Spring 2020

Deep Cellular Recurrent Neural Architecture for Efficient Multidimensional Time-Series Data Processing

Lasitha S. Vidyaratne
Old Dominion University, lasithasv@gmail.com

DEEP CELLULAR RECURRENT NEURAL ARCHITECTURE FOR EFFICIENT MULTIDIMENSIONAL TIME-SERIES DATA PROCESSING

by

Lasitha S. Vidyaratne
M.Eng. December 2009, University of Nottingham

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2020

Approved by:

Khan M. Iftekharuddin (Director)

Chunsheng Xin (Member)

Jiang Li (Member)

Norou Diawara (Member)

ABSTRACT

DEEP CELLULAR RECURRENT NEURAL ARCHITECTURE FOR EFFICIENT MULTIDIMENSIONAL TIME-SERIES DATA PROCESSING

Lasitha S. Vidyaratne
Old Dominion University, 2020
Director: Dr. Khan M. Iftekharuddin

Efficient processing of time series data is a fundamental yet challenging problem in pattern recognition. Though recent developments in machine learning and deep learning have enabled remarkable improvements in processing large scale datasets in many application domains, most are designed and regulated to handle inputs that are static in time. Many real-world data, such as in biomedical, surveillance and security, financial, manufacturing and engineering applications, are rarely static in time, and demand models able to recognize patterns in both space and time. Current machine learning (ML) and deep learning (DL) models adapted for time series processing tend to grow in complexity and size to accommodate the additional dimensionality of time. Specifically, the biologically inspired learning based models known as artificial neural networks that have shown extraordinary success in pattern recognition, tend to grow prohibitively large and cumbersome in the presence of large scale multi-dimensional time series biomedical data such as EEG.

Consequently, this work aims to develop representative ML and DL models for robust and efficient large scale time series processing. First, we design a novel ML pipeline with efficient feature engineering to process a large scale multi-channel scalp EEG dataset for automated detection of epileptic seizures. With the use of a sophisticated yet computationally efficient time-frequency analysis technique known as harmonic wavelet packet transform and an efficient self-similarity computation based on fractal dimension, we achieve state-of-the-art performance for automated seizure detection in

EEG data. Subsequently, we investigate the development of a novel efficient deep recurrent learning model for large scale time series processing. For this, we first study the functionality and training of a biologically inspired neural network architecture known as cellular simultaneous recurrent neural network (CSRN). We obtain a generalization of this network for multiple topological image processing tasks and investigate the learning efficacy of the complex cellular architecture using several state-of-the-art training methods. Finally, we develop a novel deep cellular recurrent neural network (CDRNN) architecture based on the biologically inspired distributed processing used in CSRN for processing time series data. The proposed DCRNN leverages the cellular recurrent architecture to promote extensive weight sharing and efficient, individualized, synchronous processing of multi-source time series data. Experiments on a large scale multi-channel scalp EEG, and a machine fault detection dataset show that the proposed DCRNN offers state-of-the-art recognition performance while using substantially fewer trainable recurrent units.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my adviser, Dr. Khan M. Iftekharuddin for giving me the opportunity to work in his research group and for continuous guidance and advice throughout my Ph.D. program. His mentoring helped me to expand my knowledge and skills that will help my career as a researcher. I am honored to have worked for such a renowned and visionary professor.

I would also like to thank the members of my advisory committee: Dr. Jiang Li, Dr. Norou Diawara, and Dr. Chunsheng Xin for their time as well as their valuable guidance and suggestions.

I am most grateful to my father, Vidya Vidyaratne, my late mother, Shriyani Munasinghe Vidyaratne, and my sister, Lasni Vidyaratne for the support, guidance, and for encouraging me for higher education to fulfill my ambitions.

Furthermore, I wish to acknowledge Mahbubul Alam, Linmin Pei, and Alex Glandon for being supportive friends with valuable input and advice as we collaborated on various projects throughout my Ph.D. I also wish to thank all of my colleagues at the ODU Vision Lab for their support and encouragements.

Finally, I would like to thank my wife, Ashani, who has stood by me throughout this Ph.D. journey and has supported me immensely in numerous ways.

This dissertation is dedicated to my late mother, Shriyani Munasinghe Vidyaratne, who passed away while I was a Ph.D. student, after being in a vegetative state for four years. It was with the hope of helping others like my mother that I embarked on this field of research.

