is a deep neural network for generating raw audio waveforms [Oord2016]. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones. Oord et al. demonstrated that it can be efficiently trained on data with tens of thousands of samples per second of audio. Applying to text-to-speech, they achieved state-of the-art performance, with human listeners rating it as significantly more natural sounding than the best parametric and concatenative systems for both English and Chinese.

One WaveNet can capture the characteristics of many different speakers with equal fidelity and switch between them by adjusting speaker identity. When modeling music, you can see that it generates new, realistic music fragments. They also show that it can be used as a discriminative model and return promising results for phoneme recognition [Oord2016].

Chapter 3

Preliminary Researches:

Pattern Recognition in Time Series using Various Feature Extraction Methods

3.1 Conventional Methods using Time and Frequency Features: Motor Imagery Brain Response Classification

3.1.1 Introduction

Motor imagery brain response during motor planning is one of the most popular paradigms to implement brain-computer interface (BCI) system. The brain response during motor imagery for BCI is commonly obtained using electroencephalogram (EEG) owing to its noninvasive and convenient way to record [Park2013]. The information of interest in EEG is located in well-defined frequency bands, and a number of standard algorithms have been used for feature extraction and pattern classification. We show that factor analysis and ensemble classification methods can be applied to enhance the pattern classification rate to separate two different motor imagery tasks compared to conventional classification methods. Comparative study on both synthetic benchmark examples and well established BCI motor imagery dataset supports the analysis. (Pysiobank motor/mental imagery database)

3.1.2 Methods

Formally, the mutual information of two discrete random variables *X* and *Y* can be defined as [Vergara2014]:

$$\mathrm{I(X;Y)} = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{3.1}$$

where *p*(*x*, *y*) is the joint probability distribution function of X and Y, and *p*(*x*) and *p*(*y*) are the marginal probability distribution functions of *X* and *Y* respectively. In the case of continuous random variables, the summation is replaced by a definite double integral:

$$\mathrm{I(X;Y)} = \int_Y\int_X p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\, dy \tag{3.2}$$

where *p*(*x*, *y*) is now the joint probability density function of *X* and *Y*, and *p*(*x*) and *p*(*y*) are the marginal probability density functions of *X* and *Y* respectively. It is possible to obtain the non-linear relationship between input data and output data using information visualization. Here, the mutual information was obtained by discrete random variable case using 64-ch EEG quantized band power level counting.

3.1.3 Ensemble classification method (Stacking & AdaBoost)

Stacking (sometimes called *stacked generalization*) involves training a learning algorithm to combine the predictions of several other learning algorithms. Here, SVM, MLP and Naïve Bayes classifiers were adopted as sub-classifier and logistic regression function was adopted as a meta classifier [Hall2009].

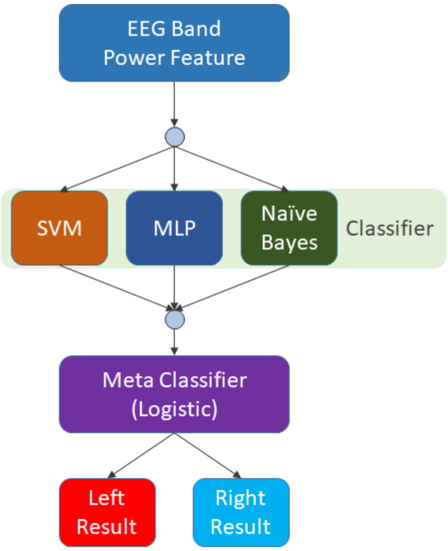


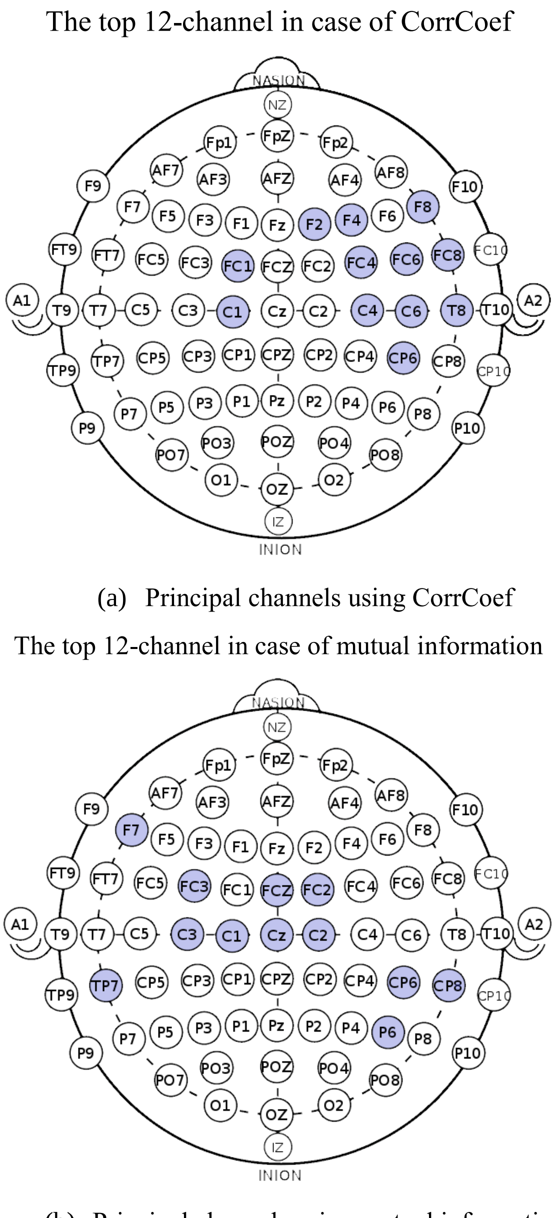
Figure 3.1 Stacking ensemble classifier

AdaBoost (adaptive boosting) is an ensemble learning algorithm that can be used for classification or regression. AdaBoost creates the strong learner (a classifier that is well-correlated to the true classifier) by iteratively adding weak learners (a classifier that is only slightly correlated to the true classifier) [Hall2009].

3.1.4 Sensitivity analysis

Sensitivity analysis for feature selection was performed for effective use of extracted features in the motor imagery brain response. 10 subjects and 64 channel EEG data are used, and each has 45 instances. The feature used was a band power feature for each channel. Feature selection criteria were correlation coefficient and mutual information.

According to the criterion, each validity was compared by looking at the accuracy of the SVM classification algorithm, and it was confirmed that the mutual information showed better performance than the correlation coefficient in the motor imagery brain response task.


NASION
NZ
Fp1 FpZ Fp2
AF7 AF3 AFZ AF4 AF8
F9 F7 F5 F3 F1 Fz F2 F4 F6 F8 F10
FT9 FT7 FC5 FC3 FC1 FCZ FC2 FC4 FC6 FC8 FC10
A1 T9 T7 C5 C3 C1 Cz C2 C4 C6 T8 T10 A2
TP9 TP7 CP5 CP3 CP1 CPZ CP2 CP4 CP6 CP8 CP10
P9 P7 P5 P3 P1 Pz P2 P4 P6 P8 P10
PO7 PO3 POZ PO4 PO8
O1 OZ O2
IZ
INION
NASION
NZ
Fp1 FpZ Fp2
AF7 AF3 AFZ AF4 AF8
F9 F7 F5 F3 F1 Fz F2 F4 F6 F8 F10
FT9 FT7 FC5 FC3 FC1 FCZ FC2 FC4 FC6 FC8 FC10
A1 T9 T7 C5 C3 C1 Cz C2 C4 C6 T8 T10 A2
TP9 TP7 CP5 CP3 CP1 CPZ CP2 CP4 CP6 CP8 CP10
P9 P7 P5 P3 P1 Pz P2 P4 P6 P8 P10
PO7 PO3 POZ PO4 PO8
O1 OZ O2
IZ
INION

The top 12-channel in case of CorrCoef

(a) Principal channels using CorrCoef

The top 12-channel in case of mutual information

(b) Principal channels using mutual information

Figure 3.2 Principal channels corresponding to correlation coefficient and mutual information

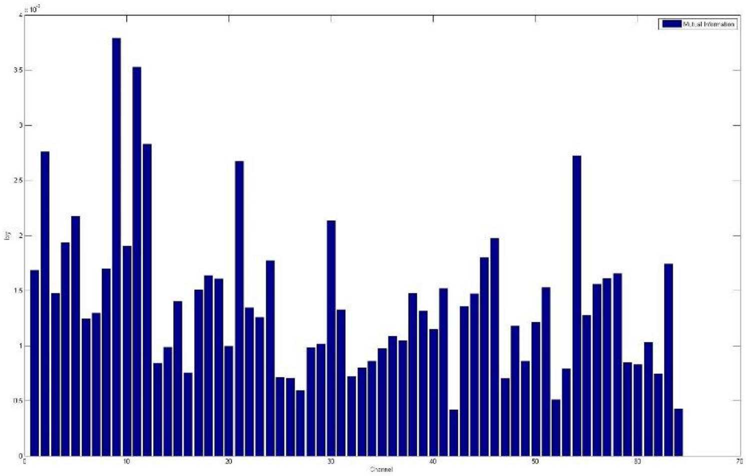
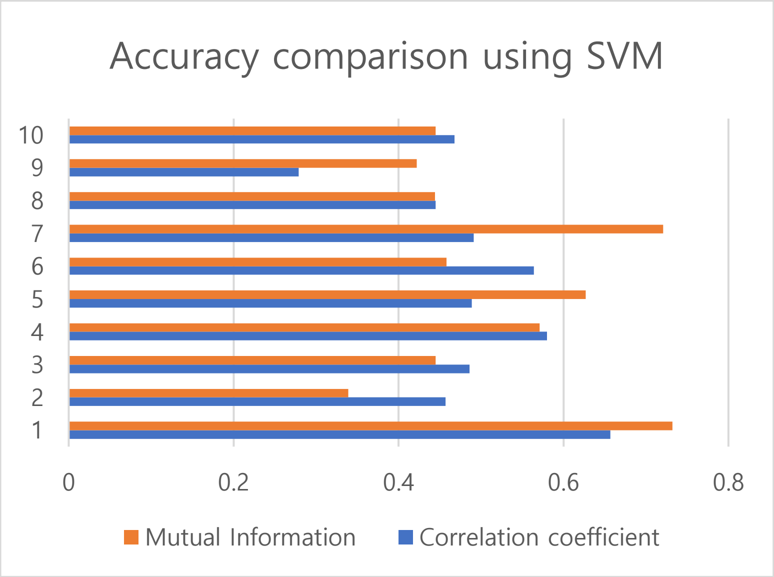


Figure 3.3 Mutual information of each channel

: Channel band power was used as a simple feature of EEG signal



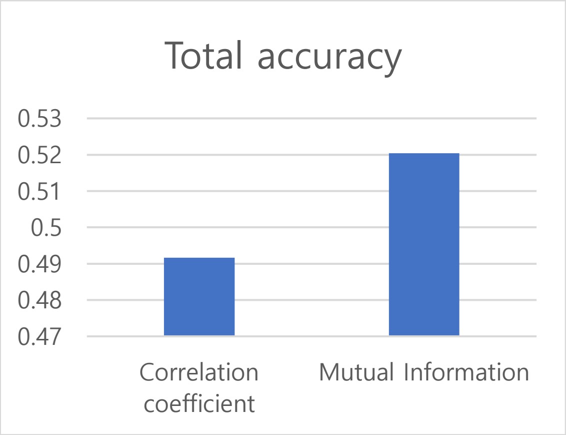


Figure 3.4 Pre-accuracy comparison for channel sensitivity
: Using a single SVM classifier

3.1.5 Classification Results

In the motor imagery brain response task, we extracted the band power feature through sensitivity analysis and then performed the pattern recognition process by inputting it. Basically, SVM, MLP and Naive Bayes algorithms are used as a single classifier and stacking and AdaBoost algorithms are used as ensemble methods. In the case of the stacking method, SVM, MLP and Naive Bayes algorithms are used as sub classifiers and logistic is used as meta classifier. Naive Bayes was used as a sub-algorithm of AdaBoost. Performance evaluation was performed with 5 fold validation, and Weka 3.6 was used as a learning tool.

As can be seen from the classification results, MLP showed the best accuracy of 54.58% in case of single classifier and AdaBoost of 54.27% in ensemble method.

Table 3.1 Classification results accuracy comparison

Dataset	SVM	MLP	Naïve Bayes	Stacking	AdaBoost
Subject1	64.89	52.44	64.89	60.44	63.56
Subject2	44.44	54.22	52.00	48.89	49.33
Subject3	49.33	59.11	52.00	56.89	56.00
Subject4	48.44	49.78	52.89	43.11	50.67
Subject5	58.67	68.44	47.11	65.33	61.33
Subject6	49.78	58.67	46.22	54.22	53.33
Subject7	44.00	49.78	50.22	37.78	44.44
Subject8	71.56	65.78	63.11	67.11	71.11
Subject9	48.00	46.22	41.78	47.11	48.89
Subject10	48.44	41.33	41.78	48.44	44.00
Averaged Accuracy	52.76	54.58	51.20	52.93	54.27

3.2 Statistical Feature Extraction Methods: ARIMA Model Based Feature Extraction Methodology

3.2.1 Introduction

The popular spread of mobile devices means a paradigm shift that is different from the use of existing electronic devices in general life. Beyond simple communication and message delivery, it is now involved in the lives of modern people and their applications are exploding. Analysis of daily life pattern of an individual is also an interesting subject that can utilize various artificial intelligence as a large application field in this flow.

In this paper, we propose a statistical feature vector extraction method using acceleration sensor and skin conductivity as typical wearable sensor data at the previous stage of application of artificial intelligence algorithm. Wearable sensor data can be exposed to many noise conditions due to its characteristics, and a suitable signal processing and extraction methodology is required. In this paper, we have extracted the main component signals of sensor data by using MEMD algorithm [Park2013] and proposed a method of applying ARIMA model, which is a statistical time series regression model, beyond simple methods such as existing signal peak extraction.

Experimental results show that the proposed feature vector extraction technique is suitable for wearable sensor data through the results of conformity test of ARIMA model [Shumway2000].

3.2.2 ARIMA Model

The ARIMA model is a traditional statistical regression model suitable for time series data showing stationarity. It is composed of a combination of an autoregressive model (AR) and a moving average model (MA). In some cases, a signal obtained by taking the derivative and log of the original signal is used to obtain the signal steadiness characteristic [Shumway2000].

In this paper, we use a representative statistical program R for the ARIMA model. We constructed the model by using the function 'auto.arima ()' to obtain the order of the ARIMA model suitable for each signal. The suitability was determined by statistical fit test.

3.2.3 Signal Processing

In this paper, four types of wearable sensor data (3 - axis accelerometer, skin conductivity: GSR) measured by smart watch at 32Hz for 17 minutes in daily life were used. Different sensor signals show different frequency characteristics. However, from the viewpoint of feature vector extraction according to ARIMA adaptation, the main frequency characteristic of skin conductivity is low frequency characteristics less than 1Hz. Accordingly, the MEMD algorithm is applied to the 8Hz low-frequency filter in consideration of the maximum sampling frequency and artifacts both of which are supported by wearable devices. In the case of the accelerometer, a total of 213 sets of experimental data were prepared by dividing the total measured data into five-second windows by preparing the same basic MEMD filtering and preprocessing as the skin conductivity [Park2013].

A multivariate empirical mode decomposition (MEMD) algorithm was used to extract signals with common signal characteristics [Park2013]. The MEMD algorithm is known to extract the dominant frequency components of time series data more efficiently than conventional digital filters (FIR or IIR) or wavelet transforms [Park2013]. Conventional digital filters and wavelet transform algorithms linearly analyze nonlinear electroencephalogram signals by relying on basis functions such as cosine function and mother wavelet function. However, MEMD algorithm extracts basis function adaptively from input signal itself, Time series signals can be analyzed more efficiently. In addition, while conventional analysis algorithms are short channel analysis methods, the MEMD algorithm can analyze multiple channel data at the same time and extract components simultaneously appearing on multiple channels more accurately. The MEMD signal processing algorithm was implemented and implemented on MATLAB.

3.2.4 ARIMA Model Conformance Test

To test the conformity of the wearable sensor data to the ARIMA model, we conducted a fit test using time-series data (Ljung-Box, Augmented Dickey-Fuller test) and Q-test. The statistical program, R, is used. As in the experimental results, it is confirmed whether the p-value value is significant after the test of steady state. And if the residual characteristic of the regression model shows white noise characteristic, it can be concluded that the ARIMA model is suitable.

3.2.5 Experimental Results

In order to apply the ARIMA model, the time series data must maintain the signal steady state characteristics. Since the ARIMA model itself is a regression model for the steady state time series data, the steady state was determined through experiments. In reality, many time-series signals may not have a steady state. Therefore, the signal or log is variably different depending on the signal characteristics [6]. In the case of wearable sensor data, the difference signal was used depending on whether the sample was normal or not.

In this paper, we tested Ljung-Box test and Augmented Dickey-Fuller (ADF) test, which are representative of normality test. Both tests show that the signal is normal when the p-value is low. As a result of p-value verification, it was confirmed that almost all the signals show normal signal characteristics at the statistical significance level as follows.

Table 3.2 Ljung-Box test result

Ljung-Box test	p-Value Mean	p-Value Median	# of > 0.05	# of > 0.05
X	0.000	0.000	0	100
Y	0.000	0.000	0	100
Z	0.000	0.000	0	100
GSR	0.000	0.000	0	100

Table 3.3 Signal steadiness judgment result (ADF test)

ADF test	p-Value Mean	p-Value Median	# of > 0.05	# of > 0.05
X	0.020	0.010	9	95.77
Y	0.027	0.010	16	92.49
Z	0.013	0.010	5	97.65
GSR	0.022	0.010	12	94.37

In this study, we tested Ljung-Box test and Augmented Dickey-Fuller (ADF) test, which are representative of normality test. Both tests show that the signal is normal when the p-value is low. As a result of p-value verification, it was confirmed that almost all the signals show normal signal characteristics at the statistical significance level as follows. After pre-processing the wearable sensor signal, the ARIMA model is obtained by determining the appropriate AR model degree and MA model degree, and then building the ARIMA model coefficient. In this case, whether the constructed ARIMA model is well-structured is confirmed as follows: residual characteristic of the original signal and predicted signal and whether white noise is generated by Q-test. That is, when the fit is good, the residual itself looks like a white noise, and the ACF (Auto Correlation Function) is correlated only in Lag '0'. The Q-test at the bottom shows whether the p-value is outside the significance level. The larger the p-value, the more the signal is white noise. This means that the constructed

ARIMA model is suitable [Shumway2000]. Experimental results show that all four signals are modeled appropriately. The AR and MA model coefficients of the ARIMA model thus constructed can represent the corresponding signal, which can be used as a main feature vector in applications using the next machine learning algorithm.

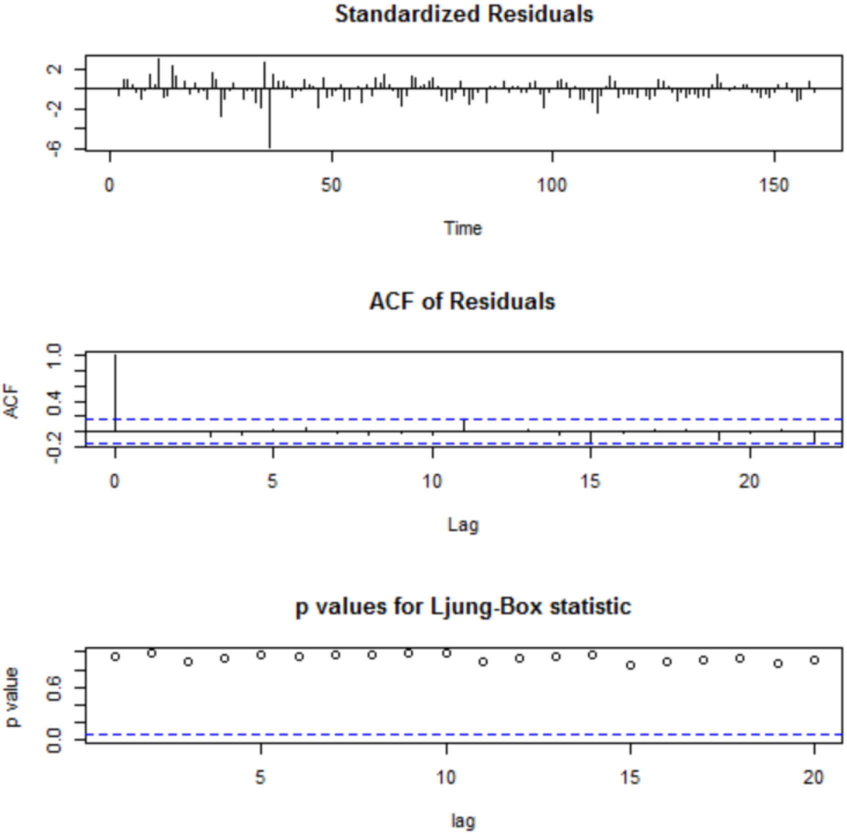


Figure 3.5 Example of ARIMA model conformity result (acceleration sensor)

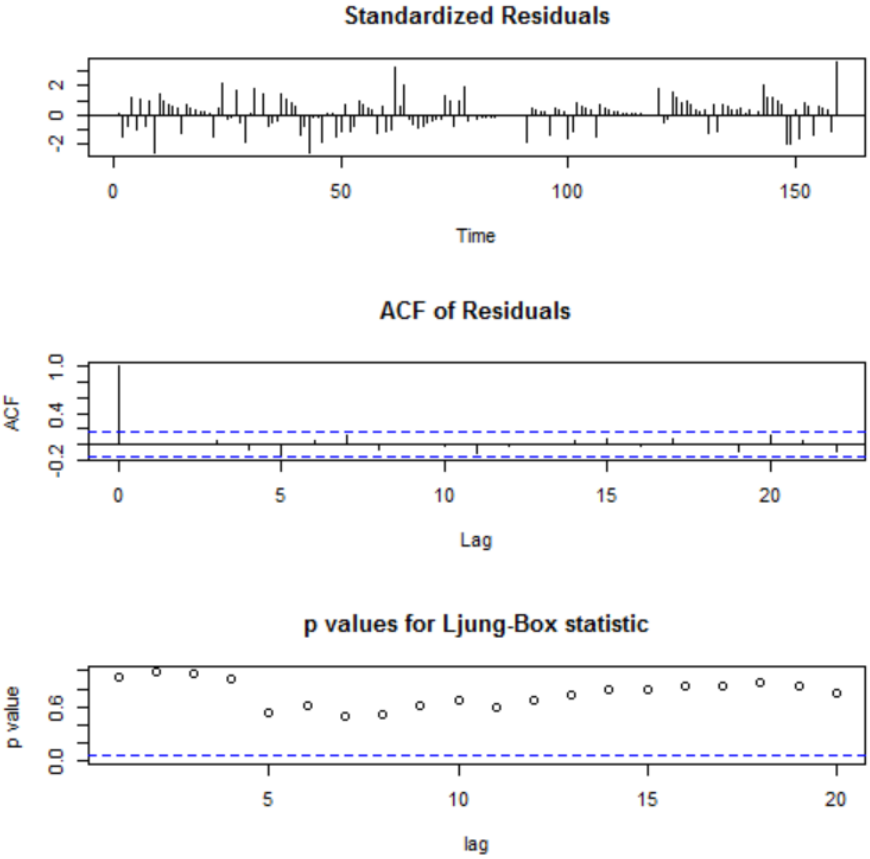


Figure 3.6 Example of ARIMA model conformity result (GSR sensor)

3.2.6 Summary

In this research, we propose a methodology that uses the MEMD algorithm and the ARIMA model, which is a time series regression model, on the feature vector extraction problem of wearable sensor data.

The proposed methodology can be applied to the feature vector extraction of the wearable sensor data by the ARIMA model, and the proposed signal preprocessing technique for the model application is confirmed to be important for the ARIMA model adaptation.

3.3 Applications on Specific Time Series Data: Human Stress States Recognition using Ultra-Short-Term ECG Spectral Feature

3.3.1 Introduction

Human mental stress in daily life is significantly related to health condition [Thorn2007], and thus continuous stress monitoring in life has become crucial for health monitoring. Among various physiological signals, ECG could be a good candidate to measure stressful states, due to its high correlation with autonomous nervous system reflecting stressful responses of the human body [Bong2012].

There have been several researches demonstrating stress monitoring systems using Heart Rate Variability (HRV) parameters [2][4]. However, HRV parameters demand relatively long time period to produce meaningful estimation of physiological status, usually 5-min long data segment [3]. However, in order to develop a practical and real-time stress monitoring system considering ambient noise and motion artifact, the current time length of ECG measurement is too long. Therefore, this study suggests a novel ECG spectral feature, extracted within ultra-short-term period [Baek2015] and representing the characteristics of ECG. The spectral features within 10-sec window could be a replacement of the HRV features, and the experimental results in this paper proved it yielded higher classification performance to detect stressful conditions compared to those of HRV.

3.3.2 Experiments

Subjects and Data Acquisition

Healthy 13 subjects participated in this study and each subject's data was analyzed by individually. Experiments were conducted in indoor laboratory environment. BIOPAC MP36 was used to record ECG signals. Vc+, Vc- and GND electrodes were attached on the right/left arms and right ankle respectively.