TABLE OF CONTENTS

Page

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER 1: INTRODUCTION 1

1.1 FEATURE ENGINEERING FOR TIME SERIES DATA 4
1.2 TIME SERIES PROCESSING WITH ARTIFICIAL NEURAL NETWORKS 5
1.3 BIOLOGICAL BASIS FOR EFFICIENT INFORMATION PROCESSING 6
1.4 CELLULAR SIMULTANEOUS RECURRENT NETWORK 8
1.5 FROM MACHINE LEARNING TO ANNS FOR TIME SERIES ANALYSIS 9
1.6 PROPOSED WORK AND CONTRIBUTIONS 10
1.7 ORGANIZATION OF THE DISSERTATION 12

CHAPTER 2: BACKGROUND OF THE STUDY 14

2.1 TIME-SERIES DATA 14
2.2 FEATURE ENGINEERING FOR TIME-SERIES DATA 14
2.2.1 FRACTAL ANALYSIS 15
2.2.2 HARMONIC WAVELET PACKET TRANSFORM 16
2.3 ARTIFICIAL NEURAL NETWORKS 20
2.4 NEURAL NETWORK ARCHITECTURES 22
2.4.1 MULTI-LAYERED FEED-FORWARD NETWORKS 22
2.4.2 RECURRENT NEURAL NETWORK 23
2.4.3 CELLULAR NEURAL NETWORK 31
2.4.4 CELLULAR SIMULTANEOUS RECURRENT NETWORK 32
2.4.5 CSRN TRAINING 35

CHAPTER 3: EFFICIENT MACHINE LEARNING PIPELINE FOR REAL TIME EPILEPTIC SEIZURE DETECTION USING SCALP EEG 38

3.1 CHAPTER OVERVIEW 38
3.2 LITERATURE REVIEW 39
3.2.1 FEATURE EXTRACTION BASED ON WAVELET TRANSFORM 39
3.2.2 SELF-SIMILARITY FEATURES WITH FRACTAL DIMENSION 40
3.2.3 FEATURE CLASSIFICATION WITH MACHINE LEARNING 40
3.2.4 SEIZURE DETECTION WITH EEG TIME SERIES DATA 41
3.3 PROPOSED REAL-TIME EPILEPTIC SEIZURE DETECTION PIPELINE 42
3.3.1 EEG DATA 43
3.3.2 FEATURE EXTRACTION 45

3.3.3 FEATURE CLASSIFICATION 48
3.4 RESULTS AND DISCUSSION 49
3.4.1 SEIZURE DETECTION SENSITIVITY 50
3.4.2 SEIZURE DETECTION LATENCY 52
3.4.3 SEIZURE DETECTION SPECIFICITY 53
3.4.4 PERFORMANCE COMPARISON WITH OTHER METHODS 54
3.4.5 PERFORMANCE EVALUATION USING SHORT TERM EEG 56
3.4.6 COMPARATIVE EVALUATION OF FEATURE IMPORTANCE 58
3.5 SUMMARY 61

CHAPTER 4: COMPARISON OF CONSTRAINED AND UNCONSTRAINED LEARNING FOR CELLULAR SIMULTANEOUS RECURRENT NETWORK IN IMAGE PROCESSING 64

4.1 CHAPTER OVERVIEW 64
4.2 LITERATURE REVIEW 65
4.2.1 TYPICAL TRAINING METHODS USED FOR CSRN 65
4.2.2 FUNCTION APPROXIMATION AND OPTIMIZATION CAPABILITY OF CSRN 66
4.3 PROPOSED TRAINING ALGORITHMS FOR CSRN BASED IMAGE PROCESSING 67
4.3.1 CSRN TRAINING WITH EXTENDED KALMAN FILTER (EKF) 67
4.3.2 CSRN TRAINING WITH UNSCENTED KALMAN FILTER (UKF) 69
4.3.3 CSRN TRAINING WITH PARTICLE SWARM OPTIMIZATION (PSO) 70
4.4 GENERALIZED SPATIAL DOMAIN IMAGE PROCESSING WITH CSRN 71
4.4.1 BINARY IMAGE PROCESSING 73
4.4.2 IMAGE FILTERING 73
4.4.3 IMAGE AFFINE TRANSFORMATIONS 74
4.4.4 COST FUNCTION FORMULATION FOR GENERALIZED IMAGE PROCESSING WITH CSRN 75
4.4.5 A QUALITATIVE COMPARISON BETWEEN CONSTRAINED AND UNCONSTRAINED TRAINING ALGORITHMS 76
4.5 RESULTS AND DISCUSSION 78
4.5.1 BINARY IMAGE PROCESSING 79
4.5.2 IMAGE FILTERING 80
4.5.3 IMAGE AFFINE TRANSFORMATIONS 81
4.6 SUMMARY 87

CHAPTER 5: DEEP CELLULAR RECURRENT NEURAL NETWORK FOR EFFICIENT TIME-SERIES ANALYSIS 89

5.1 CHAPTER OVERVIEW 89
5.2 LITERATURE REVIEW 90
5.2.1 FEATURE BASED TECHNIQUES IN TIME-SERIES PROCESSING 90
5.2.2 DEEP NEURAL NETWORKS FOR TIME SERIES PROCESSING 91
5.3 THE DEEP CELLULAR RECURRENT NEURAL NETWORK FOR LARGE-SCALE TIME SERIES PROCESSING 94

5.3.1 DCRNN ARCHITECTURE FOR TIME SERIES PROCESSING 94
5.3.2 COMPLEXITY ANALYSIS OF THE PROPOSED DCRNN ARCHITECTURE 99
5.3.3 MULTI-CHANNEL EEG PROCESSING WITH DCRNN 100
5.3.4 MACHINE FAULT DETECTION WITH DCRNN 101
5.3.5 NETWORK PREPERATION 103
5.4 RESULTS AND DISCUSSION 104
5.4.1 MULTI-CHANNEL SCALP EEG DATASET 104
5.4.1 MACHINE FAULT DETECTION DATASET 108
5.5 SUMMARY 112

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS 114

REFERENCES 120

VITA 128

LIST OF TABLES

Table Page

1. Summary of proposed contributions 11

2. Long term EEG dataset (Dataset A) 44

3. Seizure detection algorithm performance comparison using Dataset A 55

4. Seizure detection algorithm performance comparison using Dataset B (short term EEG) 57

5. Evaluation of feature importance 59

6. Comparison of constrained and unconstrained optimizers for training universal approximators 77

7. performance evaluation metric for image processing 79

8. Summary of greyscale image transformation with CSRN 84

9. Training time vs. Image size for EKF, UKF, and PSO based training of CSRN 85

10. Performance comparison of the proposed DCRNN model with other methods on seizure detection with scalp EEG 106

11. Performance comparison of the proposed DCRNN model with other methods on machine fault detection dataset 110

12. Summary of novel contributions 115

LIST OF FIGURES

Figure Page

1. The receptive field mosaic of an actual population of ganglion cells in the retina overlayed on an example natural scene. Red, blue and yellow ellipses refer to multiple ganglion cell types 7

2. An Artificial neuron model. $x_0, x_1 \dots, x_n$ denote the bias and external inputs respectively. The corresponding synaptic weights are denoted by $w_{i0}, w_{i1} \dots, w_{in}$, where i denotes the current neuron and n denotes the input number.....21

3. A multi-layered feed-forward network 23

4. Time-delayed recurrent (Elman) neural network.....24

5. Bidirectional RNN (BRNN) architecture.....26

6. Long Short-Term Memory Unit Signal Flow Diagram 27

7. Simultaneous recurrent network (SRN).....29

8. Cellular neural network architecture. This cellular neural network consist of 16 cells in a 4×4 2D grid arrangement 31

9. CSRN external architecture with an example input pattern.....33

10. The SRN architecture utilized as the core in CSRN 34

11. Seizure classification accuracy (subset of dataset B) and processing time vs. HWPT decomposition level. Level 5 gives the best compromise between seizure detection accuracy and processing time. 46

12. Pipeline of the proposed automated seizure detection algorithm. 49

13. Patient specific seizure detection results.....50

14. (a) An example EEG data segment of a patient 'chb05' used for training. (b) EEG segment of the same patient misclassified by the trained classifier. 51

15. Patient specific average seizure detection latency with minimum and maximum recorded values 52

16. Patient specific false positive score of the proposed seizure detection algorithm 53

17. Pipeline developed for comparison of training algorithms for generalized image processing 72

18. Greyscale to binary conversion results (Primary Test Cases). 80

19. Results for LPF implementation. 81

20. Greyscale image affine transformation results (Primary Test Cases). 83

21. Training time vs. Image size using EKF, UKF, and PSO for image Affine transform 85

22. Proposed DCRNN architecture. Each cell in the cellular sub-architecture hold a configurable LSTM network. Final outputs of each cell is aggregated and passed through a feed-forward network followed by classification. 94

23. The 2D grid mapping of the long-term bipolar EEG montage used in the CHB-MIT scalp EEG Dataset. The 2D grid approximation is a typical input to the proposed DCRNN. 100

24. The 2D time-sequence grid arrangement of the Jefferson Lab cavity fault detection database. 102

25. Summary of patient specific seizure detection performance of the proposed DCRNN model 105

26. Example RF waveforms extracted from cavity 1. 109

27. ROC performance curve of DCRNN for multi-class machine fault detection. 111

CHAPTER 1

INTRODUCTION

The rapid growth in computing power and ever increasing availability of data has made computational intelligence (CI) an important part of human life. In the past, the use of CI has been limited to specific applications in industrial control and robotics. However, recent advancements in CI have affected almost every aspect of human life such as intelligent transportation [1, 2], intelligent diagnosis and health monitoring for precision medicine [3-5], robotics, automation and machine health diagnosis [6, 7], and many others. Computational intelligence plays a critical role in the recognition of patterns in data relevant to a specific task. Consequently, classical pattern recognition has seen major successes in detecting patterns of inputs that are primarily static in time [8]. Typical applications for such static pattern recognition involve classification of an input vector to one of multiple classes, such as recognition of objects in an image, or establishing relationships between observations, such as regression analysis. However, most real world data obtained through an observation or measurement almost always exhibit changes with time. Though in some cases, the change of observations in time can be ignored, certain applications that particularly deal with changes across time require this additional temporal dimension to be incorporated in the pattern recognition process. For instance, many applications involving health monitoring and diagnosis, machine fault detection, speech recognition, and natural language processing require recognition of representative patterns in time. Moreover, certain applications such as processing video data for activity recognition, and monitoring multi-channel EEG for seizure detection, may require recognition of patterns that extend in both spatial and temporal dimensions. Analysis of these time series

requires computational models that are specifically capable of capturing complex patterns in time and space for time series data processing.