Experimental Protocols for Stress Conditions

We designed the five experimental protocols to elicit stressful states. The protocols consisted of Arithmetic, Stroop CWT, Interview, Visual Stimuli and Cold Pressor [Lundberg1994]. The detailed descriptions are shown in below Table 3.4.

Table 3.4 Experimental protocols

Protocols	***Description***
Arithmetic	Subtract 13 from 1022, starting from the beginning when wrong
Stroop CWT	Stress test using colored words
Interview	Short interview with various questions
Visual	Visual test using the negative videos
Cold Pressor	Press ice on skin for a short period of time

The experimental procedure for each subject was followed below Table 3.5.

Table 3.5 Experimental procedures

No.	Test sessions	Duration(min)
1	Pre-test	5
2	Arithmetic	5
3	Resting	2
4	Stroop CWT	5
5	Resting	2
6	Interview	10
7	Resting	2
8	Visual	5
9	Resting	2
10	Cold pressor	1
11	Resting	2
12	Post-test	5
Total		46

During the protocol number 12, Post-test, subjects talked casually without any stress as a control experiment compared to the stressful talking sessions. Since all the stress protocols could be affected by light movement artifacts due to the talks particularly during the sessions of Arithmetic, Stroop CWT and Interview. A similar situation such as the casual talking protocol without stressful conditions were prepared. In this study, this Post-test protocol was compared with the others as a resting state. In addition, the protocol number 10, Cold Pressor, was conducted only for 1 minute since the subjects could not last more than that due to the pain by the coldness of the ice water.

Feature Extraction

Previously, HRV (Heart Rate Variability) parameters have been widely used as a standard ECG features [Bong2012][Baek2015][Selvaraj2013], which demanded long-term datasets of the ECG signal, 5 minutes long in general. This has been an obstacle to develop a real-time practical ECG monitoring system in our daily life due to the difficulty to obtain a stable long-term 5 minutes ECG signal with standing still. Recently, H. Baek *et al.* investigated the meaningful length of ECG signal to yield similar performance as long-term one with shortening the time segment, and eventually proved 10 seconds duration HRV parameters were also significant. Thus, we decided to use the spectral feature of ECG in a 10 seconds window using spectrogram.

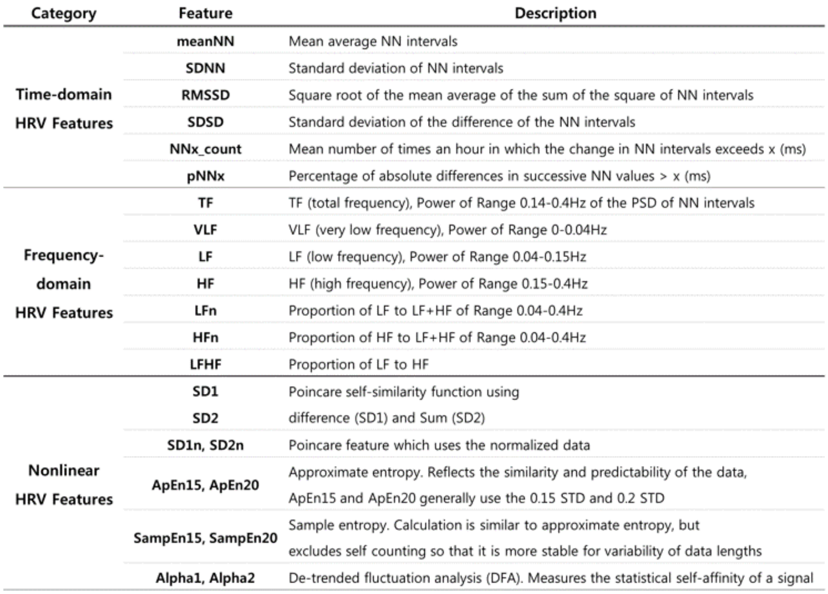
Signal Preprocessing

Signal preprocessing was conducted to extract low frequency components, since ECG signal has most information in relatively lower frequency band [8]. FIR band pass filter (0.1~150Hz) was preceded 60Hz notch filter, eliminating the power line noise.

HRV Parameters

The HRV features of ECG signal were extracted to compare with this study using the ultra-short-term (10 seconds long) spectral features. The detailed descriptions of each parameter are explained in Table 3.6.

Table 3.6 HRV feature description



Category	Feature	Description
Time-domain HRV Features	meanNN	Mean average NN intervals
	SDNN	Standard deviation of NN intervals
	RMSSD	Square root of the mean average of the sum of the square of NN intervals
	SDSD	Standard deviation of the difference of the NN intervals
	NNx_count	Mean number of times an hour in which the change in NN intervals exceeds x (ms)
	pNNx	Percentage of absolute differences in successive NN values > x (ms)
Frequency-domain HRV Features	TF	TF (total frequency), Power of Range 0.14-0.4Hz of the PSD of NN intervals
	VLF	VLF (very low frequency), Power of Range 0-0.04Hz
	LF	LF (low frequency), Power of Range 0.04-0.15Hz
	HF	HF (high frequency), Power of Range 0.15-0.4Hz
	LFn	Proportion of LF to LF+HF of Range 0.04-0.4Hz
	HFn	Proportion of HF to LF+HF of Range 0.04-0.4Hz
	LFHF	Proportion of LF to HF
Nonlinear HRV Features	SD1 SD2	Poincare self-similarity function using difference (SD1) and Sum (SD2)
	SD1n, SD2n	Poincare feature which uses the normalized data
	ApEn15, ApEn20	Approximate entropy. Reflects the similarity and predictability of the data, ApEn15 and ApEn20 generally use the 0.15 STD and 0.2 STD
	SampEn15, SampEn20	Sample entropy. Calculation is similar to approximate entropy, but excludes self counting so that it is more stable for variability of data lengths
	Alpha1, Alpha2	De-trended fluctuation analysis (DFA). Measures the statistical self-affinity of a signal

Spectral Features (Spectrogram)

The main spectral feature was extracted using the spectrogram, the short-time Fourier transform of a time series. The transformed spectral value itself in the frequency domain was utilized as the main feature in this study. The time window was set to 10 seconds based on the investigation by Beak *et al.* [Beak2015]. A spectral feature example corresponding to stress states is shown in Figure 3.7.

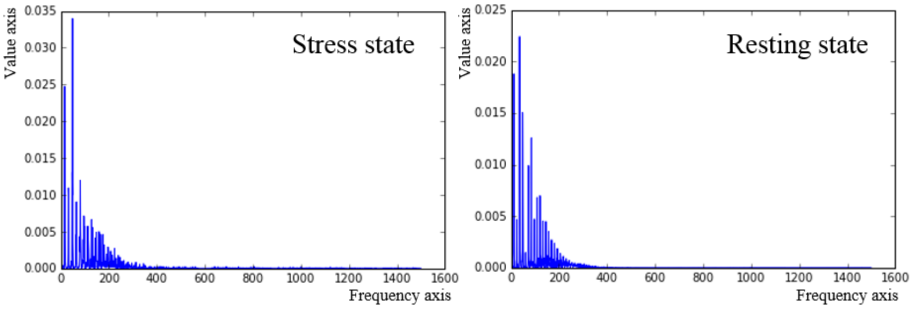


Figure 3.7 Spectral feature description

3.3.3 Classification Methods

Several machine learning classification algorithms were applied to investigate the performance 10 seconds window spectral features to separate the stressful and resting conditions. The scikit-learn machine learning library was used as main analysis tools. 5-fold cross validation was applied to measure the classification accuracies. The algorithms are listed in Table 3.7.

Table 3.7 Applied classification algorithms

Software tools	**Classification algorithms**
Python scikit-learn library	KNeighbors
	Linear SVM
	RBF SVM
	Gaussian Process
	Decision Tree
	Random Forest
	Neural Network
	AdaBoost
	Naïve Bayes
	Quadratic Discriminant Analysis

3.3.4 Experimental Results

This study investigated the binary classification problem between the stressful and resting states using the aforementioned spectral features. At first, the experiment using HRV parameters was conducted for the benchmark test with this study. Figure 3.8 shows the classification accuracy corresponding to the time window size of 1 minutes. The 1-min time window was set as the optimal window length within the frame of classification accuracy experimentally.

In this experiment, the whole experimental protocol data were mixed because of the limited small data size. The accuracy was achieved using 5-fold cross validation. As can be seen in Figure 3.9, AdaBoost classifier shows the best accuracy in HRV parameter case, 65.90%.

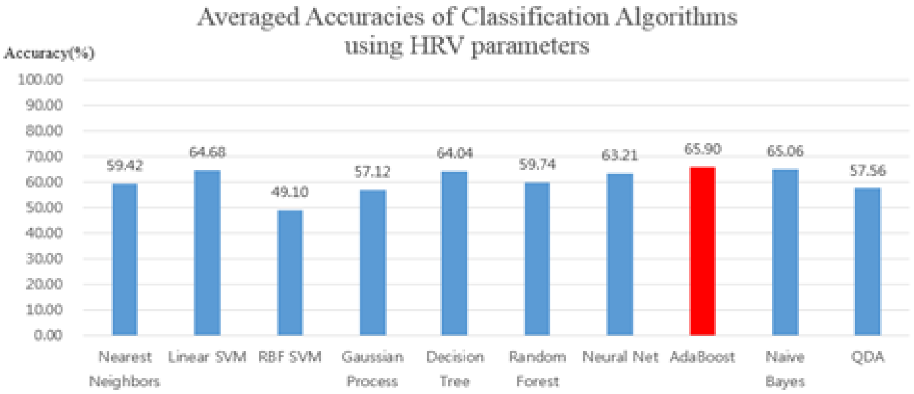


Figure 3.8 Classification result using HRV parameters

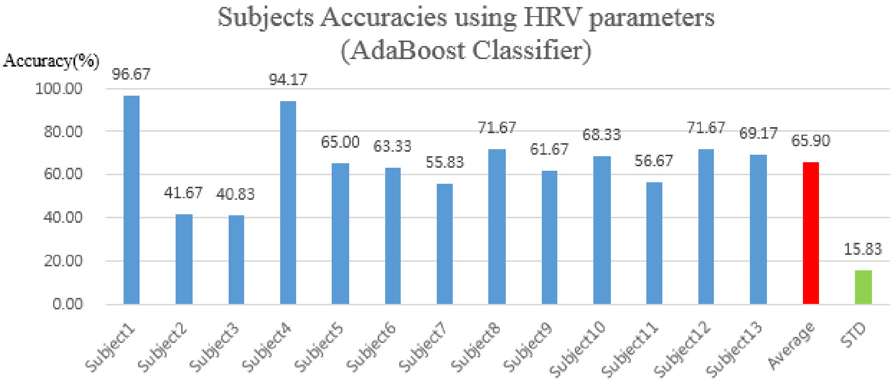


Figure 3.9 Classification result using HRV parameters for subjects

Secondly, the target experiment was taken using the ultra-short-term spectral feature. 5-fold cross validation was also conducted for the whole protocol mixed data to compare with the previous HRV experiments.

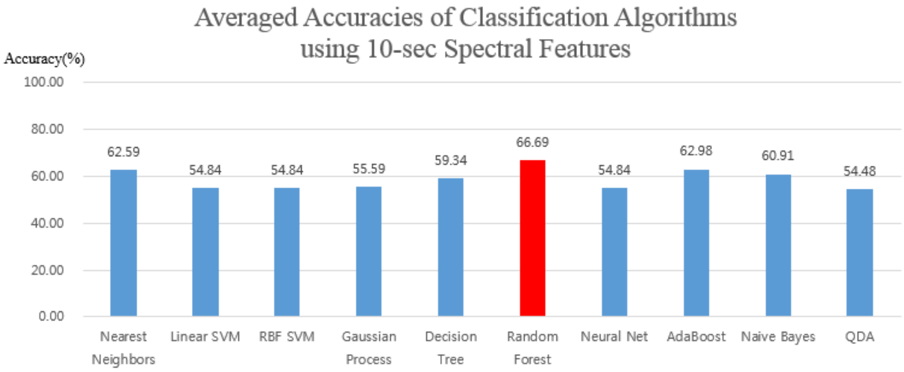


Figure 3.10 Classification result using 10-sec spectral features

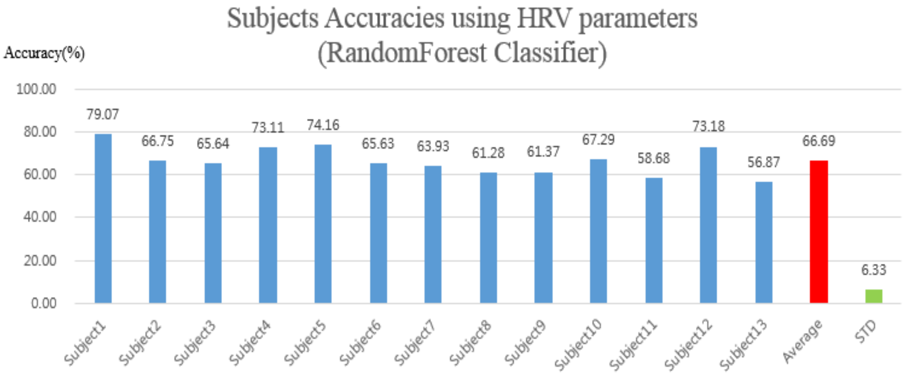


Figure 3.11 Classification result using 10-sec spectral features for subjects

As can be seen in the above results, the classification performance shows 66.69% accuracy using Random Forest classifier. Even though the result does not shows the superior performance within the frame of accuracy, but shows the feasibility of stress state classification using 10-sec ECG data and the stable result within the frame of standard deviation in Figure 3.11.

In addition to these results, protocol analysis can be possible because the ultra-short-term ECG can get more data compared to the HRV. In HRV case, only 4 instances (4-min/1-minHRV) could be collected for one protocol, therefore protocol analysis is not possible.

The protocol experiment was taken using the ultra-short-term spectral feature. 5-fold cross validation was also conducted corresponding to the experimental protocols. The performances of the various machine learning classifiers were shown in Table 3.8, which demonstrates Naïve Bayes and ensemble machines (AdaBoost) yielded higher performances on emotional stress state classification problem using the ultra-short-term spectral feature than the others.

Table 3.8 Performance comparisons of classification algorithms on experimental protocols for protocol experiments

Accuracy (%)	Arithmetic	Stroop CWT	Interview	Visual	Cold Pressor	Algorithm Mean
KNeighbors	72.31	63.44	66.62	77.69	74.23	70.86
Linear SVM	76.13	66.54	68.03	78.41	75.00	72.82
RBF SVM	69.51	65.03	66.21	70.23	67.69	67.73
Gaussian Process	77.92	69.26	69.31	80.46	77.69	74.93
Decision Tree	78.38	76.97	68.31	77.54	86.92	77.62
Random Forest	79.67	71.64	68.74	66.59	85.77	74.48
Neural Network	78.23	71.31	70.49	81.28	82.69	76.80
AdaBoost	82.15	80.08	73.59	81.85	82.69	80.07
Naïve Bayes	87.31	81.10	75.87	79.97	81.54	81.16
Quadratic Discriminant Analysis	58.31	58.82	58.62	65.90	61.54	60.64
Protocol Mean	75.99	70.42	68.58	75.99	77.58	73.71
Protocol Max	87.31	81.10	75.87	81.85	86.92	82.61

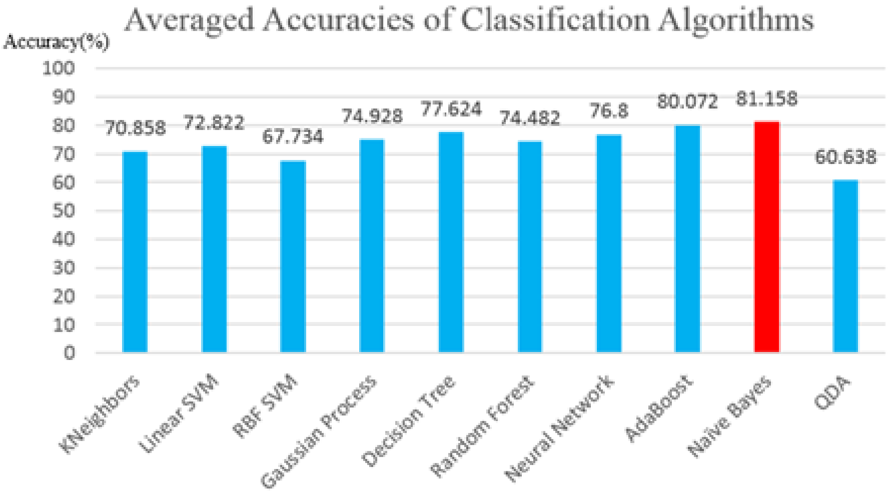


Figure 3.12 Summary of Algorithm Performance Comparison for protocol experiments

Interestingly, the Arithmetic protocol showed the best classification performance, which might infer the mental burden could make stressful state in human body.

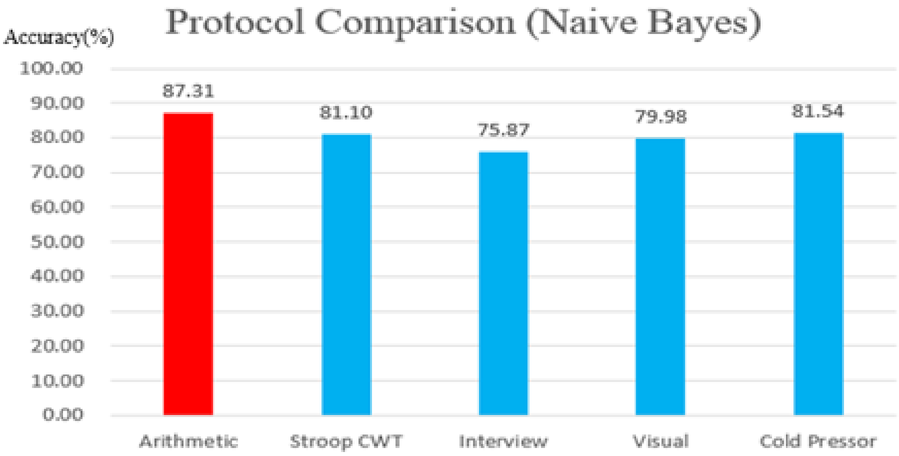


Figure 3.13 Protocol Comparison Result (Naïve Bayes)

The classification accuracies corresponding to the subjects are shown in Table 3.9 for the best classifier, Naïve Bayes.

Table 3.9 Performance comparisons of subjects on experimental protocols (Naïve Bayes)

Accuracy (%)	**Arith-metic**	**Stroop CWT**	**Inter-view**	**Visual**	**Cold Pressor**	**Subject Mean**
Sub 1	100.00	92.33	73.67	76.67	80.00	84.53
Sub 2	98.33	94.33	98.00	88.67	70.00	89.87
Sub 3	100.00	96.67	70.33	65.00	75.00	81.40
Sub 4	100.00	100.00	100.00	100.00	95.00	99.00
Sub 5	76.67	60.67	61.67	94.00	100.00	78.60
Sub 6	87.33	92.67	89.33	90.67	80.00	88.00
Sub 7	54.00	64.33	51.00	70.33	80.00	63.93
Sub 8	94.00	62.67	77.33	63.33	100.00	79.47
Sub 9	85.00	61.00	74.00	64.67	75.00	71.93
Sub 10	96.33	73.67	72.00	65.67	80.00	77.53
Sub 11	85.00	93.00	67.33	90.67	90.00	85.20
Sub 12	89.00	96.00	95.00	98.00	85.00	92.60
Sub 13	69.33	67.00	56.67	72.00	50.00	63.00
Protocol Mean	87.31	81.10	75.87	79.98	81.54	81.16
Protocol Max	100.00	100.00	100.00	100.00	100.00	99.00

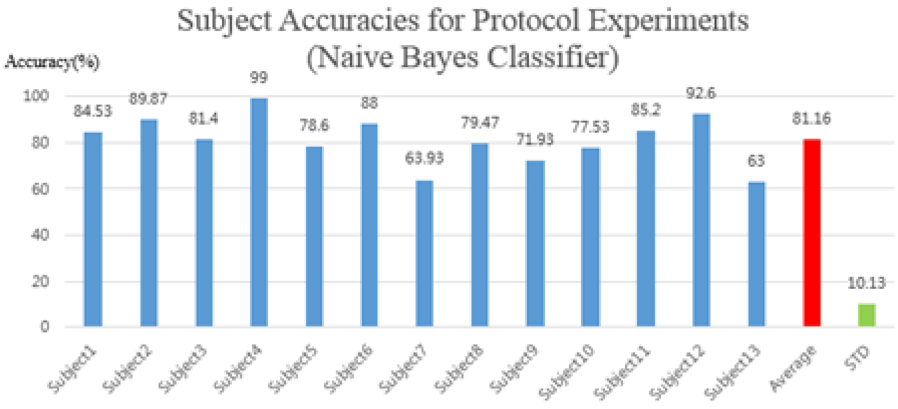


Figure 3.14 Experimental Results on All Subjects

3.3.5 Summary

Our research works proved the effectiveness of the ultra-short-term spectral feature of ECG for emotional stress state classification problem. We also demonstrated which classification algorithms were optimal for the proposed feature, and which stress situation was the most stressful among various stressful conditions.

This proposed approach could be adopted to wearable system in a real-time manner, and also implemented into various areas in easy way without domain specific feature engineering.

Chapter 4

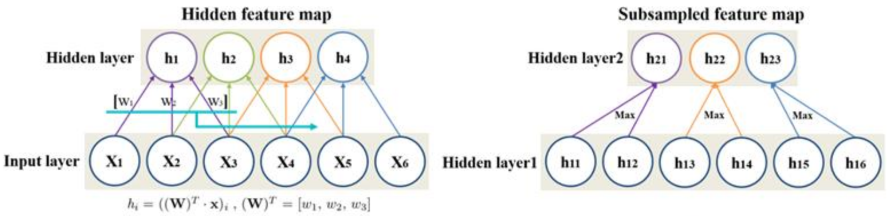
Master Frameworks for Pattern Recognition in Time Series

4.1 The Concept of the Proposed Framework for Pattern Recognition in Time Series

This thesis proposes a total framework for pattern recognition of time series data. It is an end-to-end deep learning framework which basically has no hand-crafted feature extraction process and contains deep learning design methods for both periodic and non-periodic time series signals. This chapter explains the whole concept of the master framework for time series data using the proposed methodologies.

4.1.1 Optimal Basic Deep Learning Models for the Proposed Framework

In the proposed methodology, 1D convolutional neural networks were used to generate hidden representations of time series signals. The operation procedure and details of the convolution layer are illustrated in Figure 4.1(a), where the 1D filter is implemented in each channel of the hidden layer. During the learning process of CNNs, the filter coefficients are determined with an optimization learning process using error backpropagation according to input signals and output labels.



(a) 1D convolution operation

(b) 1D pooling operation

Figure 4.1 1D convolution and pooling operations

Figure 4.1(b) illustrates the structure of the pooling layer after the convolution operation. Since the convolution operation produces similar outputs across time, subsampling by taking the maximum or average value in a particular window size makes it possible to reduce the feature dimension. In addition, the patterns of data could be well recognized owing to this subsampling, choosing only one value within the pooling operation window, despite data variations such as placement, scale and rotation of data [LeCun2012] [LeCun2015].

Due to the above reasons, 1D CNNs were chosen as the most suitable unit to generate the basic hidden features of time series signal in the entire deep learning structure. After the CNNs, MLP or RNNs can be used as an additional deep learning unit to generate the final feature representation. The proposed framework can be divided into two categories according to the periodicity of time series signals. Figure 4.2 shows the basic concept diagram of the proposed framework and the basic 1D CNNs module.