Traditional CI methods used in time series data processing involve machine learning models that constitute a pipeline of multiple processing steps that extract representative features from raw data prior to applying a classification or regression based on the application. This intermediate representation of raw data, known as 'hand-engineered' features, requires domain expertise and human interpretation of physical patterns such as texture, shape, geometry, and frequency spectral patterns, among others.

These traditional methods have been shown to perform quite well, especially in situations with limited availability of observations and with appropriate domain expertise. However, there are a few problems that may impede the performance of traditional machine learning methods in computational intelligence. First, the 'hand engineered' features are based on human interpretation; thus, they are heavily dependent on the application and expertise of a human with the specific application. Second, the hand engineered features are extracted from each observation without regard to noise and variation of data. Finally, the hand engineered features may not generalize well for all possible observations.

An alternative to hand engineering of features is automated learning of features using artificial neural networks (ANN). The ANNs essentially mimic the working principles of the biological neurons in the human brain. As such, ANNs have shown an ability to learn complex functions and are dubbed 'universal approximators'. The potential of ANNs is recently being exploited with access to large datasets, efficient training methods, and improved computing resources. Recent advancements in ANN research have led to 'deep learning' (DL) [9, 10] where large scale neural architectures are developed to tackle complex tasks. Most DLmodels are

applied to solve problems related to static data such as visual image recognition. This is due to the fact that the feed-forward information processing observed in generic ANNs is unable to account for the variations of observations through time. Consequently, a special variant of ANNs called recurrent neural networks (RNN) are developed to process time series data. The RNNs incorporate additional neural links that feed the information backward, enabling accumulation of temporal trends through memory. The success of DL and deep recurrent learning is largely due to the following aspects. Deep learning offers end-to-end trainable architectures that jointly learn feature extraction, selection, and classification. Deep learning also offers feature learning, aided by the input data and the classification targets, and avoids hand engineering of intermediate features. Thus, the DL methods often outperform classical machine learning models by dynamically learning optimal features through observations. However, current state-of-the-art DL models face a major limitation. The complexity, and the amount of trainable parameters of the DL models increase proportionally to the complexity of the data processing task. This is further exacerbated in the recurrent learning models where the additional feedback links account for further increase in complexity and trainable parameters. Especially in the presence of time series data that constitute spatial and temporal patterns, such as multi-channel EEG, typical deep recurrent models may be prohibitively cumbersome to utilize effectively. Hence, it is essential to revisit the fundamental ANN architectures and their biological relevance to design more efficient recurrent learning models for complex time-series processing. The following sections summarize current feature engineering methods used for large-scale time-series processing and discuss aspects of neural processing that may yield more efficient ANN architectures.

1.1 FEATURE ENGINEERING FOR TIME SERIES DATA

The two main problems faced in time-series processing for a certain application are 1) selecting an appropriate representation of the time series data and 2) selecting a suitable categorization/regression method [11]. With the complexity of the data and the application, a more efficient intermediate representation of time-series data may be required for improved categorization performance [11, 12]. Moreover, this intermediate representation can be a set of derived properties of data, commonly called 'features', that essentially convert the temporal classification to a more amenable static classification task [13]. These representative features could be a set of simple statistics of the time series data such as mean and variance, skewness, kurtosis, largest peak and number of zero crossings [14]. More descriptive features such as autoregressive coefficients [15], frequency power spectral features [16], and features derived from time-frequency analysis such as wavelet transform [4, 16], wavelet packet transform [4, 17], filter banks [18], and self-similarity features [13], and further hand-crafted versions of these features may also be considered to obtain a more discriminatory representation of data. However, one of the main problems associated with feature engineering is that the potency of such features depend on the data and the task at hand. Therefore, the performance of a machine learning pipeline depends on the hand selection of a subset of the aforementioned features or newly engineered features based on the domain expertise. Another strategy may be to use many feature extraction algorithms to obtain a comprehensive set of features and subsequently perform feature selection to pick the best features [19, 20]. However, the comprehensive feature extraction and additional feature selection step may substantially increase the computational cost. Nevertheless, the biggest limitation of feature engineering is the requirement of human intervention in hand crafting the features that may be relevant to given data and the task. Feature learning with ANN

largely alleviates this problem by progressively learning the best possible discriminatory feature from data.