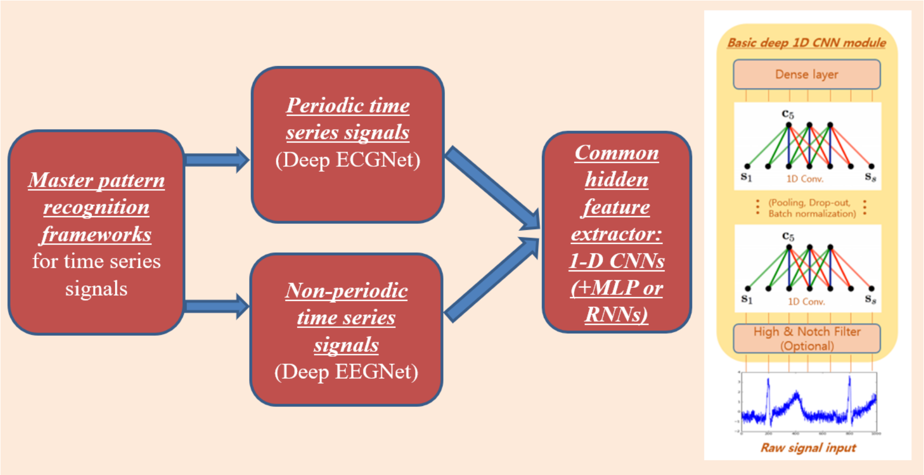

Master pattern recognition frameworks for time series signals
Periodic time series signals (Deep ECGNet)
Non-periodic time series signals (Deep EEGNet)
Common hidden feature extractor: 1-D CNNs (+MLP or RNNs)
Basic deep 1D CNN module
Dense layer
c_5
s_1
1D Conv.
s_s
(Pooling, Drop-out, Batch normalization)
c_5
s_1
1D Conv.
s_s
High & Notch Filter (Optional)
Raw signal input

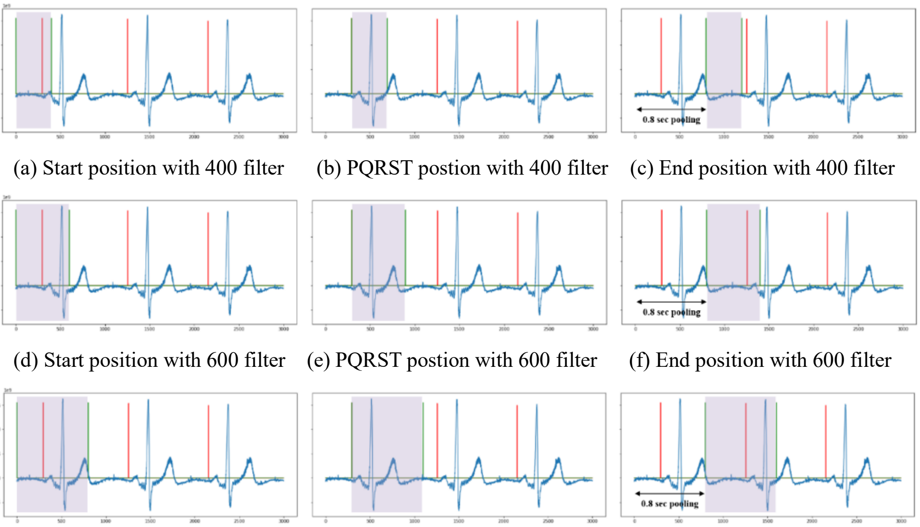
Figure 4.2 The proposed framework concept and the basic deep learning unit

4.2 Two Categories for Pattern Recognition in Time Series Data

The type of time series signals can be divided into periodic signals and non-periodic signal. Depending on the type of data, the design method of each model can be approached differently. In the case of periodic signals, it is possible to design the proper periodic characteristic of the signal directly to the model. On the other hand, in the case of non-periodic signals, it is difficult to intuitively reflect these characteristics in the model, so a design method can be taken to design the model in terms of optimization.

4.2.1 The Proposed Deep Learning Framework for Periodic Time Series Signals

The proposed model uses 1D CNNs as a basic unit. For periodic signals, it is necessary to reflect their periodic characteristics in the model. Figure 4.3 and 4.4 show the appropriate convolution filter length and pooling length based on the signal's periodic characteristics. In short, a model that reflects the periodic nature of a signal can capture key feature regions during one period of the signal, thereby enabling effective pattern recognition without special signal processing or additional processes. The detailed explanations of design methodology for periodic signals is described in Ch. 5 Deep ECGNet.



(a) Start position with 400 filter (b) PQRST postion with 400 filter (c) End position with 400 filter

(d) Start position with 600 filter (e) PQRST postion with 600 filter (f) End position with 600 filter

(g) Start position with 800 filter (h) PQRST postion with 800 filter (i) End position with 800 filter

Figure 4.3 Examples of appropriate convolution filter length settings (Where, the red line is the starting point of ECG P-wave, and the purple box is the coverage region of convolution filter. The 600 filter shows the best performance for capturing of PQRST region.)

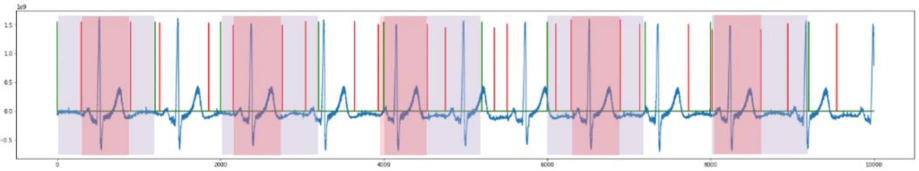


Figure 4.4 Examples of appropriate pooling length settings (Where, the red box is the PQRST regions of ECG signal and the purple box is the convolution operation region when 800-pooling applied.)

4.2.2 The Proposed Deep Learning Framework for Non-periodic Time Series Signals

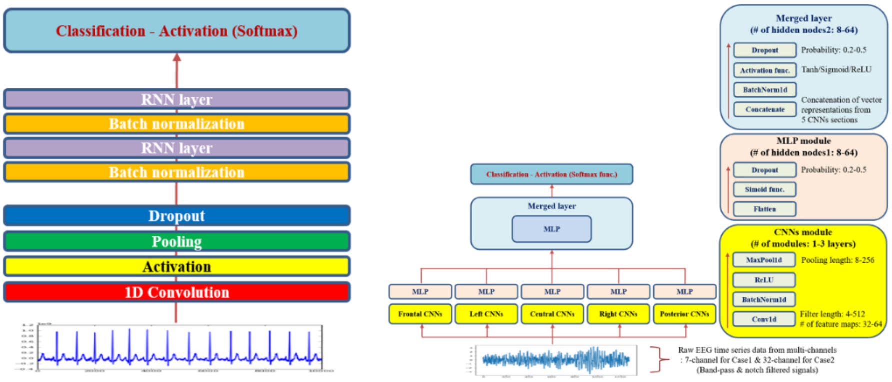
When constructing modelds for pattern recognition of non-periodic signals, it is difficult to incorporate prior knowledge such as periodic signals into the model. In this case, as you can see in the example in Table 4.1, there are difficult problems in choosing a number of hyperparameters and structures. In order to solve these complex and difficult problems, we propose the use of an appropriate automatic optimization technique. It is proposed to fix the basic deep running unit with 1D CNNs and to find the optimal combination of the hyperparameters and structures of the whole model within the set boundary. In this research, we applied the Bayesian optimization technique for the automatic end-to-end deep learning model design. The detailed explanations of design methodology for non-periodic signals is described in Ch. 6 Deep EEGNet.

Table 4.1 Examples of various hyperparameters and structures for model construction of non-periodic signals

Hyperparameters	**Data type**	**Range (Case 1/Case 2)**
# of CNNs feature maps	Integer	32-64/32-64
CNNs depth	Integer	1-3/1-3
Convolution filter length	Integer	4-256/8-512
Pooling length	Integer	16-256/8-128
RNNs (LSTM) depth	Integer	1-2/1-2
Model structure type	Enumeration	CNNs+MLP/CNNs+RNNs
# of nodes for hidden layer1	Integer	8-64/8-64
# of nodes for hidden layer2	Integer	8-64/8-64
Batch size	Integer	32,64,128/32,64,128

Regularization coefficients(L1/L2)	Float	0.0001-0.005/0.0001-0.005
Early stop epochs	Integer	30-300/30-50
Drop out probability	Float	0.2-0.5/0.2-0.5
Learning rate	Float	0.001-0.1/0.001-0.1
Momentum	Float	0.0-0.2/0.0-0.2
Optimizer type	Enumeration	Adam/RMSprop/SGD
Activation function type	Enumeration	Tanh/Sigmoid/ReLU

Figure 4.5 shows two types of implemented models in this thesis. The details are in Ch. 5 and 6.



(a) Periodic signals model (b) Nonperiodic signal model

Figure 4.5 Examples of the proposed frameworks

4.3 Expanded models of the proposed master framework for pattern recognition in time series

The master framework for pattern recognition of time series data based on the proposed 1D CNNs so far can be conceptually expanded to various forms. To give some examples, the basic framework proposed in Figure 4.5 can be expanded as shown in Figure 4.6 in a unimodal structure, and can also be expanded to a multimodal form as shown in Figure 4.7. In addition, it can be also expanded to the form of adopting the attention mechanism as shown in Figure 4.8.

In addition, the proposed methodologies for periodic and non-periodic model construction can be effectively applied to various combined end-to-end deep learning architectures.

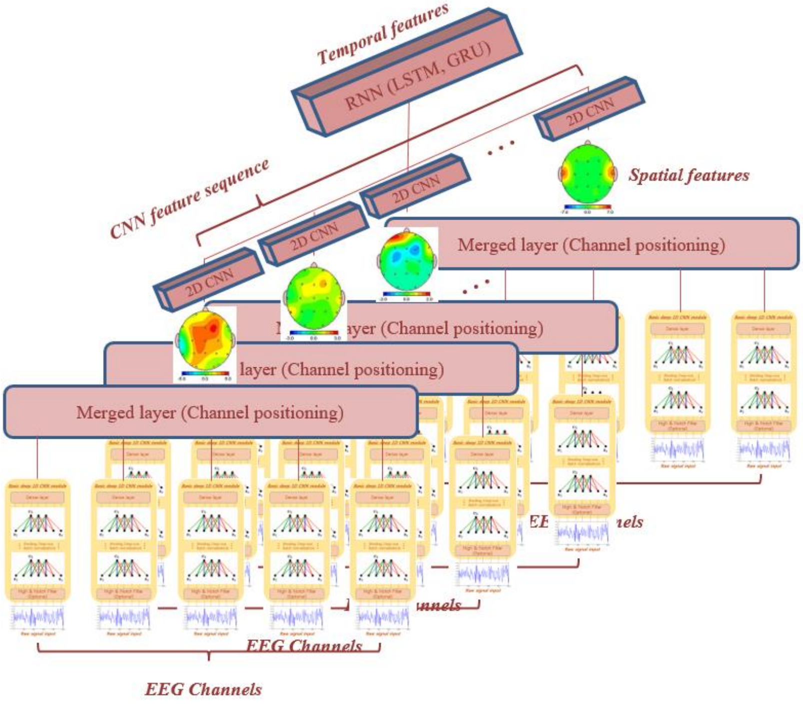

Temporal features
RNN (LSTM, GRU)
CNN feature sequence
2D CNN
2D CNN
2D CNN
2D CNN
2D CNN
Spatial features
Merged layer (Channel positioning)
layer (Channel positioning)
layer (Channel positioning)
Merged layer (Channel positioning)
EEG Channels
EEG Channels

Figure 4.6 Unimodal expansion of the proposed framework

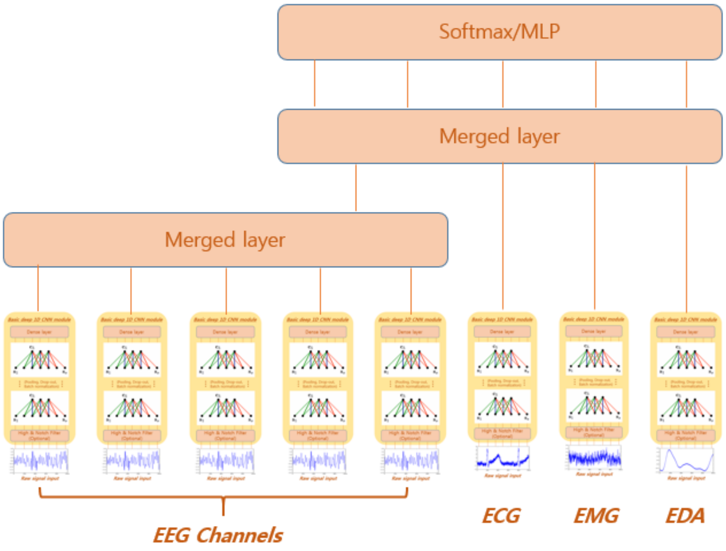


Figure 4.7 Multimodal expansion of the proposed framework

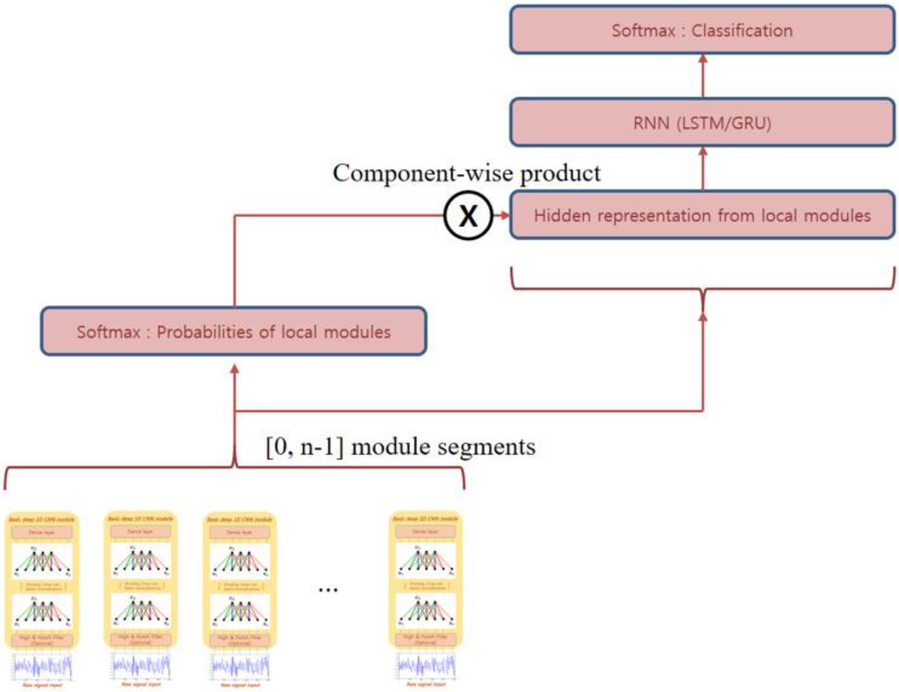

Softmax : Classification
RNN (LSTM/GRU)
Component-wise product
X
Hidden representation from local modules
Softmax : Probabilities of local modules
[0, n-1] module segments
Basic deep 1D CNN module
Dense layer
(Pooling, Drop-out, Batch normalization)
High & Notch Filter (Optional)
Raw signal input
...

Figure 4.8 Attention mechanism expansion of the proposed framework

Chapter 5

Deep Learning Model Design Methodology for Periodic Time Series Signals using Prior Knowledge: Deep ECGNet

5.1 Introduction

Psycho-physiological studies have demonstrated that the physiological features of human beings are affected by their stress levels [Ekman1983]. While the sympathetic and parasympathetic divisions in the autonomous nervous system (ANS) balance each other in stress-free condition, domination of the sympathetic division in stressful condition causes ANS unbalance. Psychological factors have been suggested to be associated with cardiovascular disorders, which are mediated by neuroendocrine and psychophysiological mechanisms [Hjortskov2004]. In addition, Belkic *et al.* have also demonstrated that psychological stress is a risk factor for cardiovascular disorders using epidemiological studies [Belkic2000]. Based on these, electrocardiogram (ECG), recorded from the electrophysiologic patterns of heart muscle during heartbeat, could be a promising candidate to provide a biomarker to estimate event-based stress level.

Traditionally, stress has been examined by psychologists using questionnaires and interview procedures [Karthikeyan2013]. However, in contrast with this subjective approach, tools were developed to assess the more objective biological stress response such as biochemical samples (urine, saliva and blood samples) and physiological signals (ECG, galvanic skin response (GSR), electromyogram (EMG), electroencephalogram (EEG), blood pressure (BP), and so on).

Conventionally, the beat-to-beat alternations, heart rate variability (HRV), from ECG have been utilized to monitor the mental stress status as well as the mortality of cardiac patients [Task Force1996]. HRV features could measure ANS activity quantitatively during mental stress, which increased the low frequency components of HRV (high sympathetic activity) [Holly1997], but decreased high frequency HRV (low parasympathetic activity) [Hjortskov2004]. Traditionally, five minute or 24 hour ECG recordings have been used to calculate HRV for the analysis of cardiac diseases [Task Force1996]. While the measurement and analysis of long-term records spanning 24 hours is difficult and not reproducible, the short-term measurement spanning 5 minutes is more practical in its setting and its signal analysis [Baek2015]. In addition, ultra short-term HRV analysis using less than 5-minute data has been recently conducted due to the increased demands for practical and reliable applications of ambulatory ECG monitoring devices. Recently, there have been several studies performing the statistical comparison of the HRV parameters between long-term and ultra short-term recordings varying from 5 minutes to 10 seconds [Baek2015]. However, these studies were mostly conducted for clinical purposes and applications, and they sticked to the traditional HRV parameters.

In recent years, deep learning methods have been yielding excellent performance in areas of pattern recognitions [LeCun2015]. Especially CNNs (Convolutional Neural Networks) has outperformed conventional approaches in an image recognition area due to its automatic extraction of data characteristics using convolution and pooling operations [LeCun2015]

[LeCun2010]. Additionally, RNNs (Recurrent Neural Networks) has been another successful deep learning model in the areas of language modeling, handwritten recognition and speech recognition [LeCun2015] [Lipton2015]. In this study, the significance of ultra short ECG recordings during mental stress were studied and a novel deep learning architecture was suggested to extract features from ECG signals in a more generic way than the conventional HRV parameters. In particular, we studied how to apply CNNs and RNNs to 1D time series data of raw ECG signals and optimize the deep learning structure and its parameters for the stress recognition based on the periodic characteristics of the ECG signals.

5.2 Materials and Methods

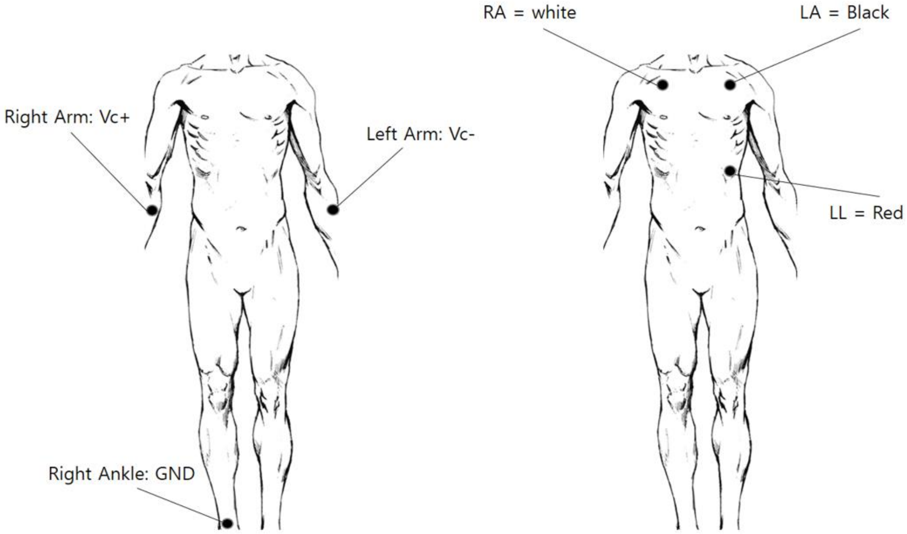
5.2.1 Subjects and Data Acquisition

Kwangwoon University Multiple Stress Stimulus Experiments: Case1

Twenty healthy male subjects were recruited at Kwangwoon university using flyers and posters. The enrolled subjects were in their 20s ranging between 22 and 26. They had not participated in an experiment using any mental/physical stress task before. After every subject submitted the written informed consent, ECG signals were recorded from Einthoven Lead I (see Figure 5.1(a)) using an ECG amplifier (Biopac MP36 system, Biopac System Inc. Goleta, CA), where the sampling frequency was 1000 Hz. As basic preprocessing, a 2000-order FIR notch filter was applied to remove the power line noise in the frequency range 58–62 Hz, and an additional 3000-order FIR bandpass filter was applied to keep the frequency components in the range 1.5–150 Hz. Subjects who suffered from cardiovascular or major psychiatric diseases were excluded since it might affect the ECG patterns.

Table 5.1 Procedure of the stressor tests

Trials	Duration (min.)
Rest session	4
Arithmetic	5
Rest session	4
CWT	5
Rest session	4
Interview	5 (+5 for preparation)
Rest session	4
Visual stimuli	5
Rest session	4
Cold pressor	1
Rest session	4



(a) ECG electrode placement for Case1 (b) ECG electrode placement for Case2

Figure 5.1 ECG signal acquisitions

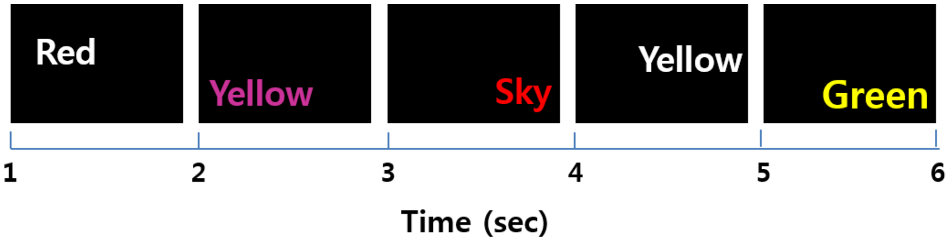
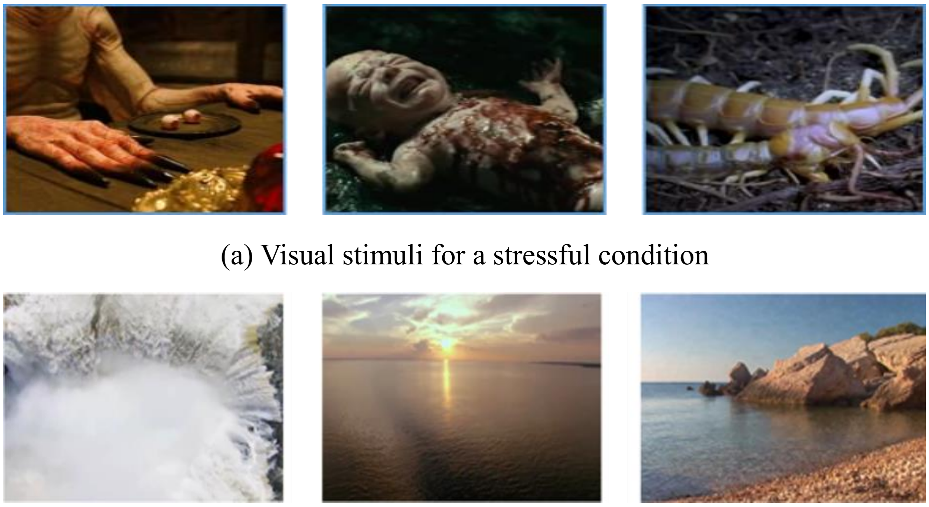


Figure 5.2 Example of the sequential visual stimuli of CWT across time.



(a) Visual stimuli for a stressful condition

(b) Visual stimuli for a resting condition

Figure 5.3 Examples of the visual stimuli to elicit

The experiment session consisted of five sessions with five different stressful stimuli including mental arithmetic test, the stroop color word test (CWT) (see Figure 5.2), an interview, a visual stimuli test (see Figure 5.3) and a cold pressor test. Every stressor test was followed by a rest period before the next started. During the recording, the subjects were instructed not to move in order to reduce the motion artifact. Table 5.1 provides the details of the experiment procedure including the sequence of the sessions and duration of the recording. All the stress tests were conducted for 5 minutes except for the cold pressor test due to the high pain.

The details of the stressful experimental protocols are described in Table 5.2.

Table 5.2 Experimental protocol descriptions for Case1

Protocols	Descriptions
Arithmetic	Subtract 13 from 1022 without writing, in which subjects start over when a wrong answer is given.
CWT	On a screen, colored words are shown, and subjects are instructed to say the color of the letters, not the color that the word represents (See Figure 5.2).
Interview	Subjects prepare an answer to the interview questions for five minutes, and then perform an interview for an additional five minutes.
Visual stimuli	Five video streams are shown to subjects, which trigger a stressful condition (See Figure 5.3).
Cold pressor	Subjects are instructed to keep their hand in cold water (0°C ~ 4°C) for a minute.

This study was reviewed and approved by the institutional review board at the Korea national institute for bioethics.

KU Leuven University Mental Arithmetic Task Experiments: Case2

Thirty healthy subjects were recruited at Maastricht University (27 female; 3 male) using flyers and posters. Inclusion criteria were i) age 18 – 35, ii) sufficient command of the Dutch language to understand and give informed consent, and iii) mental competence to understand and follow instructions. Participants were excluded if they previously participated in a study using a mental arithmetic task. In a subsample of 9 participants, heart rate was measured during the experimental task using three chest leads like Figure 5.1(b), where sampling frequency was 1000 Hz.

During the experiment, psychosocial stress was induced with a modified version of the Montrèal Imaging Stress Test (MIST), a mental arithmetic task with a psychosocial evaluation component in the form of scripted negative feedback by an experimenter (Pruessner JC, Champagne F, Meaney MJ, Dagher A, 2004). The task consisted of four runs, each divided in six 7-minute blocks (30 seconds rest; 60 seconds control; 120 seconds experimental; 30 seconds rest; 60 seconds control; 120 seconds experimental). During the experimental blocks, participants solved mathematical equations under time-pressure on a computer, while their performance was visualized as an arrow on a performance bar. The task difficulty was manipulated by a computer algorithm such that participants on average were able to provide the correct answers in time only for 60-70% of the cases. In the current version, participants were always tested in couples, sitting in separate testing rooms. A social defeat component was added by a cover story saying that participants were competing against each other and an arrow indicating the opponent's performance was visible above the performance of the participant. In reality, the opponent's arrow was manipulated by the program to always outperform the participant's and both participants were made to believe that their opponent performed better. During the control blocks, participants solved equations without time-limit and performance arrows; the rest blocks existed of passive watching of the task interface.

As basic preprocessing, a 2000-order FIR notch filter was applied to remove the power line noise in the frequency range 48–52 Hz, and an additional 3000-order FIR bandpass filter was applied to keep the frequency components in the range 1.5–150 Hz.

This study was approved by the Ethical Committee Psychology of the Maastricht University. Participants were treated in accordance with APA standards. All participants gave written informed consent prior to the study.

5.2.2 Conventional ECG Analysis Methods

Conventional ECG features were extracted by the methods shown in Table 5.3, based on the pre-defined formulas [Taelman2009]. In addition, long time series of ECG signal, 5 minutes long, were utilized for the standard HRV parameter analysis, impractical in real-time stress classification.

To compare with the performance of the proposed deep learning approach, this conventional HRV analysis was conducted as a benchmark method. For HRV experiments, two experimental cases were tested according to the correlation coefficients between standard HRV parameters and 10-sec HRV parameters [Baek2015].

The first HRV experiment was a classification experiment using only one parameter (heart rate, the correlation coefficient, > 0.9, HRV-1 case), and the second experiment was a classification experiment with 4 parameters (the correlation coefficient > 0.6, HRV-4 case). For HRV-4 case, HR, SDNN, RMSSD, pNN50 and HF parameters satisfied the conditions shown in Table 5.3 [Baek2015]. However, pNN50 was excluded because of abnormal extracted values. Therefore, four parameters (HR, SDNN, RMSSD and HF) are used as main features except pNN50 for HRV-4 case. (See Table 5.3)

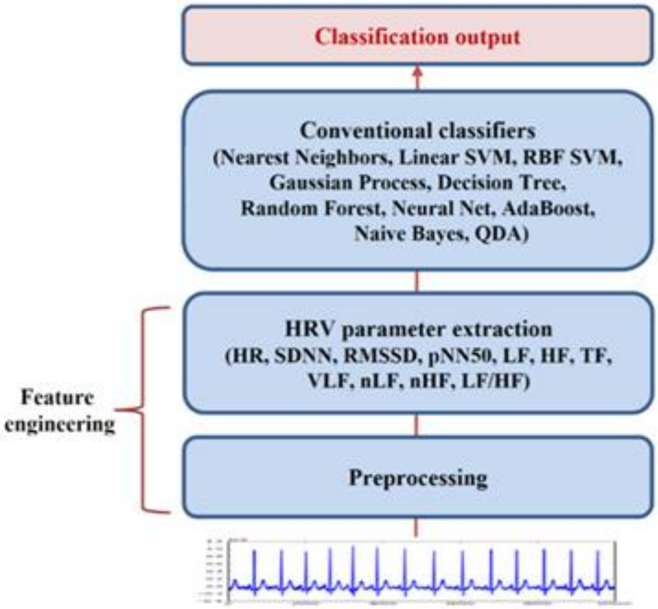
Table 5.3 The correlation coefficients between short 10-sec HRV parameters and standard 5-minute HRV parameters and their descriptions

HRV parameters	Correlation coefficients	Descriptions of HRV parameters
HR	***0.9321(‡)***	Mean average NN intervals
SDNN	***0.6203***	Standard deviation of NN intervals
RMSSD	***0.636***	Square root of the mean average of the sum of the square
pNN50	0.6528	Percentage of absolute difference in successive NN values > 50 (ms)
LF	0.4156	LF (low frequency), Power of Range 0.04-0.15Hz
HF	***0.6294***	HF (high frequency), Power of Range 0.15-0.4Hz
TF	0.5397	TF (total frequency), Power of Range 0.14-0.4Hz of the PSD of NN intervals
VLF	0.0357	VLF (very low frequency), Power of Range 0-0.04Hz
nLF	0.3869	Proportion of LF to LF+HF of Range 0.04-0.4Hz
nHF	0.3869	Proportion of HF to LF+HF of Range 0.04-0.4Hz
LF/HF	0.1503	Proportion of LF to HF

Kruskal-Wallis test: p>0.5(‡)

The spectrogram of ECG signals was also used as a benchmark method. Although frequency analysis has been a popular method to look into the physiological signal, it is also based on predefined basis functions such as sinusoids during the process of Fourier transform. And the physiological signal like ECG is nonlinear and nonstationary, hard to be generalized with a fixed basis function and deterministic/stationary model [Park2014].

This study proposes an end-to-end deep learning method to overcome these drawbacks by extracting the features from the input data itself without an unrealistic predefined basis function or model. In order to evaluate the performance of the proposed approach, the performance of HRV parameters with conventional classifiers was tested (see Figure 5.4(a)). In addition, the spectrogram/frequency analysis was also conducted to compare the proposed method with the Fourier based linear/deterministic analysis. In this spectrogram analysis, the transformed frequency amplitude value using Fourier transform were used as a feature to classifiers, depicted in Figure 5.4(b).


Classification output
Conventional classifiers
(Nearest Neighbors, Linear SVM, RBF SVM,
Gaussian Process, Decision Tree,
Random Forest, Neural Net, AdaBoost,
Naive Bayes, QDA)
HRV parameter extraction
(HR, SDNN, RMSSD, pNN50, LF, HF, TF,
VLF, nLF, nHF, LF/HF)
Feature
engineering
Preprocessing

(a) HRV parameter analysis

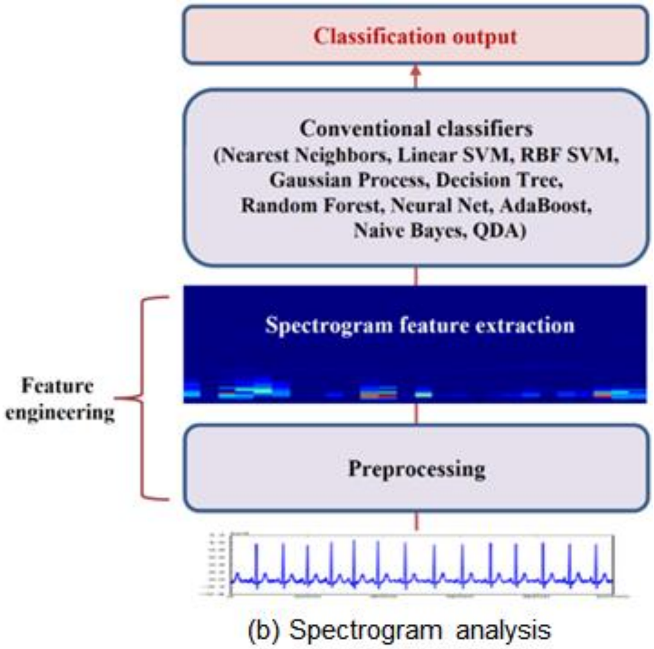

Classification output
Conventional classifiers
(Nearest Neighbors, Linear SVM, RBF SVM,
Gaussian Process, Decision Tree,
Random Forest, Neural Net, AdaBoost,
Naive Bayes, QDA)
Spectrogram feature extraction
Feature engineering
Preprocessing
(b) Spectrogram analysis

Figure 5.4 Conventional ECG anaysis methods

5.2.3 The Initial Setup of the Deep Learning Architecture

As mentioned earlier, most of the conventional data analytics for the stress monitoring using physiological signals required prior knowledge-based feature extraction. The main purpose of this study is to extract hidden feature representations using 1D convolutional neural networks (CNNs) by training the whole process from raw signal input to class label outputs in the manner of seamless end-to-end learning [LeCun2015] [LeCun2010] [Shi2017].

In the early stage of this study, the initial setup of the deep learning model was designed by deeply stacking the convolution and pooling operations in three layers (see Figure 5.6). In this model, three layers of CNNs were built with 50, 30, and 10 channel outputs, respectively. The convolution filter length of the first CNNs layer was set as 500 points (0.5-sec) to cover the particular shapes of ECG signal empirically. This structure and its parameters were decided and tuned empirically based on the previous studies on designing deep learning models [Cui2016][Yang2015].

In addition, batch normalization and drop-out layers were included to prevent overfitting, resulting in improved performance. The batch normalization was performed in mini batch units, known to prevent exploding/vanishing of parameters when gradient descent is used [Ioffe 2015]. Dropout is a regularization technique to prevent the overfitting problem of the neural network on training data. The term "dropout" refers to dropping out units (both hidden and visible) in the neural network, by which the learning is performed in the form of randomly switching off a certain node. In other words, it has the effect of implementing multiple structures of the neural networks, which can yield performance with their ensemble learning [Srivastava2014]. The loss function was the categorical cross entropy and the Adadelta algorithm was used as its optimizer. All learning parameters used the default setting according to the literature (learning-rate = 1.0, decay = 0.0) [Zeiler2012].

After the CNNs layers, the fully connected structure like Multi-layer Perceptrons (MLP) was used to derive the final output labels.

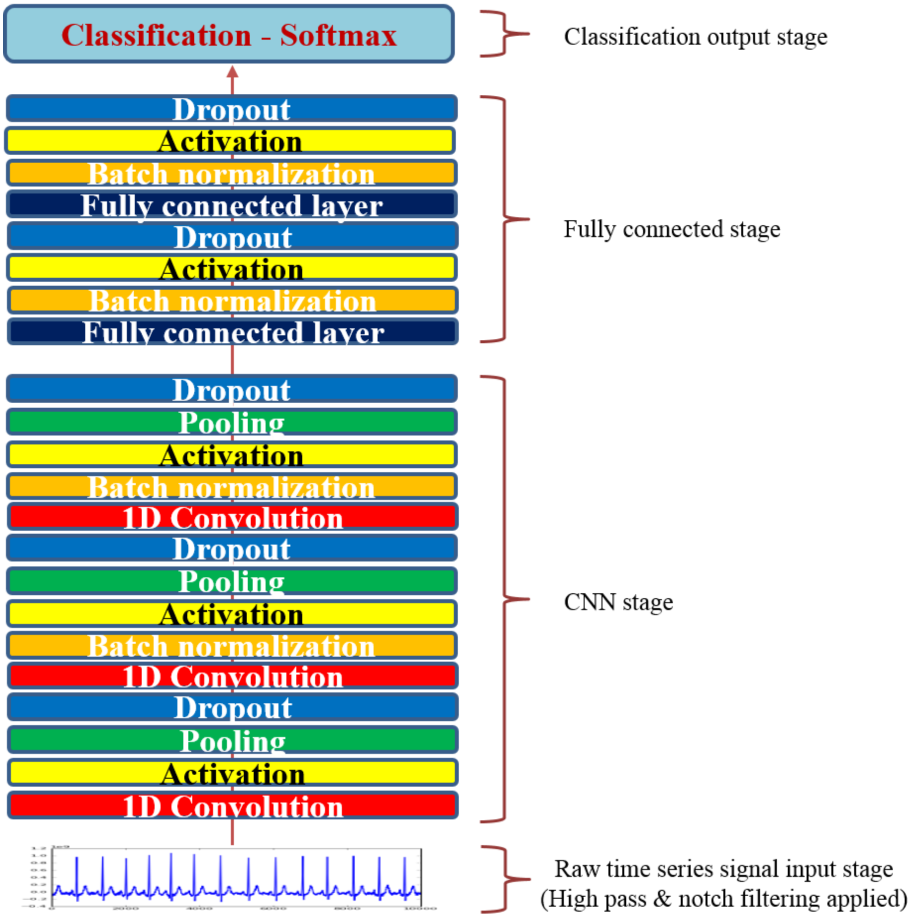


Figure 5.5 Initial setup empirical deep learning architecture

After the initial setup of the deep learning model with heuristically defined parameters, its performance was compared with those of the conventional HRV & spectrogram methods. In this benchmark test, the model outperformed the conventional methods as can be seen in Table 5.4.

Table 5.4 The performance comparison between the initial setup empirical deep learning model and conventional methods (HRV-1, HRV-4 and Spectrogram)

Accuracy	**HRV-1**	**HRV-4**	**Spectrogram**	**Initial setup empirical deep learning model**
Case1	0.6796	0.7105	0.6669	0.7933
Case2	0.5676	0.6307	0.6069	0.6625

5.2.4 The Deep ECGNet

Although the initial setup of the deep model with empirically defined parameters had already outperformed the conventional method, a systemic approach to optimize the parameters while considering the fundamentals of the CNNs algorithm was crucial to maximize the performance of our proposed model.

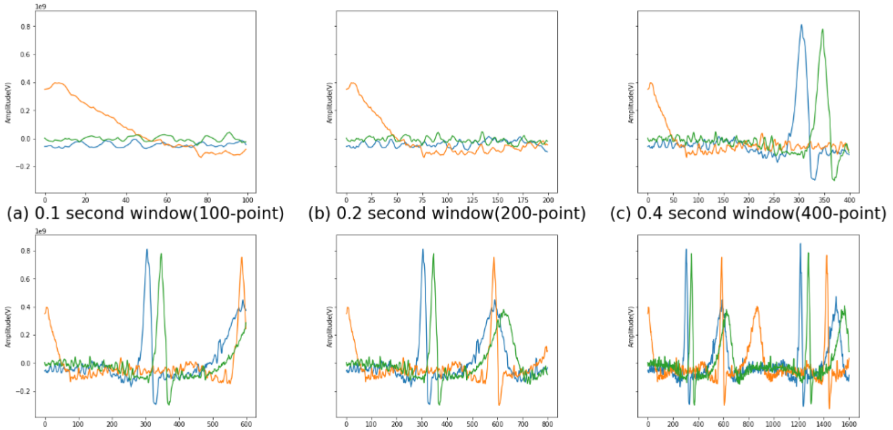
In our empirical deep learning model, phases of ECG peaks according to a partial segment of ECG signal were not considered in depth. However, to improve the performance to classify the stressful conditions, the phase of ECG peak should be considered because the referenced ground line is needed to capture the differences of ECG signal characteristics.

Peak detection and alignment algorithms to adjust peaks of signals need an additional preprocessing procedure. However, performing this preprocessing process does not fit the basic idea of deep learning that a feature can be generated by itself to recognize a pattern.

CNNs is known to be the most powerful algorithm in the image recognition area, and has an especially clear and strong advantage to deal with variances of data such as variations in the location of images. For example, in the MNIST hand written digits recognition problem, CNNs was able to recognize the digits with only a few errors even though placement, scale and rotation of the digits were changed [Simard2003]. This was due to the pooling operation, able to extract specific area features by subsampling [LeCun2012][LeCun2015]. Inspired by this characteristic of the pooling operation, it was applied to 1D ECG signals.

For 1D time series signal data, the subsampling of the pooling operation with a specific duration is able to overcome the peak phase difference problems, because the features of a signal are inevitably extracted in a given pooling window regardless of the peak point.

Therefore, the model could capture the one cycle feature of a signal without considering any preprocessing if the pooling length was set as one cycle of the ECG signal. As can be seen in Figure 5.6, six figures display typical ECG patterns of randomly picked three ECG signals from the dataset including P, Q, R, S and T waves in various length windows. In the figures, only the 0.8-sec window case (Figure 5.6(e)) could contain one whole shape of the three signals within a one segment window. In this figure, the 0.8-sec window was matched with the average heart beat duration (0.7849 sec) calculated using the ECG dataset from Table 5.5.



(a) 0.1 second window(100-point) (b) 0.2 second window(200-point) (c) 0.4 second window(400-point)

(d) 0.5 second window(500-point) (e) 0.8 second window(800-point) (f) 1.6 second window(1600-point)

Figure 5.6 Raw ECG signal plots corresponding to segment time windows

Table 5.5 Average heart rates (Heart beat duration)

Heart rate (sec, Mean/STD)	**Stress**	**Resting**	**Average**	**Total average**
Case1	0.8095/0.1085	0.7944/0.1121	0.8028/0.1104	**0.7849/0.1271**
Case2	0.7640/0.1420	0.7467/0.1492	0.7555/0.1458	

Based on this, we also assumed that the one cycle filter length of convolution operation would be effective for the same reason as with the pooling operation. The convolution and pooling operations were set to a one cycle filter length and pooling length respectively, which made the CNNs able to capture the ECG characteristics with the whole complete shape information of the signal.

To verify this assumption, various experiments were conducted corresponding to the various convolution filter and pooling lengths. These experimental results are shown in the result section.

And although a CNNs architecture usually uses a fully connected layer to the output[LeCun2015][LeCun2010], a RNNs structure with LSTM was adopted for the proposed model considering the sequential patterns of the ECG time series. To use a RNNs model in the final layer, it is necessary to preserve the sequential time order of the CNNs output representation of the ECG signals.

As shown in Figure 5.7, the proposed simple model architecture based on the hypothesis was designed to compare it with the initial setup of the deep learning model. This control model has only one CNNs stage which could produce the feature representation with a one cycle filter length and pooling length, and also has two RNNs layers to generate feature vectors using sequential features extracted by CNNs stage.

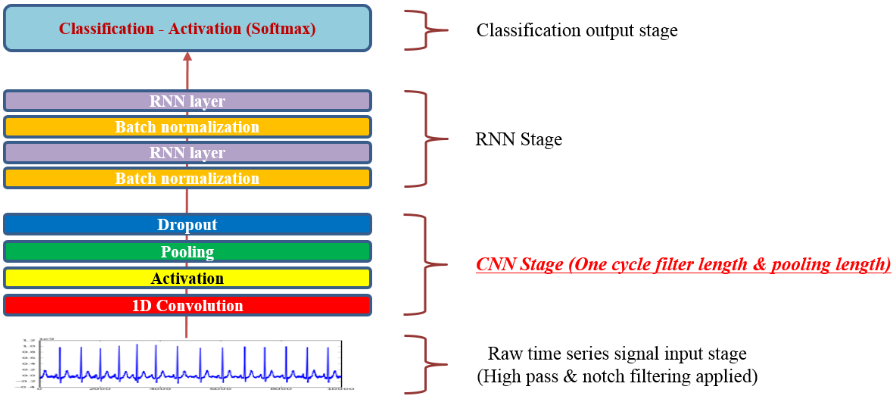

Classification - Activation (Softmax)
Classification output stage
RNN layer
Batch normalization
RNN layer
Batch normalization
RNN Stage
Dropout
Pooling
Activation
1D Convolution
CNN Stage (One cycle filter length & pooling length)
Raw time series signal input stage
(High pass & notch filtering applied)

Figure 5.7 The proposed optimal RCNN deep learning architecture: Deep ECGNet

In order to analyze the performance of the entire deep learning architecture, additional comparative architecture models were also constructed in Figure 5.8.

Figure 5.8(a) is the initial setup of the deep learning model with RNNs layers instead of fully connected layers, Figure 5.8(b) is the general deep CNNs model version of the Deep ECGNet which has the fully connected layers instead of RNNs layers, and Figure 5.8(c) is the proposed reference control model. The proposed model is denoted as “Deep ECGNet”.

As can be seen in Table 5.5, the Deep ECGNet requires less hardware resource and has faster execution time than those of the initial setup of the deep learning model.

For each model, comparative experiments, in terms of accuracy, were conducted on both Case1 and Case2.

All experiments of our studies were conducted with a 5-fold cross-validation method with 60% training, 20% validation and 20% test sets, and on a 3.30GHz Intel Core i5 processor with 64GB of RAM and two NVIDIA TiTAN Xp GPU with Ubuntu Linux v14.04.5. And the scikit-learn v0.18.1 library and the KERAS v2.0.2 were used for conventional classification and deep learning algorithms, respectively.

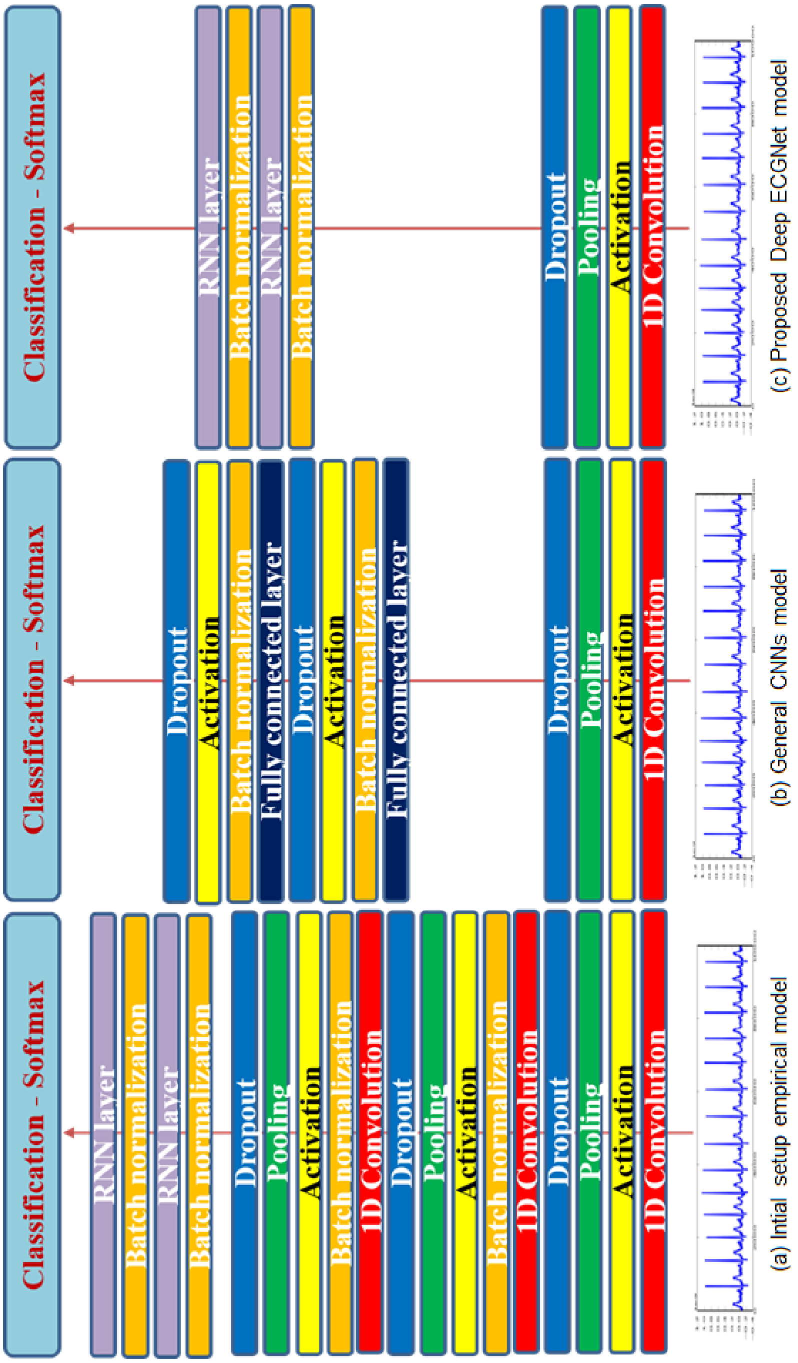


Figure 5.8 The various deep learning architectures

Table 5.6 The usage of resources and runtimes according to model structures

Structure	**3ConvLayer 500Pooling 2RNN (Figure 5.10(a))**	**3ConvLayer 800Pooling 2RNN (Figure 5.10(a))**	**1ConvLayer 800Pooling 2FullyConnected (Figure 5.10(b))**	**1ConvLayer 800Pooling 2RNN (Figure 5.10(c))**
# of parameters	49,092	49,092	43,724	39,772
Memory usage (byte)	4,752,992	4,550,400	3,802,856	3,805,096
Runtime(sec) : Case1	19,782	19,604	16,733	17,714
Runtime(sec) : Case2	13,674	13,442	11,291	12,115

5.3 Experimental Results

The main purpose of this study is to classify the stressful and resting states corresponding to the ECG responses to stress eliciting stimuli. This binary classification problem was analyzed using the proposed deep learning methods and conventional methods as benchmark tests.

In order to evaluate the performance using conventional features, 10 conventional machine learning algorithms were used as classifiers. As shown in Table 5.7, HRV-1 and HRV-4 yielded an average classification accuracy of 67.96% and 71.05% for Case1, and of 56.76% and 63.07% for Case2. The highest classification result for Case1, 72.90 % with HRV-4, was obtained using the Random Forest algorithm, and the highest classification result for Case2 was 66.64 % using the MLP algorithm. As shown in Table 5.8, the average accuracies using spectrogram features were 58.71% for Case1 and 54.83% for

Case2, respectively. The maximum accuracy for Case1, 66.69 %, was obtained using the Random Forest algorithm and the maximum accuracy for Case2 was 60.69% using the Naïve Bayes algorithm.