1.2 TIME SERIES PROCESSING WITH ARTIFICIAL NEURAL NETWORKS

The ANNs have been widely studied since their inception [21] due to their ability to approximate complex functions. With improvements in computational resources and training methods, deep neural networks have been quite successful in solving various problems in areas such as robotics [22], image recognition [23], and text recognition [24]. The typical feed-forward neural networks are predominantly used in processing data that is static in time due to an inability to process temporal relations owing to the limited forward information processing nature. Recurrent neural network (RNN) [25] or a time-delay neural network (TDNN) is a variant of ANN with the added capability of information aggregation through feedback connections. RNNs such as Elman and Jordan architectures [26-28] process time series by reading samples sequentially in time, and the feedback connections aid in retaining valuable information through time steps. Further improvements to the feedback units in retaining memory through longer time-sequences have been made through developments such as Long Short-term Memory (LSTM) [29] units and Gated Recurrent Units (GRU) [30]. Large-scale deep versions of such recurrent neural networks architecture have been successfully utilized for several problem domains [31-34]. Though deep learning has shown superior performance in most problem domains, current state-of-the-art models suffer from a major limitation. Depth, complexity, and amount of trainable parameters associated with these models grow proportionally to the complexity of the input dimensionality and the given task. This problem is further exacerbated in recurrent learning models as the additional feedback links constitute even more trainable

parameters. Therefore, such architectures can grow prohibitively large in the presence of large-scale and multi-source time-series data such as EEG. Therefore, this research focuses on designing efficient neural network architectures to analyze multi-source and time series datasets. Due to the ANNs basis in biological neuronal processing in the brain, we believe it is prudent to look back into the working principles of biological neuronal systems to identify more efficient structural modeling.

1.3 BIOLOGICAL BASIS FOR EFFICIENT INFORMATION PROCESSING

Neurobiologists' findings [35-39] suggest that primate visual brain can be characterized by a set of parallel, multistage systems that are specialized to process information simultaneously. The parallel information processing in the visual system begins from the retina. The retina in the primate visual system is composed of different types of ganglion cells [40]. Each ganglion cell type is thought to tile the retina, providing a complete representation across the entire visual field [41]. The soma of each ganglion cell appears to be regularly spaced across the retina so that, collectively, their dendritic field cover the retina uniformly and with constant overlap as shown in Fig. 1 [35]. This highly ordered mosaic architecture is then linked to parallel pathways into the input layers and compartments of other components in the visual system (e.g.

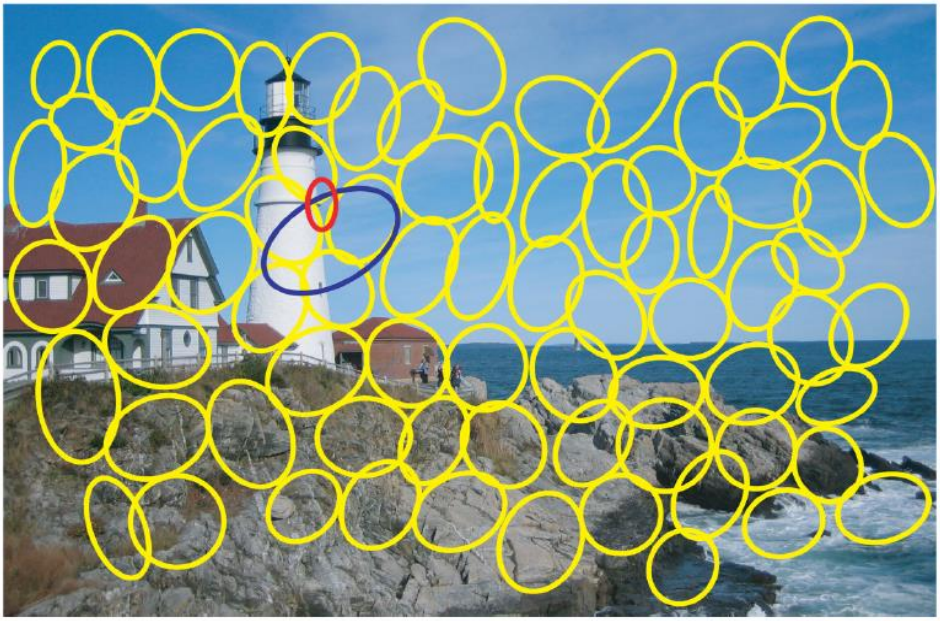

Fig. 1. The receptive field mosaic of an actual population of ganglion cells in the retina overlayed on an example natural scene. Red, blue and yellow ellipses refer to multiple ganglion cell types[1].

primary visual cortex) [35]. Each component processes the information in many different ways and for many different behaviors such as object recognition, scene understanding, etc. [36]. The information received at the primary visual cortex area also encounters a series of simultaneous processing [37]. Therefore, this parallelized 'divide and conquer' type information processing seems to be an integral part of the primate brain evidenced at multiple levels of processing. Consequently, our above observations on the biological neuronal system suggests that integrating some type of a distributed information processing scheme in our computational models may improve the efficacy of the model and also may lead to a more biologically plausible architecture.