Table 5.7 Conventional method results accuracies – HRV parameters

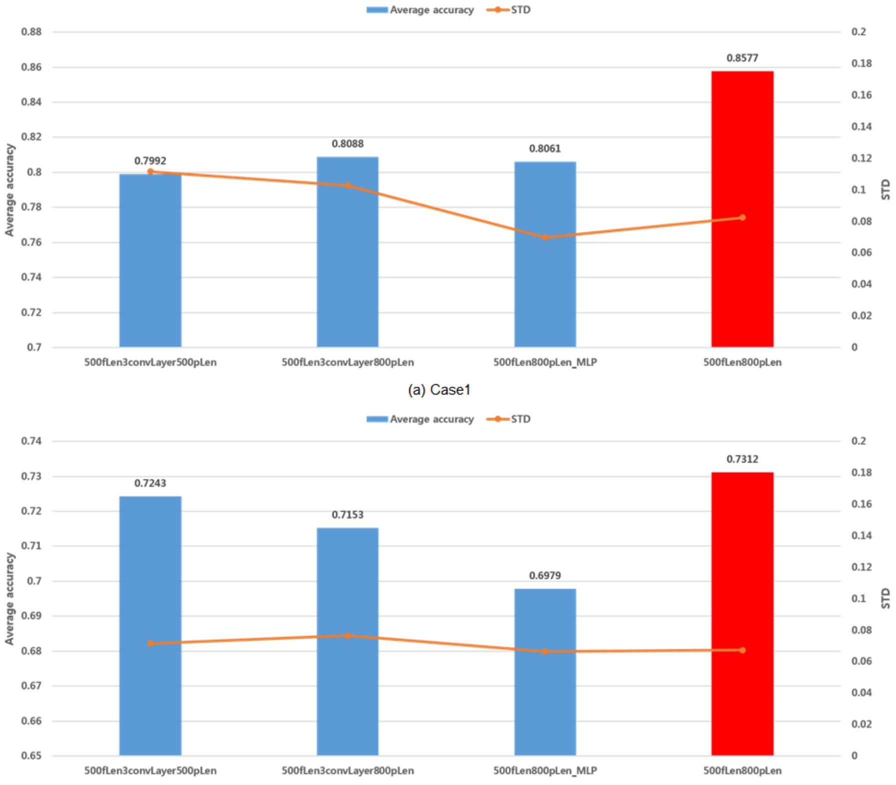
Algorithms	Case1		Case2	
	HRV-1	HRV-4	HRV-1	HRV-4
Nearest neighbors	0.6807	0.7108	0.5646	0.6389
Linear SVM	0.6801	0.6937	0.5660	0.6505
RBF SVM	***0.6949***	0.7239	0.5583	0.6225
Gaussian process	0.6920	0.7261	0.5597	0.6447
Decision tree	0.6591	0.6875	0.5590	0.6082
Random forest	0.6710	***0.7290***	0.5708	0.6220
MLP	0.6813	0.7188	0.5757	***0.6664***
AdaBoost	0.6528	0.6994	0.5715	0.5730
Naïve Bayes	0.6926	0.7023	0.5743	0.6288
QDA	0.6915	0.7131	***0.5757***	0.6521
Average accuracy	0.6796	0.7105	0.5676	0.6307
STD	0.0146	0.0143	0.0069	0.0266

Table 5.8 Conventional method results accuracies – Spectrogram

Algorithms	Case1	Case2
Nearest neighbors	0.6259	0.5913
Linear SVM	0.5484	0.5700
RBF SVM	0.5484	0.5000
Gaussian process	0.5559	0.5000
Decision tree	0.5934	0.5344
Random forest	***0.6669***	0.5363
MLP	0.5484	0.5619
AdaBoost	0.6298	0.5587
Naïve Bayes	0.6091	***0.6069***
QDA	0.5448	0.5231
Average accuracy	0.5871	0.5483
STD	0.0441	0.0360

Based on the hypothesis, the deep learning architecture experiments were conducted with the assumed optimal convolution filter and pooling size. (Where, the ‘fLen’ means convolution filter length, and ‘pLen’ means pooling operator length.)

In Figure 5.9, the first two results (500fLen3convLayer500pLen and 500fLen3convLayer800pLen) are the initial setups of the deep learning network with different pooling lengths (Figure 5.8(a)). The third result (500fLen800pLen _MLP) is the model having 2 fully connected layers instead of RNNs layers (Figure 5.8(b)). The final result is the proposed Deep ECGNet (Figure 5.8(c)). As can be seen in Figure 5.9, our proposed Deep ECGNet model (Figure 5.8(c)) showed the best results (85.77% and 73.12%), better than those of the other deep learning architectures.



(a) Case1

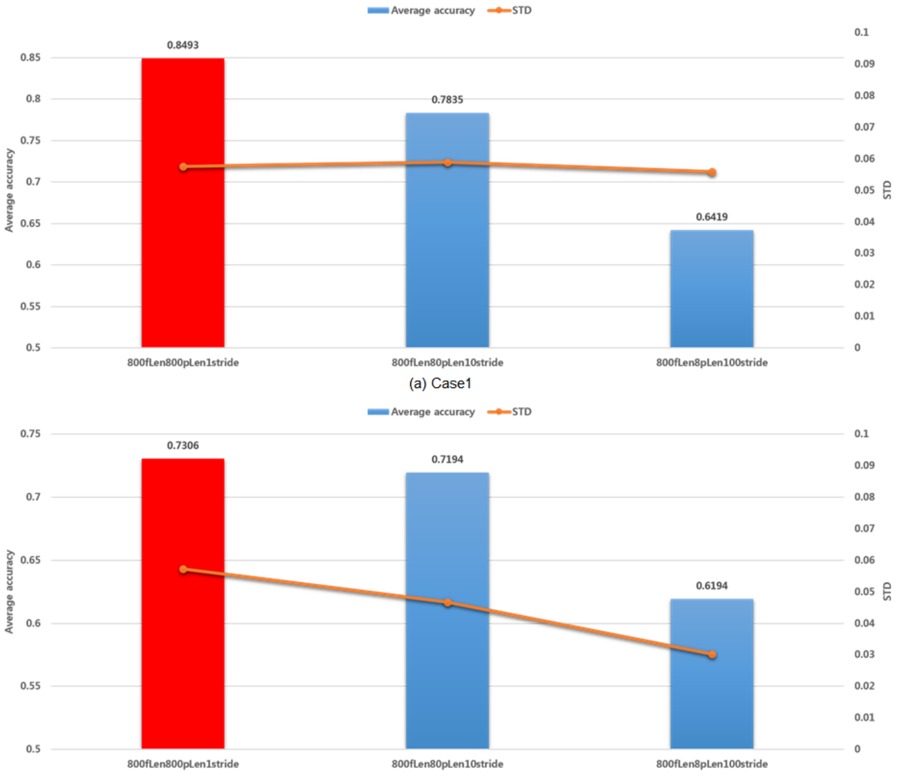
(b) Case2

Figure 5.9 The experimental results corresponding to various deep learning architectures

In order to reduce dimensions of data, a stride size adjustment can be considered in the CNNs algorithm as well as the pooling operation. The stride size in CNNs is the step size of convolution operation from the current to the next convolution operation. Default setting is a unit stride that the convolution operation of every sliding moves in one data point. Therefore, for example, if the stride size is set to 10, it will give the 1/10 sampling effect to data. For reducing the training time, this experiment was conducted because the convolution operation was a dominant factor for running time of the training process. Figure 5.10 shows the experimental results of this.

Experimental setups for three cases were made to yield the same sequence feature length after the convolution layers as the same as that of the previous 800-point pooling case. In Figure 5.10, the first one ‘800fLen800pLen1stride’ was the original model, and the second one ‘800fLen80pLen10stride’ was that having 10-stride and 80-point pooling. The final one ‘800fLen 8pLen100stride’ was that having 100-stride and 8-point pooling. All these three models generated the same length sequence data after the CNNs stage.

However, as you can see in Figure 5.10, as the stride size was increased, the accuracy was decreased. This result infers that the amount of information is decreased as stride is increased. In other words, if the stride size is set to 10, the number of features after the convolution operation will be reduced 1/10 compared to that of the unit stride case. Therefore, if the hardware resource and time for training are enough, the unit stride size would be recommendible based on these result. However, longer stride size could be applied in the initial setup of the structural optimization process, because the running time is faster than that of a unit stride case because of the reduced multiplication operations of convolution operation as shown in Table 5.9.



(a) Case1

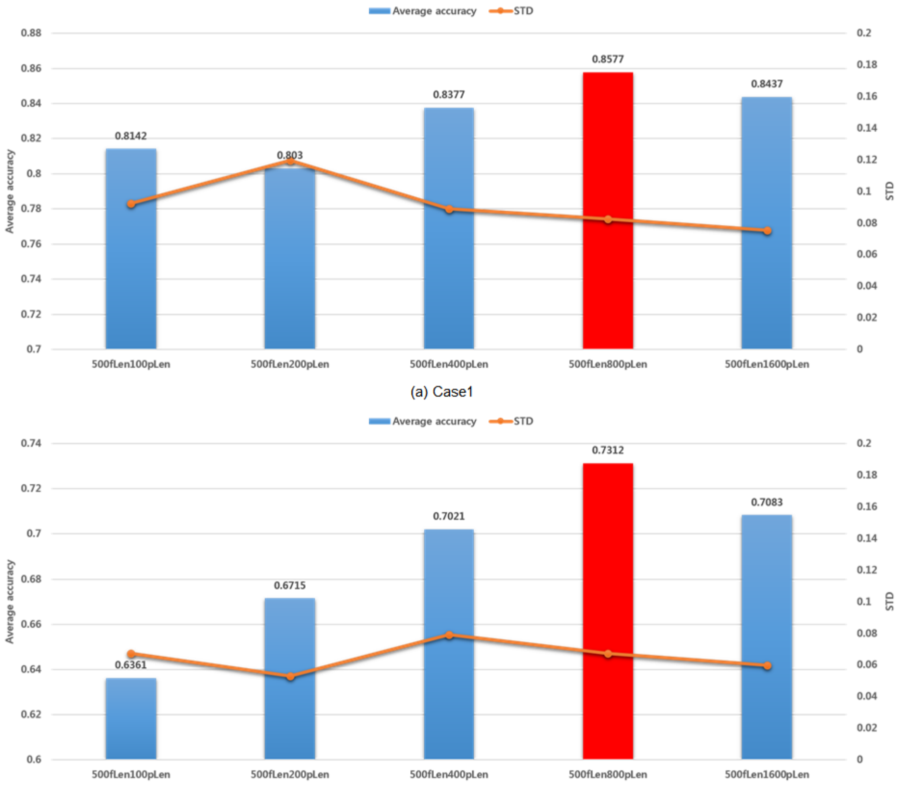
(b) Case2

Figure 5.10 The experimental results corresponding to stride sizes

Table 5.9 Runtime comparison results according to stride sizes

Runtime (sec)	Unit stride	10-stride	100-stride
Case1	17,714	11,348	8,597
Case2	11,136	6,140	5,197

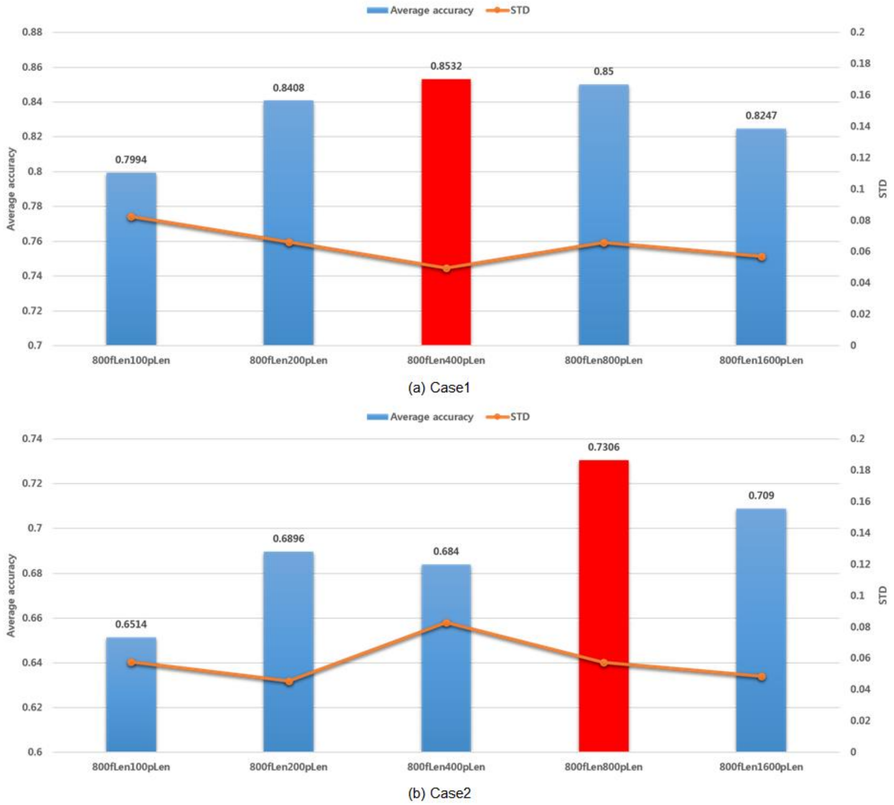
From the constructed optimal RCNN architecture based on the hypothesis about one cycle length for convolution filter length and pooling length and the results in Figure 5.9, several experiments were also conducted to find the optimal parameters. First of all, optimal pooling length experiments were conducted with fixed convolution filter length. As you can see in the Figure 5.11 and 5.12, each fixed convolution filter length was 500-point and 800-point, respectively, because the convolution filter length of the first initial setup of the deep learning model was 500-point and one-cycle length of ECG signal was 800-point corresponding to Table 5.5 (Average heart beat duration: 0.7849 seconds, 800-point in 1000 Hz sampling rate). As you can see in the graphs, experimental results showed the best classification accuracies when pooling length was set to 800-point. The only '800fLen400pLen' of Figure 5.12(a) Case1 showed slightly better performance than that of '800fLen800pLen', but the difference was negligible. Based on these results, the established hypothesis that the optimal pooling operation length was one cycle of ECG signal could be accepted.



(a) Case1

(b) Case2

Figure 5.11 The pooling length experimental results corresponding to various pooling sizes with fixed 500-point convolution filter length (For example, where '800conv800pooling' means 800-point convolution filter length and 800-point pooling length.)



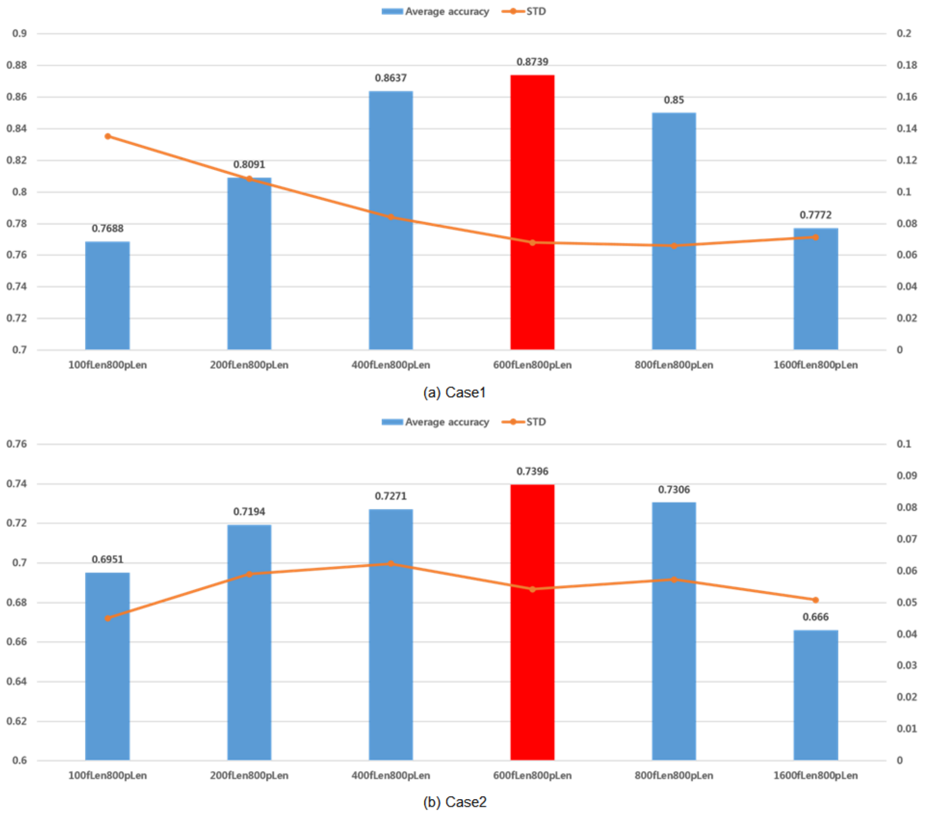
(a) Case1

(b) Case2

Figure 5.12 The pooling length experimental results corresponding to various pooling sizes with fixed convolution filter length(800-point)

Similar to the pooling length experiments, convolution filter length experiments were also conducted. For this experiment, the optimal fixed 800-point pooling length based on the results in Figure 5.11 and 5.12 were set for all optimal convolution filter experiments. At the beginning of the experiment, the filter lengths were set to 100, 200, 400, 800 and 1600 points in the same manner as the pooling length experiment. However, as you can see in Figure 5.13, both Case1 and the Case2 showed the best performances (86.37% and 73.06%) when 400-point and 800-point convolution filters were applied, respectively. So, we added an experiment of 600-point filter length between

400-point and 800-point, and finally, both cases showed the best performances (87.39% and 73.96%) with the 600-point convolution filter.



(a) Case1

(b) Case2

Figure 5.13 The convolution filter length experimental results corresponding to various convolution filter sizes with fixed pooling length (800-point)

The reason why it was showing the best result at 600-point is analyzed as follows.

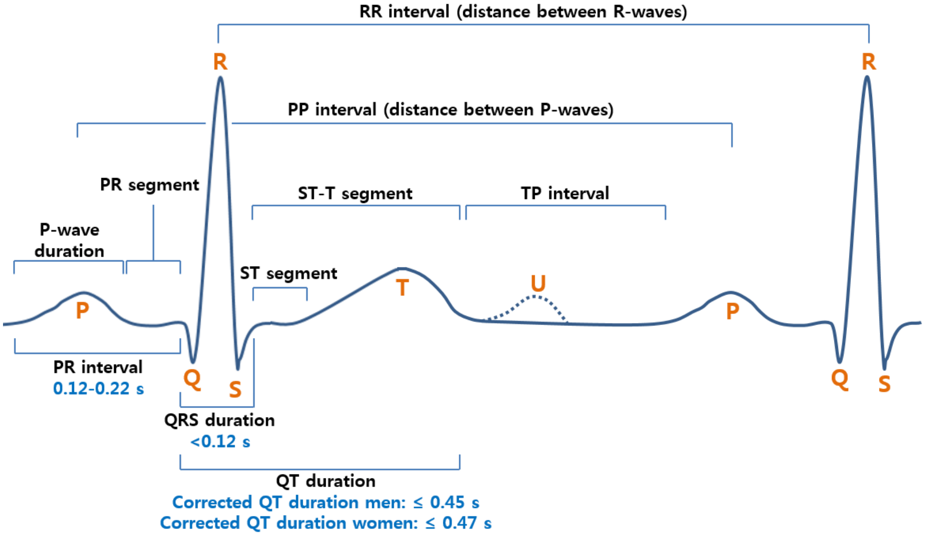
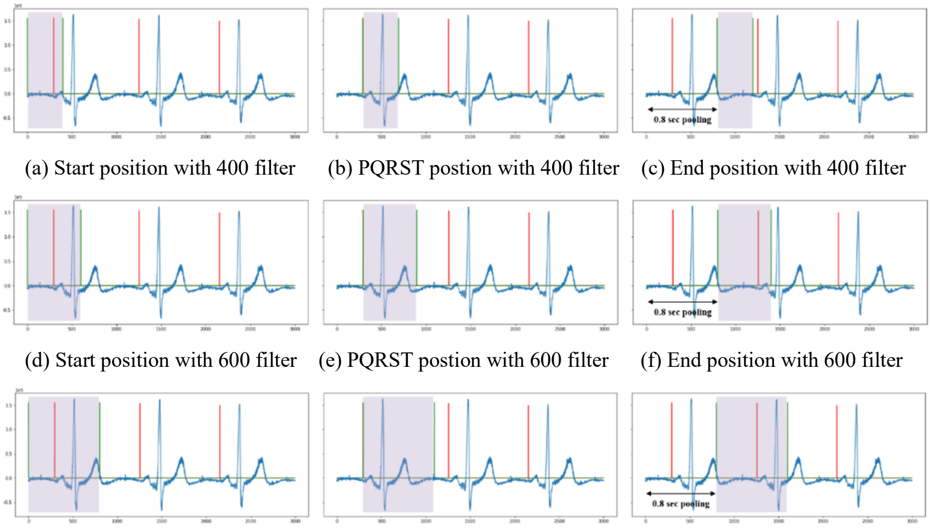

RR interval (distance between R-waves)
R
R
PP interval (distance between P-waves)
PR segment
ST-T segment
TP interval
P-wave duration
ST segment
T
U
P
P
PR interval
0.12-0.22 s
Q
S
Q
S
QRS duration
<0.12 s
QT duration
Corrected QT duration men: ≤ 0.45 s
Corrected QT duration women: ≤ 0.47 s

Figure 5.14 The ECG curve with its most common waveforms. Important intervals and points of measurement are depicted. ECG interpretation requres knowledged of these waves and intervals.

The ECG signal has inherent waveform characteristics shown in Figure 5.14. Therefore, it is very important to analyze the characteristics of all these P, Q,R,S and T waveforms [ecgwaves2017]. From the inherent characteristics of these ECG P, Q, R, S and T waveforms, we were able to find out why the 600-point convolution operation showed the best performance. To perform the 800-point pooling operation, the convolution operation must be performed 800 times, moving the convolution filter. Figure 5.15 shows the data region (purple box) covered by the filter from the start position to the end position of convolution operation according to the convolution filter lengths. For the 400-point filter, there is no interference with the next signal as shown in Figure 5.15(c), but it can not cover the entire P, Q, R, S and T(in this case, T-wave) of the ECG signal as shown in Figure 5.15(b). For the 800-point filter, it covers the entire P, Q, R, S and T regions as shown in Figure 5.15(h). However, there is a large interference with the next signal as shown in Figure 5.15(i). The 600-point filter shows the best performance in Figure 5.13, the entire P, Q, R, S and T regions are covered as shown in Figure 5.15(e), and there is a less interference with the next signal as shown in Figure 5.15(f). Therefore, the 600-point convolution filter length was the optimal filter length, and moreover, we were able to find out that this 600-point (0.6 sec) filter length was consitent with the normal PQRST duration (PR interval + QT duration = 0.57~0.67 sec: 570~670 points in 1KHz sampling rate) from the Figure 5.14.



(a) Start position with 400 filter (b) PQRST postion with 400 filter (c) End position with 400 filter

(d) Start position with 600 filter (e) PQRST postion with 600 filter (f) End position with 600 filter

(g) Start position with 800 filter (h) PQRST postion with 800 filter (i) End position with 800 filter

Figure 5.15 The coverages of convolution operation within ECG heart beat duration (0.8 sec) with 0.4sec (a),(b),(c), 0.6 sec(d),(e),(f), 0.8sec (g),(h),(i) width convolution filters, respectively. (Where, the red line is the starting point of ECG P-wave, and the purple box is the coverage region of convolution filter.)

In addition to the results in Figure 5.15, you can see that the proposed setup of parameters is able to capture the PQRST waveforms with minimum interference in the convolution operatoin region when 800-pooling is applied regardless of the segment window positions as shown in Figure 5.16.

As a result, the specific combination of convolutiona filter (0.6 sec) and pooling (0.8 sec) lengths is able to extract characteristics of one ECG waveform regardless of the signal segment window position without any feature engineering methods.

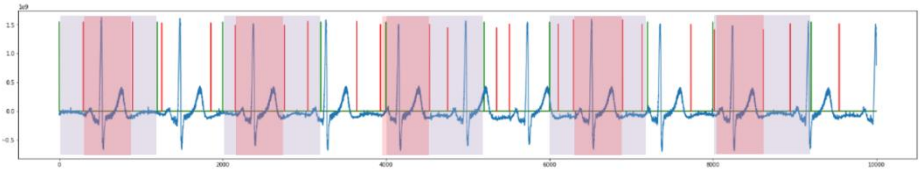
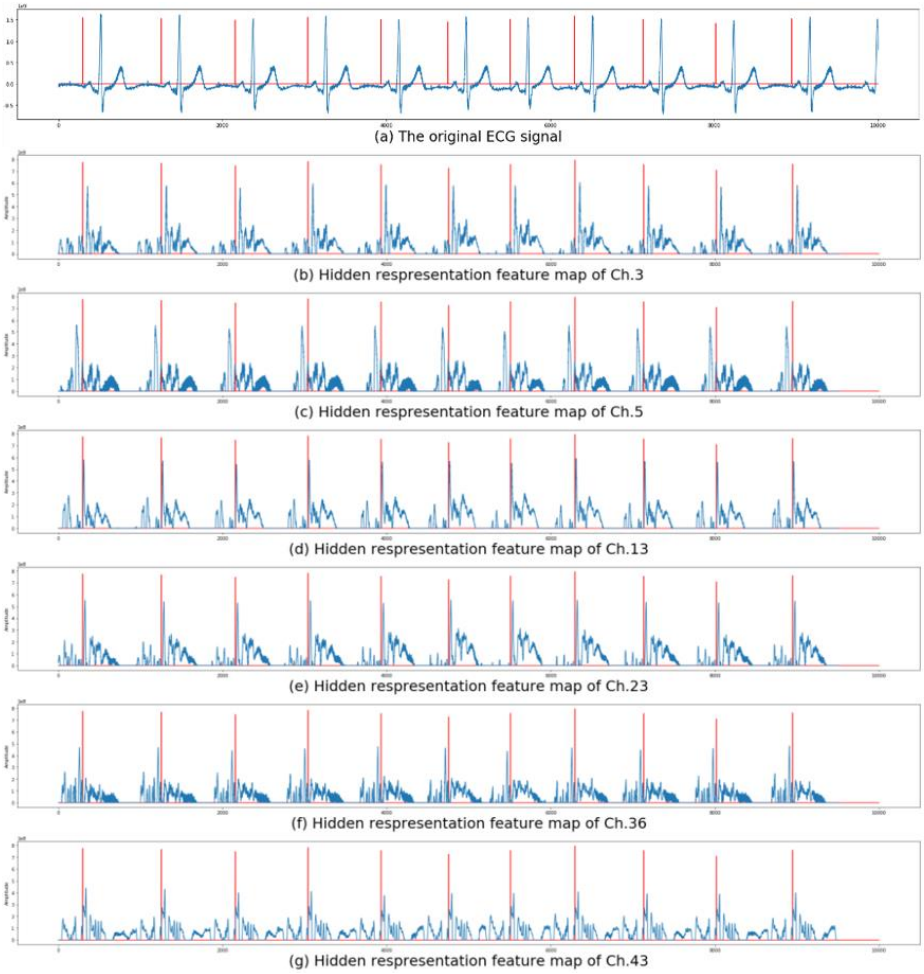


Figure 5.16 The captured PQRST waveforms in the convolution operatoin region with 800-pooling

(Where, the red box is the PQRST regions of ECG signal and the purple box is the convolution operation region when 800-pooling applied.)

To validate whether the trained convolution layer generates specific feature patterns, we visualized the representation of the hidden layer after convolution and the activation function layer (ReLU) as shown in Figure 5.17. For the Deep ECGNet, the convolution layer output was set to generate 50 channel feature maps and we could see specific spiky patterns around the starting point of the origianl ECG P-wave (red line) in several channels as shown in Figure 5.17(b)-(g). This result means that the trained convolution filter worked properly as a kind of masking filter for the P, Q, R, S and T waveforms of ECG signals.



(a) The original ECG signal

(b) Hidden respresentation feature map of Ch.3

(c) Hidden respresentation feature map of Ch.5

(d) Hidden respresentation feature map of Ch.13

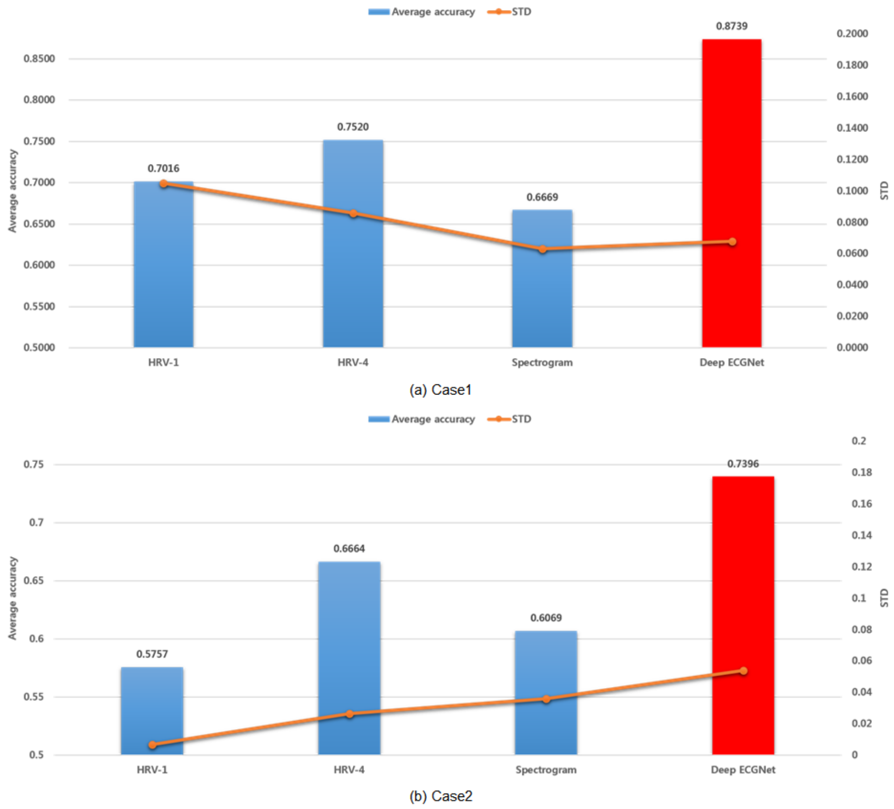
(e) Hidden respresentation feature map of Ch.23

(f) Hidden respresentation feature map of Ch.36

(g) Hidden respresentation feature map of Ch.43

Figure 5.17 Visualizations for the hidden layer representation exmples after convolution & activation function layer (ReLU) (Where, the red line is the starting point of ECG P-wave.)

Lastly, as shown in Figure 5.18, it was confirmed that the proposed deep learning framework showed the highest stress determination performance compared to the conventional stress analysis methods. For the proposed optimal RCNN deep learning model, the result showed the highest accuracy of 87.39% for Case1 and 73.96% for Case2, respectively. These results indicate the performance improvements of 16.22% for Case 1 and 10.98% for Case 2, respectively, as compared to the HRV-4 method.



(a) Case1

(b) Case2

Figure 5.18 The final experimental results compared with conventional ECG analysis methods

Figure 5.19 shows the procedure of the feature extraction sequence of the Deep ECGNet. As you can see on the plot in Figure 5.19, specific overlapping different patterns are observed after the final RNNs stage corresponding to the stress (top left plot) and resting states (top right plot), respectively.

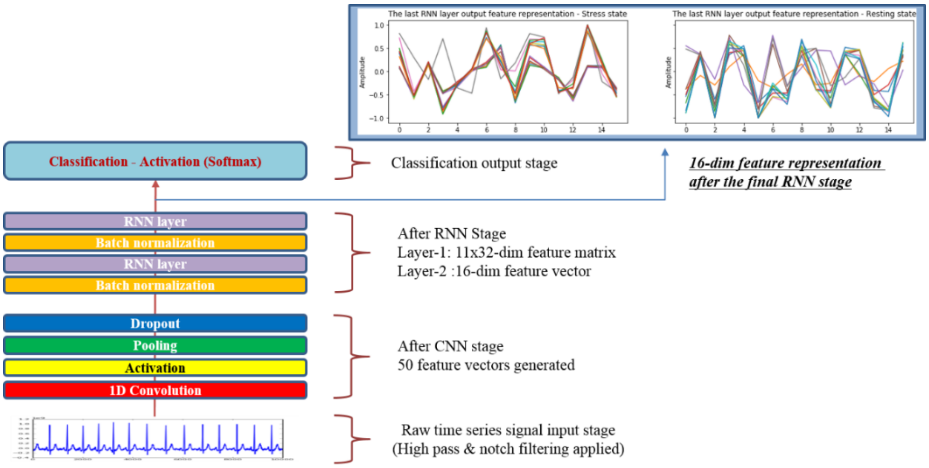

The last RNN layer output feature representation - Stress state
The last RNN layer output feature representation - Resting state
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
12
14
Classification - Activation (Softmax)
Classification output stage
16-dim feature representation after the final RNN stage
RNN layer
Batch normalization
RNN layer
Batch normalization
After RNN Stage
Layer-1: 11x32-dim feature matrix
Layer-2 :16-dim feature vector
Dropout
Pooling
Activation
1D Convolution
After CNN stage
50 feature vectors generated
Raw time series signal input stage
(High pass & notch filtering applied)

Figure 5.19 The procedure of the proposed system to generate optimal features

5.4 Summary

This study proposes an optimal deep learning framework to analyze ECG signals for monitoring mental stress of humans without conventional feature engineering. Another aspect is the theoretical study on how to design the deep learning structure for data with periodicity.

The experimental results demonstrated that the proposed method produced superior performance compared to the conventional methods. In particular, there have been a few numbers of significant HRV parameters for an ultra short-term 10-sec ECG signal according to [Baek2015], the method of using HRV parameters could only use limited information to estimate the mental stress. In addition, the 10-sec short-time window could not provide enough resolution in frequency, lacking sufficient spectral information from the spectrogram features.

On the other hand, the proposed deep learning model could successfully extract the basic feature characteristics of ECG signals for the determination of the stress conditions through 1-D CNNs & RNNs stages. Its whole training process could determine proper filter coefficients like a masking filter to extract important features of ECG signals.

Through the various experiments, we proposed the optimal deep learning framework for stress recognition with the optimal convolution filter length and pooling length. The key point is that the proposed model is able to extract key features of the exact one cycle ECG signal based on the specific durations. This process is a typical designing process of a deep learning model using prior knowledges of data.

Moreover, RNNs was added to extract the final feature using the extracted sequential features from the CNNs stage. In the RNNs representation, consistent overlapping patterns across trials could be noticed, that is, the RNNs at the end stage worked properly as a pattern decoder (Figure 5.19).

Based on the results, the structure of the network model could be more compact and concise than that of the initial setup of the deeper network. The number of parameters and memory usage were also reduced compared to those of the initial setup of the deeper model. And the proposed model showed even better performance than that of the previous various methods. This means that a very deep structure is not necessary if we know prior knowledge of given data and could apply it to the model in the aspects of performance and hardware resources.

This study could be applicable to other time series signals with periodicity.

For future works, a multi-modal physiological signal deep learning framework for human emotion recognition will be studied using the results from this study.

Chapter 6

Deep Learning Model Design Methodology for Non-periodic Time Series Signals using Optimization Techniques: Deep EEGNet

6.1 Introduction

There has been a recent surge of research into the application deep learning (DL) or deep neural networks (DNNs) in biomedical fields including various physiological signals such as electroencephalogram (EEG), electrocardiogram (ECG), electromyogram (EMG) and so on [Jirayucharoensak2014][Ren2014][Hwang2018]. Ren and Wu designed a deep belief network to extract significant features from EEG responses to motor imagery tasks [Ren2014]. In addition, Jirayucharoensak *et al.* utilized a deep learning network to discover unknown feature correlation between EEG signals recorded during various emotional states [Jirayucharoensak2014]. ECG responses during stressful states were also analyzed using deep learning models for the process of feature extraction and classification simultaneously [Hwang2018]. The deep learning approach has also been applied to EMG and accelerometer data to estimate human limb movement [Xia2018][Seok2018].

Among these various applications, deep learning models were often used as a classifier separating feature inputs extracted by traditional feature engineering [Jirayucharoensak2014][Xia2018]. In addition, Ren *et al.* have also shown that deep learning algorithms can be designed for preprocessing, feature extraction from raw input data [Ren2014]. Therefore, an all-inclusive process including preprocessing and classification could be implemented by designing a DNNs algorithm that serves as an end-to-end model. Conventional feature extraction methods such as Fourier analysis using linear and fixed basis functions are inevitably the next best solution because of the nonlinearity and nonstationarity of physiological signals [Park2013].

From a signal processing perspective, DNNs could generate data-driven basis functions which have weight vectors in the network, using input data without any predefined model. This is more generic and natural when compared to conventional feature engineering. Previously, Hwang *et al.* addressed the end-to-end DNN model from ECG raw data, designing the model by considering the physiological properties of ECG signals. Their end-to-end DNNs model, without any preprocessing outperformed, the classification process based on traditional feature engineering [Hwang2018].

For the development of human computer interface (HCI) system, human-machine interaction is a key issue to be considered, where machines provide services to human, and in return humans provide feedback to the machines in response to the service. Among various types of reactional feedback from humans (e.g. buttons, speech, movement, facial expressions etc.), the genuine human response to the machine service is crucial to improve HCI systems. Research in affective computing is an effort to estimate and model genuine responses in the form of human emotions [Dente2006][Rutkowski2007]. There have been studies to model human affective responses using several kinds of feedback to HCI services. For example, one study investigated speech signals as a method to detect human emotions. This study achieved human-level classification performance [Dellaert1996]. Another study also used facial expressions to monitor emotions with 80-90 % classification accuracy [Chen2003]. In addition, autonomic nervous system (ANS) responses (e.g. changes in skin temperature, galvanic skin response (GSR), respiration and heart rate (HR)) have also been monitored as affective indicators of human emotional states [Scheirer2002][Crosby20014].

Recently, EEG signals have received considerable attention in the research of affective computing. EEG signals are elicited from the central nervous system (CNS) to yield genuine emotional states, even at the unconscious level. The coherence between EEG signals and the affective states of human subjects have been detected in previous studies [Balconi2009][Kislova2009]. When humans receive a stimulus from their environment, ANS firstly processes the stimulus and then send the information to the hypothalamus. The hypothalamus is in charge of incoming/triggering neurological signals from/to their corresponding ANS events, such as the increase/decrease of HR or GSR [Kandel2000]. The stimulus in the hypothalamus is subsequently passed to the amygdala in the subcortical region, where the stimulus information is compared with past experiences and connected to affective responses. These flows of stimulus information explain the relationship between ANS and CNS during affective states, which has been proven by neuro-image studies illustrating the simultaneous increase of the brain activity in cortical (frontal, insular and anterior temporal), subcortical (amygdala, thalamus and hypothalamus) and midbrain regions with GSR and HR during emotional mental tests [Damasio2000][Lane1999]. The primary role of the hypothalamus, matching stimuli with past emotional memories, emphasizes the importance of monitoring the CNS effects, resulting in the crucial role of EEG recording in extracting true emotional states.

Due to the low signal-to-noise ratio (SNR) of EEG signals, spectral analysis in frequency domain has often been applied to EEG studies. Conventionally, EEG signals are filtered into several frequency bands, delta (0.1-4Hz), theta (4-8Hz), alpha (8-13Hz), beta (13-30Hz) and gamma (30-55Hz), using Fourier or wavelet analyses [Park2014]. Features are calculated using the band-passed signals, which are fed into a classifier. However, the preprocessing procedure of the filter is mainly based on predefined basis functions, such as sinusoids for Fourier and mother wavelet for wavelet analyses respectively. This could be an unnatural assumption on the nonlinear and nonstationary characteristics of EEG signals. In addition, the predefined frequency bands of EEG signals are not universally accepted for all kinds of mental activities. There could be other significant oscillations, crossing the predefined frequency bands corresponding to the emotional states.

This study presents a deep neural network algorithm to estimate the natural oscillations or features corresponding to the affective states from raw EEG input signals without any preprocessing method, in a data-driven end-to-end model approach. Until recently, there have been active researches on hyperparameters optimization and structural optimization of neural architectures [Elsken2018][Falkner2018][Li2017]. In particular, the hyperparameters and the structure of the DNNs, such as the number of filters, layers, etc., are also defined in a data-driven manner using the Bayesian optimization approach [Snoek2012]. These fully data-driven DNNs architecture could automatically extract the natural oscillation components in the EEG signals, and could provide a different view or phenomenon in cognitive physiology compared to those in traditional neuroscience.

To demonstrate the performance of the proposed approach, both the experimental dataset created by Kwangwoon University and Koelstra's publicly available DEAP database [Koelstra2012] were utilized in this study.

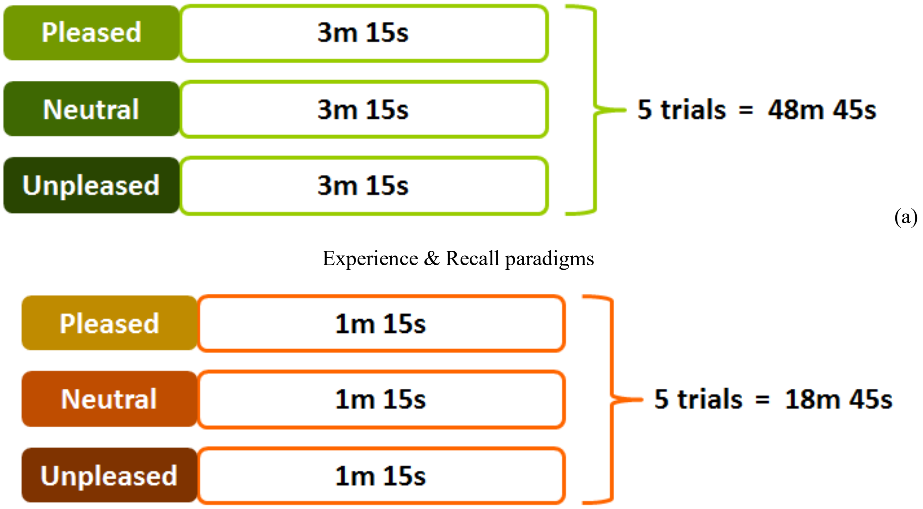
6.2 Materials and Methods

6.2.1 Subjects and Data Acquisition

Emotion Dataset based on Emotional Video Experiments: Case1

Nine healthy male and female subjects were recruited at Kwangwoon University using flyers and posters. The enrolled subjects were in their 20s, ranging in age from 21 to 28. They had not participated in an experiment using any mental/physical emotional task before. The experiments were conducted after every subject submitted the written informed consent. EEG signals were recorded at positions AF3, AF4, FCz, T7, T8, PO3 and PO4 according to the 10-20 system [Jasper1958]. EEG signals were sampled at 256Hz. (Biopac MP36 system, Biopac System Inc. Goleta, CA) As a preprocessing step, a 2000th-order FIR notch filter and a 3000th-order FIR band-pass filter were applied to remove the power line noise around 60 Hz, and keep the frequency components in the range 1.5–45 Hz. The experiment sessions consisted of three emotional paradigms – experience, observation and recall. During the experiment, participants watched video clips designed for different paradigms. The experience paradigm reveals one's emotional state based on his/her previous experience when he/she watches emotional video clips. The observation paradigm reveals one's sympathetic emotional state based on emotional facial expressions of others. Finally, the recall paradigm elicits one's recalled emotional state when a subject is reminded of past emotional memories. All three paradigms have three emotional valance levels – pleased, neutral and unpleased. Five trials were conducted for every paradigm. Figure 6.1 displays the details of the data length corresponding to the paradigms, where the length of experience and recall paradigms were 3 minutes and 15 seconds, while that of the observation paradigm was 1 minute and 15 seconds. Examples of various emotional video clips used in the experiments are displayed in Figure 6.2. In particular, the images of facial expressions show the video clips used for the observation paradigm. A 10 second time window segmentation was applied to make the training/validation/test data.

This study was reviewed and approved by the institutional review board at the Korea National Institute for Bioethics.

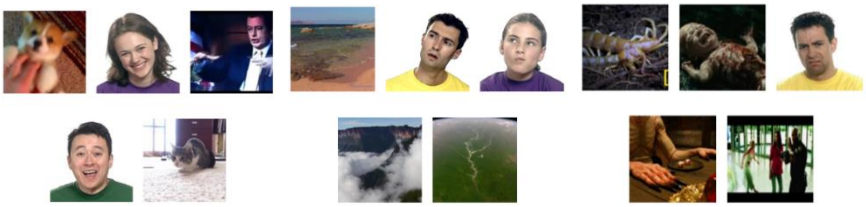


(a)

Experience & Recall paradigms

(b) Observations paradigm

Figure 6.1 Experimental prosesses corresponding to the emotional paradigms



(a) Pleased video clips (b) Neutral video clips (c) Unpleased video clips

Figure 6.2 Examples of the emotional video clips to elicit

DEAP Database: Case2

The DEAP (A Database for Emotion Analysis Using Physiological Signals) database, made by Koelstra *et al*., was used as a benchmark dataset for the performance evaluation of our proposed method [Koelstra2012]. The DEAP database is an open database and contains multi-modal physiological signals on 32 subjects including 32-channel EEG signals. The data were recorded as each participant watched forty 1 minute long excerpts of music videos. Participants rated each video in terms of the various emotional levels. The Self-Assessment Manikins (SAM) scoring method was chosen to express subject's emotion states in the DEAP database. We used the provided preprocessed Python data cubes from the DEAP webpage (128Hz down sampled, removed eye-blinking artefacts and band-pass filtered each channel to 4-45Hz), and a 10 second time window segmentation was also applied to make the training/validation/test data.

6.2.2 Conventional EEG Analysis Methods

Conventionally, frequency band powers were extracted as EEG features. EEG signals are filtered into several frequency bands (e.g., delta (0.1-4Hz), theta (4-8Hz), alpha (8-13Hz), beta (13-30Hz) and gamma (30-55Hz)) using Fourier or wavelet analysis [19]. The power features are then calculated using the band-passed signals, which are fed into a classifier. Both Case1 and Case2 have five power features for all the channels. The classification performance using these conventional frequency power features were compared with that of the proposed deep learning approach. Koelstra *et al*. also used this conventional method to classify human emotional states using their DEAP database [Koelstra2012].

As a benchmark test, human emotional state classifications were tested against ten conventional shallow machine learning algorithms using 5-band power features which were generated by the BioSPPY bio-signal processing library. A 10 second time window segmentation was also applied to make the training/test data. Table 6.1 shows a list of applied conventional machine learning algorithms. The Scikit-learn Python library package was used to apply these various algorithms. All parameters on these algorithms were set to their default values. Figure 6.3 shows the conventional shallow machine learning approaches. All experiments on conventional methods were conducted with a 5-fold cross-validation method with 80% training and 20% test sets for every algorithm.

Table 6.1 A list of conventional shallow machine learning algorithms

Applied conventional algorithms
Nearest Neighbors
Linear SVM
RBF SVM
Gaussian Process
Decision Tree
Random Forest
Multi-Layer Perceptron (MLP)
AdaBoost
Naïve Bayes
Quadratic Discriminant Analysis (QDA)

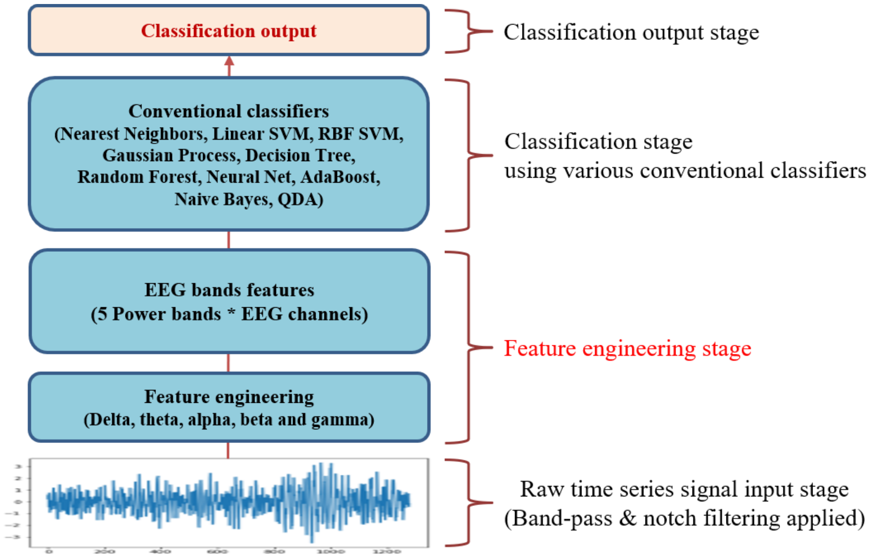

Classification output
Classification output stage
Conventional classifiers
(Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Neural Net, AdaBoost, Naive Bayes, QDA)
Classification stage
using various conventional classifiers
EEG bands features
(5 Power bands * EEG channels)
Feature engineering stage
Feature engineering
(Delta, theta, alpha, beta and gamma)
Raw time series signal input stage
(Band-pass & notch filtering applied)

Figure 6.3 Conventional EEG power bands analysis methods

6.2.3 Basic Deep Learning Units and Optimization Technique

Convolutional Neural Networks (CNNs) were used as a basic unit to make a hidden representation of EEG signals in the whole learning architecture. And LSTM was also used to create features containing sequential information using hidden representation generated from the CNNs layer.

Furthermore, L1/L2 Regularization, Dropout and Batch Normalization techniques were also applied to improve generalization abilities of the proposed model [Ng2004][Srivastava2014][Ioffe2015].

In order to build the optimal Deep Learning model, the Bayesian optimization algorithm was applied to the proposed methodology as an optimization technique. The setting of the Bayesian optimization in this experiment was based on previous findings by Snoek *et al*. [Snoek2012]. It was implemented using the Spearmint Bayesian optimization python library.

6.2.4 Optimization for Deep EEGNet

In the early stages of this study, the initial setup of the deep learning model for EEG signal analysis was inspired by previous research of Deep ECGNet [Hwang2018]. Therefore, the Deep ECGNet model configuration was applied to that of the EEG signal analysis. However, it was not trainable at all for the EEG signals since EEG signals are noisy, non-stationary signals, while ECG signals exhibit predictable periodicity characteristics. As such, EEG signals could not be analyzed using Deep ECGNet methodology. Therefore, we began to look at this problem as a very complex optimization problem and sought ways to deal with it as such.

This design issue could be due to a complex optimization problem which has a highly expensive black box objective function. In order to solve this, Bayesian optimization was applied to configure the optimal deep learning model, Deep EEGNet. The whole optimization process includes not only hyperparameters optimization, but also model structural optimization. Table 6.2 shows a list of hyperparameters and structures to be tuned. Each optimization target has its own type and optimization range.