[1] Image obtained from [35] J. J. Nassi and E. M. Callaway, "Parallel processing strategies of the primate visual system," *Nature Reviews Neuroscience*, vol. 10, pp. 360-372, 2009. for explanation purposes only.

1.4 CELLULAR SIMULTANEOUS RECURRENT NETWORK

The cellular simultaneous recurrent network (CSRN) is a unique ANN architecture that mimics both the distributed computing properties and the local feedback inherent in the visual processing of the human brain. The CSRN architecture consists of a 2D cellular arrangement similar to the cellular neural networks proposed in [42-44]. This cellularity in the CSRN promotes parallelized processing of inputs that are distributed in 2D space. Each cell of the CSRN consists of a simultaneous recurrent network (SRN). The SRN is essentially a generalized multi-layered perceptron (GMLP) augmented for biologically inspired localized feedback connections called simultaneous recurrency. These biologically inspired properties give the CSRN superior function approximation capabilities and the ability to share resources by sharing trainable weights. The CSRN was first introduced by Pang et al. [45] who demonstrated its capability in efficient approximation of highly dynamical functions by learning a complex 2D maze traversal task. Identifying its unique ability to learn 2D topological relationships, the CSRN is further extended to learn basic image processing tasks such as image transformations [46], image binarization [46, 47], filtering [48], and registration [49].

Though efficient at learning complex functions, the cellularity and the simultaneous recurrency makes CSRN quite a complex neural network architecture. This in turn makes the training and applying modifications to the architecture quite difficult. The CSRN is initially trained with the backpropagation through time (BPTT) algorithm for the 2D maze traversal task in [45]. Though successful, the use of BPTT is time consuming, especially for large scale applications such as image transformation. The SRN training with extended Kalman filter (EKF) is subsequently introduced for such applications [46]. Though the overall training time is improved with EKF, it is still plagued by issues with instability, and the computation of Jacobian

for the EKF algorithm is still performed using BPTT, which inhibits structural changes. Therefore, we still consider the proper training of CSRN to be an open problem that we plan to address within this dissertation.

1.5 FROM MACHINE LEARNING TO ANNS FOR TIME SERIES ANALYSIS

Machine learning (ML) based time series analysis follows the classical method of hand-crafted feature extraction, feature selection, and classification. Consequently, each of these stages in the ML pipeline involves design and implementation of algorithms based on the experience and the domain expertise of the user. For instance, processing of multi-channel EEG for seizure detection may require the use of time-frequency feature extraction methods, such as wavelet transform to capture the discriminatory features. Inevitably, such a process leads to improvements in existing methods and the discovery of new methods and algorithms that may provide further insight into the data and the application domain. Moreover, a contemporary ML pipeline may provide the necessary baseline platform for further research with insight into the current limitations of algorithms and domain knowledge for a specific task. Therefore, it may be prudent to study the contemporary ML methods for large-scale time series analysis as our initial goal.

ANNs have been widely used within ML pipelines as a trainable classification model. Though ANNs are regarded as universal function approximators, the limited scalability and associated difficulties in training severely limited its application for more complex tasks. The recent advent of deep learning and improvements in computing hardware have allowed ANNs to be effective and grow in size and depth at the same time. Such DL models have shown the ability to learn complex tasks, regularly achieving performance superior to contemporary ML methods.

However, typical deep ANN structures tend to grow prohibitively large in the presence of complex, multi-dimensional datasets such as large-scale time series studies in this work. There may also be further issues in effective training of such architectures especially with limited availability of training data and computing resources. Consequently, we believe a comprehensive preliminary study of contemporary ML methods applied to complex data may provide insights that may be beneficial for designing an efficient generalized DL model as a final contribution.

1.6 PROPOSED WORK AND CONTRIBUTIONS

This dissertation proposes to systematically address the prevailing intricacies in efficient analysis of complex multi-dimensional time series data. Consequently, we first investigate the efficacy of conventional machine learning with hand crafted feature extraction for a large scale multi-channel EEG classification task. A novel

ML pipeline is proposed for high accuracy EEG classification with real-time performance, achieving state-of-the-art in ML based EEG processing for seizure detection.

Table 1. Summary of proposed contributions

Research Goal	Topic	Contributions
1	Contemporary machine learning for multi-dimensional time series analysis	Characterization and development of a novel ML based pipeline for high accuracy EEG classification for real-time seizure detection
2	Characterization and generalization of a contemporary cellular recurrent neural architecture and training for highly dynamical function approximation	Comprehensive investigation of training and generalization of the cellular SRN architecture for a multitude of topological image processing applications
3	Development of a novel efficient deep learning architecture for spatially distributed multi-sensor signal time series processing	Introduction of novel bidirectional deep cellular recurrent neural network (DCRNN) architecture for efficient processing of complex multi-dimensional time-series data for multiple application domains

Next, we investigate the efficacy of ANN based feature learning for complex multi-sensor signal data analysis. For this, we first characterize and develop a contemporary cellular recurrent neural network architecture for complex dynamic function approximation tasks. Specifically, we investigate and address the intricacies in generalization and training of the complex cellular SRN architectures for several applications.

Finally, we introduce a novel state-of-the-art bi-directional deep cellular recurrent neural network (DCRNN) architecture for efficient processing of large-scale time series data from spatially distributed multi-sensor systems. We also obtain the generalization of the DCRNN for analysis multi-sensor time series data for different application domains. Table 1 summarizes the proposed contributions of this dissertation.

The research contributions of this dissertation have resulted in several peer reviewed publications as follows. The research findings related to research Goal 1 are published in IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE) [17]. The findings of research Goal 2 contributed to several publications in the proceedings of the International Joint Conference on Neural Networks (IJCNN) [44, 48, 50] and the Society of photo-Optical Instrumentation Engineers (SPIE) [49, 51]. Finally, the overall contributions of research Goal 3 are under preparation to be submitted to IEEE Transactions on Neural Networks and Learning Systems (TNNLS).

1.7 ORGANIZATION OF THE DISSERTATION

This dissertation is organized as follows. Chapter 2 describes the necessary background on the feature engineering methods used for the machine learning framework developed for multi-channel EEG time series analysis. As such, the sub-sections cover brief summaries of harmonic wavelet transform (HWPT) and fractal dimension analysis (FD). The remaining sections provide a brief understanding of artificial neural networks (ANNs), different types of architectures for feed-forward and recurrent learning, and cellular neural networks with a brief introduction to cellular simultaneous recurrent networks and associated learning schemes. Chapter 3 discusses the proposed ML pipeline developed for efficient processing of large-scale EEG time-series data. This chapter details the feature extraction and the multi-level moving window based analysis technique developed for the proposed pipeline, followed be experimental result and a comparison with state-of-the-art techniques. Chapter 4 investigates the efficacy and trainability of the CSRN architecture for several complex topological image processing tasks. This chapter obtains a generalization of CSRN architecture for multiple topological mapping

tasks and analyzes the trainability using several state-of-the-art learning algorithms based on variations in the input dimensionality and the respective cellularity of the model. Chapter 5 discusses the proposed novel Deep Cellular Recurrent Neural Network (DCRNN) architecture for efficient processing of multi-source time series datasets. This chapter demonstrates the generalizability of the architecture for time series data on multi-channel EEG and machine fault classification. The chapter also highlights the efficacy of the proposed architecture cellular weight sharing functionality. Finally, the dissertation concludes with a summary and future work in chapter 6.

CHAPTER 2

BACKGROUND OF THE STUDY

This chapter provides a brief discussion of the techniques required to understand this dissertation. Accordingly, we summarize several hand engineered feature extraction techniques used for time-series processing that is utilized in chapter 3 of this work. We then illustrate the fundamentals of artificial neural networks (ANNs), basic ANN architectures and their training methods, working principles of recurrent neural networks, and the role of feedback connections in time-series data processing. In the next few sub-sections, we briefly go over each of these techniques.

2.1 TIME-SERIES DATA

Time series data is essentially a sequence of values obtained through observations over time. The sequence of observations are usually evenly spaced through time and commonly represented [52] as a vector $= \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$, where each element $x^{(t)} \in R^m$ of X is a vector of m values such that $x^{(t)} = \{x_1^{(t)}, x_2^{(t)}, \dots, x_m^{(t)}\}$. The size of m is determined by the dimensionality of input at time t, and may have an impact on the processing speed the time series based on the algorithm used.

2.2 FEATURE ENGINEERING FOR TIME-SERIES DATA

This section provides details on the feature engineering algorithms utilized in the ML based data processing pipeline developed for the automated seizure detection with multi-channel scalp EEG.

2.2.1 FRACTAL ANALYSIS

Fractal geometry has been exploited as a tool to describe, model, and analyze complex objects or curves found in nature [53]. Fractal dimension (FD) can be used to quantify the complexity of a self-similar pattern [54]. Strict self-similarity is a property characterized by artificially generated mathematical objects such as the Sierpinski triangle [55]. In time series analysis, fractals can be used to measure the complexity of a waveform [56].