Table 6.2 Items to be optimized for the improvement of Deep EEGNet performance

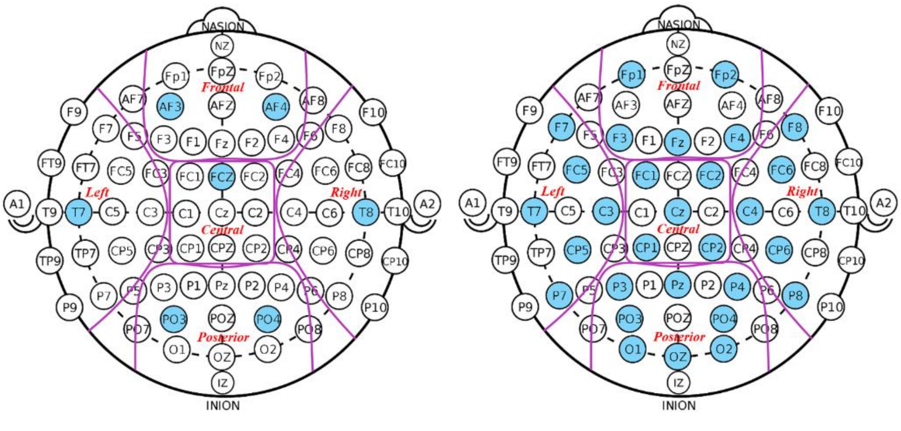
Hyperparameters	**Data type**	**Range (Case 1/Case 2)**
# of CNNs feature maps	Integer	32-64/32-64
CNNs depth	Integer	1-3/1-3
Convolution filter length	Integer	4-256/8-512
Pooling length	Integer	16-256/8-128
RNNs (LSTM) depth	Integer	1-2/1-2
Model structure type	Enumeration	CNNs+MLP/CNNs+RNNs
# of nodes for hidden layer1	Integer	8-64/8-64
# of nodes for hidden layer2	Integer	8-64/8-64
Batch size	Integer	32,64,128/32,64,128
Regularization coefficients(L1/L2)	Float	0.0001-0.005/0.0001-0.005
Early stop epochs	Integer	30-300/30-50
Drop out probability	Float	0.2-0.5/0.2-0.5
Learning rate	Float	0.001-0.1/0.001-0.1
Momentum	Float	0.0-0.2/0.0-0.2
Optimizer type	Enumeration	Adam/RMSprop/SGD
Activation function type	Enumeration	Tanh/Sigmoid/ReLU

In addition to the hyperparameter optimization, the Deep EEGNet design process includes structural optimization. EEG signals are sequential time series data, which are able to take advantage of an RNNs structure with LSTM due to its sequential pattern of feature representations after CNNs layers. Deep EEGNet is designed via two different structures, CNNs+MLP and CNNs+RNNs (CRNNs), with various deep layers.

6.2.5 Deep EEGNet Architectures using the EEG Channel Grouping Scheme

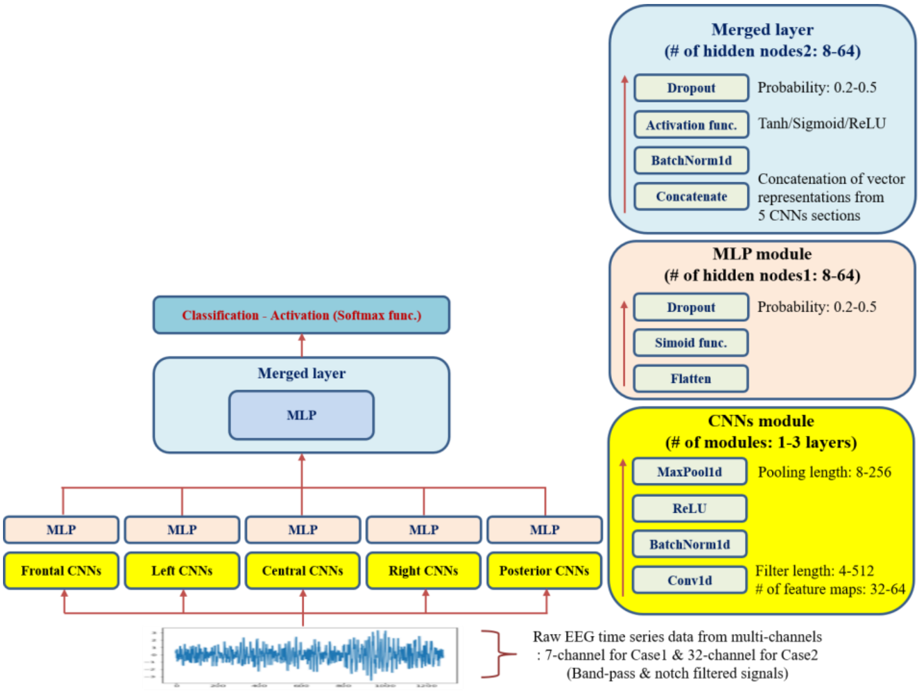
EEG signals are surface electrical activity of the brain, and have functional characteristics of brain localities. Therefore, they need to be analyzed as local groups, not single individual channels, since spatially adjacent channels are able to share similar information to each other. Lainscsek *et al*. studied the effects of local channel groups with the characteristics of local brain activities [Lainscsek2013].

In this experiment, brain areas are divided into 5 groups - frontal, posterior, left, right and central regions. Figure 6.4 illustrates the international 10-20 EEG placement system and the chosen 7 and 32 electrodes are highlighted in blue circles for the Case1 and Case2 experiments [Jasper1958]. All the single channels are grouped into 5 regions and these grouped signals are fed into the Deep EEGNet. The Deep EEGNet has independent CNNs models for each different channel group. These models can be able to generate unique feature representations corresponding to different brain areas.

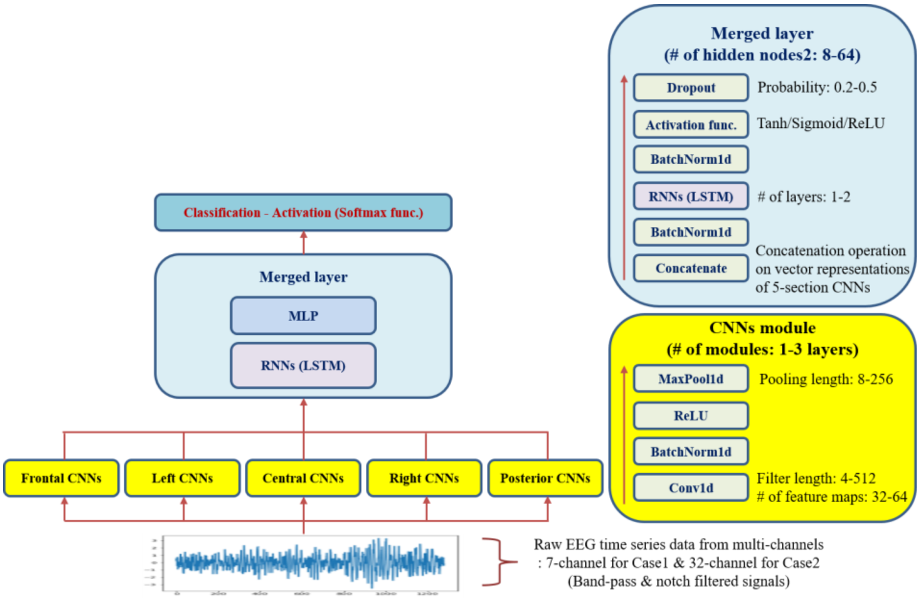


(a) Case1 (b) Case2

Figure 6.4 EEG electrode positions & groups of 7-channel for Case1 and 32-channel for Case2



(a) Deep EEGNet: CNNs+MLP structure and optimization hyperparameters



(b) Deep EEGNet: CNNs+RNNs (CRNNs) structure and optimization hyperparameters

Figure 6.5 Proposed Deep EEGNet architectures

Figure 6.5 illustrates the proposed model architectures regarding the channel groups, which will be optimized using a Bayesian optimization process. As loss functions, the Binary Cross Entropy Loss (BCELoss) function was used for 2-class classification, and the CrossEntropyLoss function was used for 3-class classification. In order to optimize the model architectures, comparative experiments were conducted on both Case1 and Case2 experiments depending on the classification accuracy. All experiments were conducted with a 5-fold cross-validation method with 60% training, 20% validation and 20% test sets using an early stopping mechanism monitoring a validation loss.

Training were conducted on a 3.30GHz Intel Core i5 processor with 64GB of RAM and two NVIDIA TiTAN Xp GPUs with Ubuntu Linux v14.04.5. The Scikit-learn v0.18.1 library, the BioSPPY 0.6.1, the Pytorch v0.3.0.post4 and the Spearmint Bayesian optimization library were utilized to implement the conventional machine learning, extraction of band powers, deep learning algorithms and hyperparameter optimization, respectively.

6.3 Experimental Results

This study tries to recognize human emotional states based on the EEG responses to specific emotional stimuli. All experiments investigated binary/multiclass classification problems among multiple emotional responses during various emotional paradigms. As a benchmark test, ten conventional shallow machine learning methods were compared, and the results are summarized in Table 6.3 and 6.4. These results are averaged accuracies across all subjects and algorithms, and 5-fold cross validations were applied for rigorous testing.

As shown in Table 6.3, Case1 yielded averaged 2-class and 3-class classification accuracies of 58.17% and 41.51%, respectively. The best 2-class classification result, 65.68%, was obtained using the AdaBoost algorithm. The best 3-class classification result, 47.53%, was obtained using the Gaussian Process algorithm. As shown in Table 6.4, Case2 yielded averaged 2-class and 3-class classification accuracies of 54.73% and 43.53% respectively. The best 2-class classification result, 58.33%, and the best 3-class classification result, 48.52%, were both obtained using the RBF SVM algorithm. In both 2-class and 3-class classifications for all cases, types of the SVM or ensemble algorithms showed relatively better performance among conventional shallow machine learning methods.

Table 6.3 Accuracies using conventional methods – Case1

Algorithms	**2-Class (Accuracy / STD)**	**3-Class (Accuracy / STD)**
Nearest Neighbors	0.6160 / 0.1319	0.4284 / 0.1114
Linear SVM	0.5519 / 0.1098	0.3926 / 0.0835
RBF SVM	0.5525 / 0.1096	0.3934 / 0.0842
Gaussian Process	0.5383 / 0.0984	***0.4753 / 0.1434***
Decision Tree	0.6204 / 0.1348	0.4272 / 0.0917
Random Forest	0.6247 / 0.1359	0.4370 / 0.1104
MLP	0.4994 / 0.0149	0.3333 / 0.0000
AdaBoost	***0.6568 / 0.1508***	0.4597 / 0.1214
Naïve Bayes	0.6500 / 0.1507	0.4564 / 0.1253
QDA	0.5074 / 0.0725	0.3477 / 0.0666

Averaged ac curacy/STD	0.5817 / 0.1109	0.4151 / 0.0938

Underscored values represent the best performance algorithm for each case.

Table 6.4 Accuracies using conventional methods – Case2

Algorithms	**2-Class (Accuracy / STD)**	**3-Class (Accuracy / STD)**
Nearest Neighbors	0.5372 / 0.1628	0.4088 / 0.1589
Linear SVM	0.5605 / 0.1318	0.4427 / 0.1230
RBF SVM	***0.5833 / 0.0516***	***0.4947 / 0.0480***
Gaussian Process	0.5313 / 0.1379	0.4470 / 0.0946
Decision Tree	0.5448 / 0.1646	0.4137 / 0.1640
Random Forest	0.5645 / 0.1512	0.4625 / 0.1488
MLP	0.5439 / 0.1527	0.4358 / 0.1568
AdaBoost	0.5535 / 0.1617	0.4436 / 0.1560
Naïve Bayes	0.538 / 0.1623	0.4241 / 0.1601
QDA	0.5164 / 0.1503	0.3804 / 0.1576
Averaged acc uracy/STD	0.5473 / 0.1427	0.4353 / 0.1368

Underscored values represent the best performance algorithm for each case.

As shown in Table 6.5, the deep learning structures and hyperparameters were determined based on the Bayesian optimization results for Case1. For saving optimization time, only 1-fold validation was applied for evaluation of the objective function during the optimization process. Ater finishing the optimization and setting up models, 5-fold cross validation was taken for every subject. All experiments were conducted with trial blind manner for every subject. Table 6.6 shows the performance summarizations of Case1, where the optimized Deep EEGNet yielded 73.21% averaged accuracy for 2-class classification and 57.90% averaged accuracy for 3-class classification. In Table 6.5, the CNNs+MLP structure can be seen as the optimal structure for the Case1 database.

Table 6.5 The optimal hyperparameters & model structure - Case1

Hyperparameters	**2-Class**	**3-Class**
# of CNNs feature maps	43	32
CNNs depth	1-layer	1-layer
Convolution filter length	291	164
Pooling length	16	256
RNNs (LSTM) depth	-	-
Model structure type	CNNs+MLP	CNNs+MLP
# of nodes for hidden layer1	8	20
# of nodes for hidden layer2	8	55
Batch size	32	128
Regularization coefficients(L1/L2)	0.0001/0.000712	0.0001/0.00328
Early stop epochs	199	144
Drop out probability	0.2	0.2
Learning rate	0.001	0.001
Momentum	-	-
Optimizer type	Adam	Adam
Activation function type	Sigmoid	ReLU

Table 6.6 Classification accuracies using Deep EEGNet - Case1

Subjects	2-Class	3-Class
Subject1	0.9056	0.6074
Subject2	0.6889	0.5815
Subject3	0.8000	0.6296
Subject4	0.4778	0.4296
Subject5	0.7611	0.6519
Subject6	0.5944	0.4111
Subject7	0.7389	0.6370
Subject8	0.8722	0.6667
Subject9	0.7500	0.5963
Average	***0.7321***	***0.5790***
STD	0.1624	0.1274

Underscored values represent the averaged performance of Deep EEGNet for each case.

As can be seen in Table 6.7, the deep learning structures and hyperparameters were also determined based on the Bayesian optimization results for Case2. In this case, the optimization process was taken based on the class labels of the Valence level.

Based on the results of Table 6.5 and Table 6.7, common optimization feature points are CNNs depth (1-layer), Structure (CNNs + MLP), Drop-out probability (0.2), Optimizer (Adam) and Activation function (ReLU). These results show that, like Deep ECGNet, CNNs + MLP structure can output optimal performance with proper CNN parameter even if CNN depth is not deep. CNNs + MLP structure shows that its pattern is more effective than sequential characteristic of data. In addition, commonly used opimizer and activation functions are also effective in this application.

Table 6.7 The optimal hyperparameters & model structure - Case2

Hyperparameters	2-Class	3-Class
# of CNNs feature maps	64	32
CNNs depth	2-layer	1-layer
Convolution filter length	4	154
Pooling length	93	256
RNNs (LSTM) depth	-	1-layer
Model structure type	CNNs+MLP	CNNs+RNNs
# of nodes for hidden layer1	53	24
# of nodes for hidden layer2	58	60
Batch size	32	32
Regularization coefficients(L1/L2)	0.000506/0.003362	0.004/0.003122
Early stop epochs	31	40
Drop out probability	0.2	0.3
Learning rate	0.001	0.001
Momentum	-	-
Optimizer type	Adam	Adam
Activation function type	ReLU	ReLU

Table 6.8 Classification accuracies using Deep EEGNet – Case2

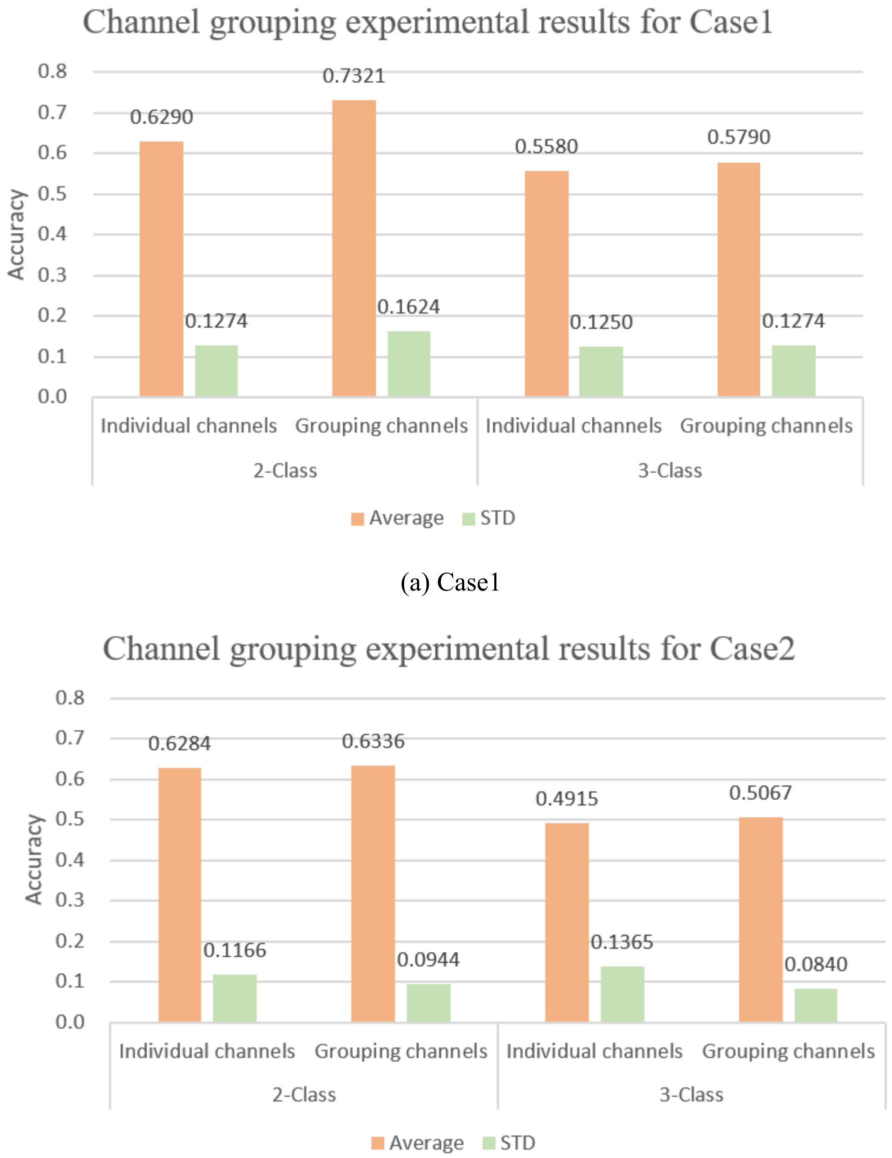
Subjects	**2-Class**	**3-Class**
Subject1	0.5595	0.4279
Subject2	0.6250	0.5032
Subject3	0.5798	0.5979
Subject4	0.6058	0.4635
Subject5	0.5462	0.4790
Subject6	0.7008	0.5777
Subject7	0.7041	0.6290
Subject8	0.6039	0.5243
Subject9	0.6633	0.6557
Subject10	0.7300	0.4803
Subject11	0.6300	0.4014
Subject12	0.6085	0.4260
Subject13	0.7187	0.4260
Subject14	0.6675	0.4778
Subject15	0.7875	0.5745
Subject16	0.6592	0.4215
Subject17	0.5453	0.5579
Subject18	0.6966	0.6442
Subject19	0.6246	0.4915
Subject20	0.7302	0.5400
Subject21	0.5113	0.4319
Subject22	0.6150	0.3900
Subject23	0.7133	0.5663
Subject24	0.5294	0.4603
Subject25	0.5437	0.3488
Subject26	0.5846	0.5758
Subject27	0.7592	0.5391

Subject28	0.5908	0.5286
Subject29	0.6927	0.5063
Subject30	0.5560	0.6290
Subject31	0.5414	0.3587
Subject32	0.6767	0.5805
Average	***0.6344***	***0.5067***
STD	0.1252	0.0840

Underscored values represent the averaged performance of Deep EEGNet for each case.

Table 6.8 shows the performance summarizations of Case2, where the optimized Deep EEGNet yielded 63.44% classification accuracy for 2-class classification, and 50.67% classification accuracy for 3-class classification. Table 6.7 demostrated that the CNNs+MLP structure with 2-layer CNNs for 2-class classification and the CNNs+RNNs structure with 1-layer CNNs for 3-class classification were determined as optimal deep learning structures for the Case2 database.

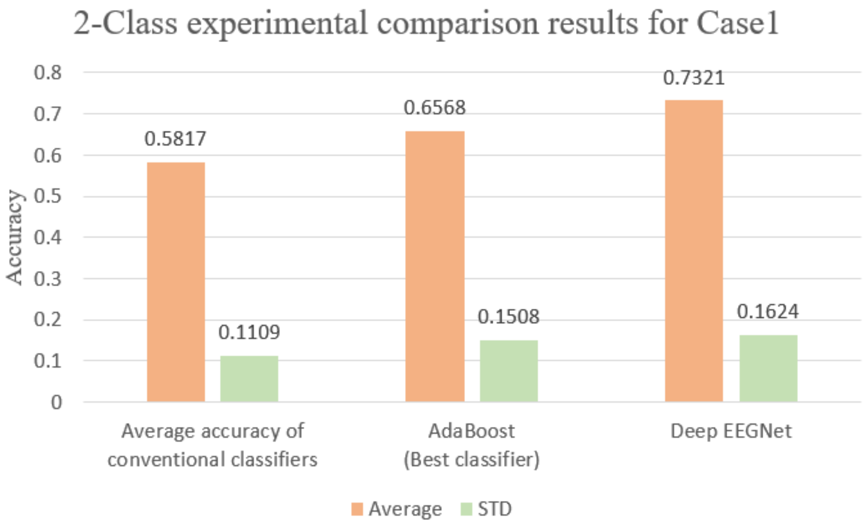
We also experimented with the effectiveness of the channel grouping scheme for using local characteristics of EEG signals. As you can see in Figure 6.6, the performance of the channel grouping scheme generally shows better and more stable results than those schemes using an individual channel which uses only one CNNs model for all channels. Especially, in the case of 2-class classification for Case1, we could achieve an averaged classification accrurracy improvement of 16.39% using the proposed channel grouping scheme.



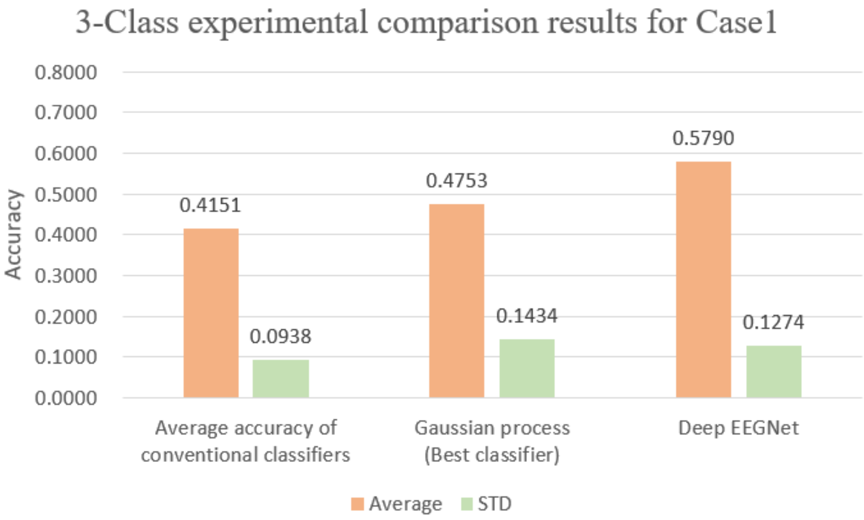
(a) Case1

(b) Case2

Figure 6.6 The comparison results for channel grouping for Case1 and Case 2

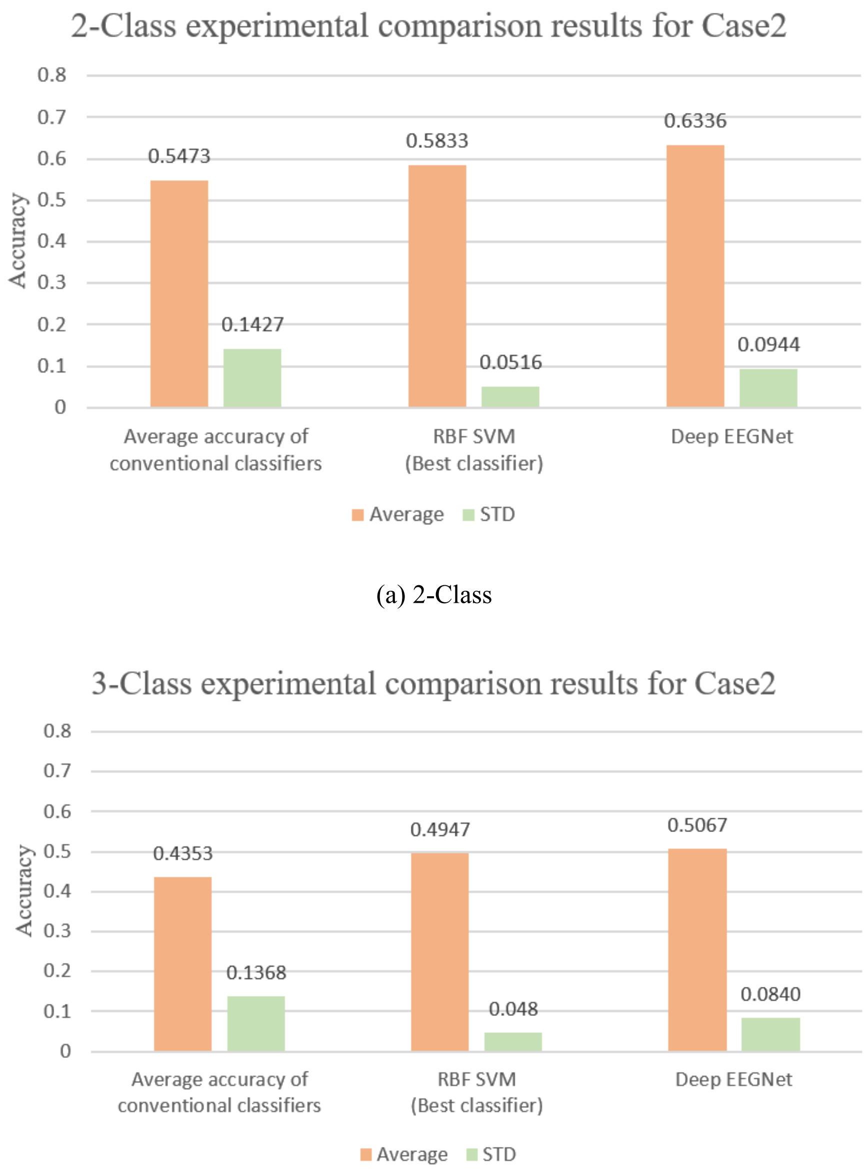


(a) 2-Class



(b) 3-Class

Figure 6.7 The comparison results for Case 1 (VS Convenctional methods)



(a) 2-Class

(b) 3-Class

Figure 6.8 The comparison results for Case 2 (VS Conventional methods)

Figure 6.7 and 6.8 show the final results of this study. We compared the results of the proposed model with those of the best algorithms which showed the highest accuracy among conventional machine learning algorithms. In summary, the proposed deep learning model, Deep EEGNet, showed superior results than those of conventional machine learning methods in both Case1 and Case2 for 2-class and 3-class human emotional clasification problems. In Case1, the performance of Deep EEGNet was 11.46% higher than that of the best conventional algorithm, AdaBoost, for 2-class classifiaction and 21.82% higher than that of the best conventional algorithm, Gaussian process, for 3-class classifiaction. Furthermore, in Case2, the performance of Deep ECGNet is 8.62% higher than that of the best conventional algorithm, RBF SVM, for 2-class classifiaction and 2.43% higher than that of the best conventional algorithm, RBF SVM, for 3-class classifiaction. In particular, the proposed model in Case 2 for binary classification shows notably better classification performance of 63.36% compared to Koelstra's previous study using Naïve Bayes algorithm, which managed only 57.6% classification accuracy [Koelstra2012].

6.4 Summary

This study proposes an optimal deep learning design methodology to analyze EEG signals for the recognition of human emotional states without traditional feature engineering. As such, this study intends to outline how to determine deep learning structures and hyperparameters for nonstationary, nonlinear and noisy EEG data using the Bayesian optimization technique. The results demonstrated that the proposed method produced superior performance compared to conventional methods. The proposed deep learning model could successfully extract basic feature characteristics of EEG signals for the determination of human emotional states using the various optimal 1D CNNs+MLP or CNNs+RNNs structures.

The various experimental results illustrate that the proposed Deep EEGNet provides an optimal deep learning framework for the recognition of human emotional states with the optimal DNNs architectures and hyperparameters. Note that the proposed model is able to be trained by nonstationary and noisy raw EEG signals without any cumbersome feature extraction procedure. These design concepts could provide an automatic optimizing strategy to build DNNs architectures for the analysis of raw EEG signals, which could be varied depending on data due to its data driven manner (see Table 6.5 and 6.7).

In contrast to the methodology of Deep ECGNet which was based on the prior knowledge of periodicity of ECG signals, Deep EEGNet can be trained using an efficient automatic optimization methodology without any preprocessing based on prior knowledge of given data. In addition, the performance of Deep EEGNet could be improved if larger datasets, more varied optimization parameters, more complex types of structures, and more optimization time are provided. Finally, this study could also be applicable to other nonstationary, nonlinear and noisy time series data.

Chapter 7

Concluding Remarks

7.1 Summary of Thesis and Contributions

There are two major obstacles to overcome for seamless end-to-end deep learning pattern recognition of raw time series signal data. The first is the nonstationary, nonlinear, and noisy nature of common time series signal data, which must be overcome without special feature engineering of the data. The second problem is how to design a deep learning model with good generalization characteristics and learnability when using such data. This thesis made three contributions in these respects.

The first contribution was to select a basic deep learning unit of pattern recognition of time series signals. For the learning of the end-to-end deep learning model, we need to generate a hidden representation by directly inputting the raw signal. The 1d convolution neural networks are suitable for this, and the way of the parameters settings is shown through several experiments. Since CNNs can maintain the sequential nature of the hidden representation from their inputs, they can be extended to the CRNN model by using LSTM as well as deep fully connected neural networks as a classifier.

The second contribution is to suggest a design method for a deep learning model architecture of signal with periodic characteristics. Deep learning is a way to learn the characteristics of data by accumulating deeper layers as the characteristics of data become more complicated. In other words, if the characteristics of the data can be reflected in the model, the layer does not have to be a multi-layered or complex model. In this thesis, we present a new design paradigm for periodic ECG signals. The heart of a normal person is running at a cycle of about 0.8 seconds on average, and a cycle of 0.6 seconds in the form of a specific waveform within a period of 0.8 seconds. The Deep ECGNet, which was previously applied to the characteristics of ECG waveforms in terms of signals, was proposed as a model design technique that can determine human stress by using CNN as a basic feature extractor. In other words, this thesis proposes a model design methodology that can accurately recognize the pattern characteristics within one cycle by reflecting the periodic characteristics of signals to the CNN model parameters. The proposed Deep ECGNet showed better performance using only raw signal data without any special feature engineering processes compared to existing conventional techniques. It showed not only better performance but also a more compact structure than empirically designed deeper learning models. The idea of modeling the periodicity of signals in this way is expected to be fully exploited for periodic signals with similar characteristics.

The third contribution is to suggest an automatic optimization design method using only raw signals for nonstationary, nonlinear and noisy signals. The EEG signal is a representative signal with nonstationary, nonlinear and noisy characteristics in time series data, which makes it very challenging to analyze patterns. Therefore, even when the deep learning method has previously been applied to the analysis of such EEG signals, the methods have used hand-crafted extracted features rather than methods using raw signals. However, in this thesis, we propose an end-to-end deep learning optimization method using only raw EEG signals, so that a deep learning model designed with the proposed methodology can be effectively applied for EEG analysis without hand-crafted feature extraction. The performance of the proposed method is better than that of current conventional methods. These optimization techniques include not only simple hyperparameters but also structural optimization methods. The proposed methods can be applied to nonstationary, nonlinear and noisy EEG signals as well as time series data which are not well known in advance. Thus, the most notable of this study is that it suggests a general design methodology for an end-to-end deep learning model of time series data with such characteristics.

In conclusion, the significance of this dissertation is that it proposes end-to-end deep learning design methodologies that were not available in the past using only raw time series signal data obtained from various experiments.

7.2 Limitations of the Proposed Methods

This thesis proposes end-to-end deep learning design methodologies for pattern recognition in time series signals. All the experimental results of this study were performed by creating a model for each subject. That is, since the data characteristics of the subjects are so different from each other, the current model alone is limited in its ability to construct a subject-independent integrated model. These problems can cause the proposed model to have either very good or very poor performance for any particular subject. As a result, robustness issues may manifest themselves in the form of large standard deviation in the performance of the entire model. Therefore, the framework design technique needs to be extended so that it can reflect the characteristics of all subjects for generalization aspects.

Also, it was difficult to obtain a large amount of data because the experiments were basically conducted to measure physiological signals of subjects.

Therefore, the validity of the big data of the model was not verified. In addition, the current framework has only validated physiological signals and have not been validated for general time series signals. From the viewpoint of generalization of the proposed methodology, verification of general sensor signals and other continuous time series data have also not been verified.

Finally, from an optimization point of view, the proposed model methodology can show limitations in the variety of structural optimizations. In other words, it searches only candidate models that are defined in advance, and therefore, a new strategy of structural optimization is needed to overcome such limitations.

7.3 Suggestions for Future Works

This thesis proposes unimodal end-to-end deep learning design methodologies. In the case of general sensor data, multimodal data are often used, and the problem of combining them is also an important research topic. The end-to-end deep learning design methodology of multimodal structures should be investigated as a future study topic. These studies should include efficient learning methods for large amounts of data, effective combination of heterogeneous data, and research on ideal model structures using attention mechanisms and residual connections.

In addition, the proposed models in this thesis were developed for each subject. This has several disadvantages with regards to learning. As such, it is difficult to make a unified master algorithm that recognizes all subjects at once.

Moreover, the proposed models have long learning time and bad generalization characteristics. Therefore, research on the transfer learning algorithm for time series data with different characteristics is also a potential research avenue for future investigation. Through this future work, researchers will hopefully be able to develop an integrated model with generalization characteristics for several different subject's data, which would have many advantages in terms of its model size and learning time.

Bibliography

Ahirwal, M. K., & Londhe, N. D. (2012). Power spectrum analysis of EEG signals for estimating visual attention. *International Journal of computer applications*, *42*(15).

Baek, H. J., Cho, C. H., Cho, J., & Woo, J. M. (2015). Reliability of ultra-short-term analysis as a surrogate of standard 5-min analysis of heart rate variability. *Telemedicine and e-Health*, *21*(5), 404-414.

Balconi, M., & Mazza, G. (2009). Laterality, EEG alpha band power and behavioural inhibition (BIS) and activation (BAS) systems in emotional face comprehension. *JOURNAL OF THE INTERNATIONAL NEUROPSYCHOLOGICAL SOCIETY*, *15*(S2), 24-24.

Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*.

Belkic, K., Landsbergis, P., Schnall, P., Backer, D., Theorell, T., Siegrist, J., ... & Karasek, R. (2000). Psychosocial factors: review of the empirical data among men. *Occupational Medicine, State of the Art Reviews, the Workplace and Cardiovascular Disease*, *15*(1), 24-46.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281-305.

Bong, S. Z., Murugappan, M., & Yaacob, S. (2012). Analysis of electrocardiogram (ecg) signals for human emotional stress classification. In *Trends in Intelligent Robotics, Automation, and Manufacturing* (pp. 198-205). Springer, Berlin, Heidelberg.

Chen, X. W., & Huang, T. (2003). Facial expression recognition: a clustering-based approach. *Pattern Recognition Letters*, *24*(9-10), 1295-1302.

Crosby, M. E., Auernheimer, B., Aschwanden, C., & Ikehara, C. (2001, November). Physiological data feedback for application in distance education. In *Proceedings of the 2001 workshop on Perceptive user interfaces* (pp. 1-5). ACM.

Cui, J., & Freeden, W. (1997). Equidistribution on the sphere. *SIAM Journal on Scientific Computing*, *18*(2), 595-609.

Cui, Z., Chen, W., & Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*.

Damasio, A. R., Grabowski, T. J., Bechara, A., Damasio, H., Ponto, L. L., Parvizi, J., & Hichwa, R. D. (2000). Subcortical and cortical brain activity during the feeling of self-generated emotions. *Nature neuroscience*, *3*(10), 1049.

Dellaert, F., Polzin, T., & Waibel, A. (1996). Recognizing emotion in speech. In *Fourth International Conference on Spoken Language Processing*.

Dennis, J., Tran, H. D., & Li, H. (2011). Spectrogram image feature for sound event classification in mismatched conditions. *IEEE signal processing letters*, *18*(2), 130-133.

Dente, E., Bharath, A. A., Ng, J., Vrij, A., Mann, S., & Bull, A. (2006). Tracking hand and finger movements for behaviour analysis. *Pattern Recognition Letters*, *27*(15), 1797-1808.

Ekman, P., Levenson, R. W., & Friesen, W. V. (1983). Autonomic nervous system activity distinguishes among emotions. *Science*, *221*(4616), 1208-1210.

Elsken, T., Metzen, J. H., & Hutter, F. (2018). Neural architecture search: A survey. *arXiv preprint arXiv*:1808.05377

Falkner, S., Klein, A., & Hutter, F. (2018). BOHB: Robust and efficient hyperparameter optimization at scale. *arXiv preprint arXiv*:1807.01774.

Ha, J., Kim, K. M., & Zhang, B. T. (2015, January). Automated Construction of Visual-Linguistic Knowledge via Concept Learning from Cartoon Videos. In *AAAI* (pp. 522-528).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.

Hatami, N., Gavet, Y., & Debayle, J. (2018, April). Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)* (Vol. 10696, p. 106960Y). International Society for Optics and Photonics.

Hjortskov, N., Rissén, D., Blangsted, A. K., Fallentin, N., Lundberg, U., & Søgaard, K. (2004). The effect of mental stress on heart rate variability and blood pressure during computer work. *European journal of applied physiology*, *92*(1-2), 84-89.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Holly, R. M., Alison, H. N., Carlo, E. N., Egbert, U. N., Carl, K. H., Barbara, A. N., ... & Jaime, D. M. (1997). Impact of acute mental stress on sympathetic nerve activity and regional blood flow in advanced heart failure. *Circulation*, *96*(6), 1835-1842.

https://ecgwaves.com/ekg-ecg-interpretation-normal-p-wave-qrs-complex-stsegment-t-wave-j-point (last accessed September 15, 2017).

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... & Liu, H. H. (1998, March). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences* (Vol. 454, No. 1971, pp. 903-995). The Royal Society.

Hwang, B., You, J., Vaessen, T., Myin-Germeys, I., Park, C., & Zhang, B. T. (2018). Deep ECGNet: An Optimal Deep Learning Framework for Monitoring Mental Stress Using Ultra Short-Term ECG Signals. *TELEMEDICINE and e-HEALTH*.

Ioffe S, Szegedy C (2015).Batch Normalization: Accelerating Deep Network

Training by Reducing Internal Covariate Shift. *CoRR.–2015.–Vol. abs/1502.03167.–URL: http://arxiv. org/abs/1502.03167*.

Jasper, H. H. (1958). The ten-twenty electrode system of the International Federation. *Electroencephalogr. Clin. Neurophysiol.*, *10*, 370-375.

Jirayucharoensak, S., Pan-Ngum, S., & Israsena, P. (2014). EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation. *The Scientific World Journal*, *2014*.

Kandel E, Schwartz J, Jessell T. Principles of Neural Science, *Mc Graw Hill*, (2000)

Karthikeyan, P., Murugappan, M., & Yaacob, S. (2013). Detection of human stress using short-term ECG and HRV signals. *Journal of Mechanics in Medicine and Biology*, *13*(02), 1350038.

Kim, K. H., Bang, S. W., & Kim, S. R. (2004). Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, *42*(3), 419-427.

Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015, August). Convolutional Neural Networks for patient-specific ECG classification. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*(pp. 2608-2611). IEEE.

Kislova, O. O., & Rusalova, M. N. (2009). EEG coherence in humans: Relationship with success in recognizing emotions in the voice. *Neuroscience and behavioral physiology*, *39*(6), 545-552.

Koelstra, S., Muhl, C., Soleymani, M., Lee, J. S., Yazdani, A., Ebrahimi, T., ... & Patras, I. (2012). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, *3*(1), 18-31.

Lane, R. D., Chua, P. M., & Dolan, R. J. (1999). Common effects of emotional valence, arousal and attention on neural activation during visual processing of pictures. *Neuropsychologia*, *37*(9), 989-997.

Lainscsek, C., Hernandez, M. E., Weyhenmeyer, J., Sejnowski, T. J., & Poizner, H. (2013). Non-linear dynamical analysis of EEG time series distinguishes patients with Parkinson's disease from healthy individuals. *Frontiers in neurology*, *4*, 200.

LeCun, Y. (2012, October). Learning invariant feature hierarchies. In *European conference on computer vision* (pp. 496-505). Springer, Berlin, Heidelberg.

LeCun, Y. (2015). LeNet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In *ISCAS* (Vol. 2010, pp. 253-256).

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1), 6765-6816.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmén, P., Engström, M., & Dohns, I. E. (1994). Psychophysiological stress and EMG activity of the trapezius muscle. *International journal of behavioral medicine*, *1*(4), 354-370.

Mehdiyev, N., Lahann, J., Emrich, A., Enke, D., Fettke, P., & Loos, P. (2017). Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry. *Procedia Computer Science*, *114*, 242-249.

Murugappan, M., Murugappan, S., & Zheng, B. S. (2013). Frequency band analysis of electrocardiogram (ECG) signals for human emotional state classification using discrete wavelet transform (DWT). *Journal of physical therapy science*, *25*(7), 753-759.

Ng, A. Y. (2004, July). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning* (p. 78). ACM.

Park, C., Looney, D., ur Rehman, N., Ahrabian, A., & Mandic, D. P. (2013). Classification of motor imagery BCI using multivariate empirical mode decomposition. *IEEE Transactions on neural systems and rehabilitation engineering*, *21*(1), 10-22.

Park, C., Plank, M., Snider, J., Kim, S., Huang, H. C., Gepshtein, S., ... & Poizner, H. (2014). EEG gamma band oscillations differentiate the planning of spatially directed movements of the arm versus eye: multivariate empirical mode decomposition analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(5), 1083-1096.

Ranganathan, H., Chakraborty, S., & Panchanathan, S. (2016, March). Multimodal emotion recognition using deep learning architectures. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on* (pp. 1-9). IEEE.

Rehman, N., & Mandic, D. P. (2010, May). Multivariate empirical mode decomposition. In *Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences*(Vol. 466, No. 2117, pp. 1291-1302). The Royal Society.

Ren, Y., & Wu, Y. (2014, July). Convolutional deep belief networks for feature extraction of EEG signal. In *Neural Networks (IJCNN), 2014 International Joint Conference on* (pp. 2850-2853). IEEE.

Rutkowski, T. M., Mandic, D., & Barros, A. K. (2007). A multimodal approach to communicative interactivity classification. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, *49*(2), 317-328.

Scheirer, J., Fernandez, R., Klein, J., & Picard, R. W. (2002). Frustrating the user on purpose: a step toward building an affective computer. *Interacting with computers*, *14*(2), 93-118.

Selvaraj, J., Murugappan, M., Wan, K., & Yaacob, S. (2013). Classification of emotional states from electrocardiogram signals: a non-linear approach based on hurst. *Biomedical engineering online*, *12*(1), 44.

Seok, W., & Park, C. (2018). Recognition of Human Motion with Deep Reinforcement Learning. *IEIE Transactions on Smart Processing & Computing*, *7*(3), 245-250.

Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, *39*(11), 2298-2304.

Shumway, R. H., & Stoffer, D. S. (2000). Time series analysis and its applications. *Studies In Informatics And Control*, *9*(4), 375-376.

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003, August). Best practices for convolutional neural networks applied to visual document analysis. In *null* (p. 958). IEEE.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951-2959).

Sobin, C., & Alpert, M. (1999). Emotion in speech: The acoustic attributes of fear, anger, sadness, and joy. *Journal of psycholinguistic research*, *28*(4), 347-365.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

Taelman, J., Vandeput, S., Spaepen, A., & Van Huffel, S. (2009). Influence of mental stress on heart rate and heart rate variability. In *4th European conference of the international federation for medical and biological engineering* (pp. 1366-1369). Springer, Berlin, Heidelberg.

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Heart Rate Variability: Standards of measurement, physiological interpretation, and clinical use. Eur Heart J 1996;7:1043–1065

Thorn, B. E., Pence, L. B., Ward, L. C., Kilgo, G., Clements, K. L., Cross, T. H., ... & Tsui, P. W. (2007). A randomized clinical trial of targeted cognitive behavioral treatment to reduce catastrophizing in chronic headache sufferers. *The Journal of Pain*, *8*(12), 938-949.

Turner, J. T., Page, A., Mohsenin, T., & Oates, T. (2014, March). Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection. In *2014 AAAI Spring Symposium Series*.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016, September). WaveNet: A generative model for raw audio. *In SSW* (p. 125).

Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural computing and applications*, *24*(1), 175-186.

Xia, P., Hu, J., & Peng, Y. (2018). EMG-Based Estimation of Limb Movement Using Deep Learning With Recurrent Convolutional Neural Networks. *Artificial organs*, *42*(5), E67-E77.

Yang, J., Nguyen, M. N., San, P. P., Li, X., & Krishnaswamy, S. (2015, July). Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *Ijcai*(Vol. 15, pp. 3995-4001).

Zhang, B.-T. (2008). Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory. *IEEE Computational Intelligence Magazine*, 3(3):49– 63.

Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zheng, B. S., Murugappan, M., & Yaacob, S. (2012, September). Human emotional stress assessment through Heart Rate Detection in a customized protocol experiment. In *Industrial Electronics and Applications (ISIEA), 2012 IEEE Symposium on*(pp. 293-298). IEEE.

초 록

시계열 데이터의 패턴 인식 문제는 4차 산업 혁명의 패러다임 전환과 함께 매우 중요한 인공 지능의 한 분야가 되었다. 이에 따라, 지난 몇 년간 이와 관련된 많은 연구들이 이루어져 왔으며, 최근에는 심층 학습망 (deep learning networks) 모델을 이용한 연구들이 주를 이루어 왔다. 시계열 데이터는 비정상, 비선형 그리고 잡음 (nonstationary, nonlinear and noisy) 특성으로 인하여 시계열 데이터의 패턴 인식 수행을 위해선, 데이터의 주요한 특징점을 추출하기 위한 최적화된 모델의 설계가 필수적이다.

본 논문은 대표적인 시계열 데이터인 생체 신호를 사용하여 여러 특징 벡터 추출 방법 (hand-crafted feature engineering methods)을 이용한 패턴 인식 기법에 대하여 논할 뿐만 아니라, 궁극적으로는 특징 벡터 추출 과정이 없는 종단 심층 학습망 설계 방법론에 대한 연구 내용을 담고 있다. 시계열 신호는 시간 축 상에서 크게 주기적 신호와 비주기적 신호로 구분할 수 있는데, 본 연구는 이러한 두 유형의 신호들에 대한 패턴 인식을 위해 두 가지 종단 심층 학습망에 대한 설계 방법론을 제안한다.

첫 번째 제안된 방법론을 이용해 설계된 모델은 신호의 주기적 특성을 이용한 Deep ECGNet이다. 심장 근육의 전기 생리학적 패턴으로부터 기록된 심전도 (Electrocardiogram, ECG)는 이벤트 기반 스트레스 수준을 추정하기 위한 척도 (bio marker)를 제공하는 유효한 데이터가 될 수 있다. 전통적으로 심전도의 심박수 변동성

(Herat Rate Variability, HRV) 매개변수 (parameter)는 심장 질환 환자의 정신적 스트레스 상태 및 사망률을 모니터링하는 데 사용되었다. 하지만, 표준 심박수 변동성 매개 변수는 측정 주기가 5분 이상으로, 측정 시간이 길다는 단점이 있다. 본 논문에서는 심층 학습망 모델을 이용하여 10초 간격의 ECG 데이터만을 이용하여, 추가적인 특징 벡터의 추출 과정 없이 인간의 스트레스 상태를 인식할 수 있음을 보인다. 제안된 설계 기법은 ECG 신호의 주기적 특성을 모델에 반영하였는데, ECG의 은닉 특징 추출기로 사용된 1D CNNs 및 RNNs 모델의 주요 매개 변수에 주기적 특성을 반영함으로써, 한 주기 신호의 스트레스 상태에 따른 주요 특징점을 종단 학습망 내부적으로 추출할 수 있음을 보였다. 실험 결과 제안된 방법이 기존 심박수 변동성 매개변수와 spectrogram 추출 기법 기반의 패턴 인식 방법보다 좋은 성능을 나타내고 있음을 확인할 수 있었다.

두 번째 제안된 방법론은 비 주기적이며 비정상, 비선형 그리고 잡음 특성을 지닌 신호의 패턴인식을 위한 최적 종단 심층 학습망 자동 설계 방법론이다. 뇌파 신호 (Electroencephalogram, EEG)는 중추 신경계 (CNS)에서 발생되어 무의식 상태에서도 본연의 감정 상태를 나타내는데, EEG 신호의 낮은 신호 대 잡음비 (SNR)로 인해 뇌파를 이용한 감정 상태 판정을 위해서 주로 주파수 영역의 스펙트럼 분석이 뇌파 연구에 적용되어 왔다. 통상적으로 뇌파 신호는 푸리에 (Fourier) 또는 웨이블렛 (wavelet) 분석을 사용하여 여러 주파수 대역으로 필터링 된다. 이렇게 추출된 주파수 특징 벡터는 보통 얕은 학습 분류기 (shallow machine learning classifier)의 입력으로 사용되어 패턴 인식을 수행하게 된다. 본 논문에서는 이러한

기본적인 특징 벡터 추출 과정이 없는 베이지안 최적화 (Bayesian optimization) 기법을 이용한 종단 심층 학습망 자동 설계 기법을 제안한다. 베이지안 최적화 기법은 초 매개변수 (hyperparamters)를 최적화하기 위한 기계 학습 분야의 대표적인 최적화 기법인데, 최적화 과정에서 평가 시간이 많이 소요되는 목적 함수 (expensive black box function)를 갖고 있는 최적화 문제에 적합하다. 이러한 베이지안 최적화를 이용하여 기본적인 학습 모델인 1D CNNs 및 RNNs의 전체 모델의 초 매개변수 및 구조적 최적화를 수행하는 방법을 제안하였으며, 제안된 방법론을 바탕으로 Deep EEGNet이라는 인간의 감정상태를 판별할 수 있는 모델을 제안하였다. 여러 실험을 통해 제안된 모델이 기존의 주파수 특징 벡터 (band power feature) 추출 기법 기반의 전통적인 감정 패턴 인식 방법보다 좋은 성능을 나타내고 있음을 확인할 수 있었다.

결론적으로 본 논문은 시계열 데이터를 이용한 패턴 인식문제를 여러 특징 벡터 추출 기법 기반의 전통적인 방법을 통해 설계하는 방법부터, 추가적인 특징 벡터 추출 과정 없이 원본 데이터만을 이용하여 종단 심층 학습망을 설계하는 방법까지 제안하였다. 또한, 다양한 실험을 통해 제안된 방법론이 시계열 신호 데이터를 이용한 패턴 인식 문제에 효과적으로 적용될 수 있음을 보였다.

주요어: 패턴 인식, 시계열 데이터, 종단 심층 학습, 하이퍼 파라미터 최적화, 정서적 컴퓨팅

학 번: 2014-30319