This work proposes a seizure detection method employing the box-counting method, a fast and commonly used FD estimation algorithm, for estimating the complexity of the EEG waveform [57]. Let the size of a box be ε, where $(\varepsilon > 0)$, and the minimum number of boxes of size ε needed to cover signal s, be $N_B(\varepsilon)$. The box-counting dimension of s, $N_B(\varepsilon)$ is proportional to $1/\varepsilon^{d_B}$ as $\varepsilon \to 0$ where d_B is the FD of s. This can be expressed using a positive constant k where,

$$\lim_{\varepsilon \to 0} \frac{N_B(\varepsilon)}{1/\varepsilon^{d_B}} = k. \tag{1}$$

Using natural logarithm on Eq. (1) we have,

$$\lim_{\varepsilon \to 0} (\ln N_B(\varepsilon) - \ln 1/\varepsilon^{d_B}) = \ln k. \tag{2}$$

Rearranging Eq. (2) offers FD d_B given as,

$$d_B = \lim_{\varepsilon \to 0} \frac{\ln N_B(\varepsilon) - \ln k}{\ln 1/\varepsilon}. \tag{3}$$

In Eq. (3) as $\varepsilon \to 0$, $(-\ln k \,/\, \ln(\frac{1}{\varepsilon})) \to 0$. Therefore, we have,

$$d_B = \lim_{\varepsilon \to 0} \frac{\ln N_B(\varepsilon)}{\ln 1/\varepsilon}. \quad (4)$$

Equation (4) shows that, if the relationship between $\ln N_B(\varepsilon)$ and $\ln 1/\varepsilon$ is linear, the FD value d_B is obtained as the slope of the straight line formed by $\ln N_B(\varepsilon)$ and $\ln 1/\varepsilon$.

2.2.2 HARMONIC WAVELET PACKET TRANSFORM

The rhythmic patterns in most seizure related EEG occurs within multiple frequency ranges [58, 59]. Therefore, it is essential to accurately capture these spectral features to increase the sensitivity of seizure detection. Moreover, the non-stationary nature of EEG prompts the use of time-frequency analysis methods to extract such features [58, 60, 61].

Accordingly, this work utilizes a variant of wavelet packet transform known as harmonic wavelet packet transform for feature extraction in EEG. Generic discrete wavelet packet transform methods require recursive calculations for systematic signal decomposition into subsequent levels. In contrast, discrete harmonic wavelet packet decomposition is obtained using harmonic kernels similar to the Fourier basis function and does not require recursive calculation to achieve higher frequency resolutions [7].

Wavelet transform is obtained by computing the correlation between the signal *x(t)* and the conjugate $\overline{w}(t)$ of the wavelet function $w(t)$ given as,

$$X(t) = \int_{-\infty}^{\infty} x(\tau)\overline{w}(\tau - t)\, d\tau. \quad (5)$$

X(t) denotes the resulting wavelet coefficients in Eq. (5).

Harmonic wavelet transform, on the other hand, uses the harmonic basis function [62]. The design of harmonic wavelets is based on the idea that its frequency spectrum is restricted to non-overlapping octave bands [63]. The frequency domain expression for the harmonic wavelet [7, 62] is given as,

$$W_{m,n}(\omega) = \begin{cases} \dfrac{1}{2\pi(n-m)} & for\ \ 2\pi m \le \omega \le 2\pi n, \\ 0 & Otherwise \end{cases} \tag{6}$$

where *m* and *n* are wavelet scaling parameters and take real values [30]. The time domain expression for the harmonic wavelet is computed using inverse Fourier transform as follows,

$$w_{m,n}(t) = \int_{-\infty}^{\infty} W_{m,n}(\omega) e^{i\omega t} d\omega \ = \frac{e^{i2\pi nt} - e^{i2\pi mt}}{i2\pi(n-m)t}, \tag{7}$$

where $i^2 = -1$. Equation (7) can be further generalized by adding the wavelet translation parameter *k* such that $w_{m,n}(t) \rightarrow w_{m,n}\left(t - \frac{k}{n-m}\right)$. The harmonic wavelet transform is finally expressed as [7],

$$h_{wt}(m,n,k) = (n-m)\int_{-\infty}^{\infty} x(\tau)\overline{w}_{m,n}\left(\tau - \frac{k}{n-m}\right)d\tau, \tag{8}$$

where $h_{wt}(m,n,k)$ are harmonic wavelet coefficients. Furthermore, the harmonic wavelet transform can be easily computed in the frequency domain by inner product between the signal and conjugate of the wavelet function:

$$H_{WT}(m,n,\omega) = X(\omega)\overline{W}_{m,n}(\omega), \quad (9)$$

where $\overline{W}_{m,n}(\omega)$ is the conjugate of $W_{m,n}(\omega)$. This is the Fourier transform of the harmonic wavelet at scale (m,n), and $X(\omega)$ is the Fourier transform of the signal $X(t)$. As harmonic wavelets are compact in the frequency domain, the harmonic wavelet transform is easily computed using the Fourier transform operation.

The bandwidth of harmonic wavelets is determined by the scaling parameters *m* and *n*. The generic octave band frequency spectrum of harmonic wavelets can be obtained by replacing *m* and *n* with $m = 2^j$ and $n = 2^{j+1}$ such that $(n-m) = 2^j$; where *j* denotes the octave scaling of the wavelet.

Using appropriate values for scaling parameters *m* and *n*, it is possible to generate harmonic wavelets that are scaled to represent different regions in the frequency spectrum with the same bandwidth. This capability is used in deriving the Harmonic Wavelet Packet Transform [7]. Frequency sub-bands for harmonic wavelet packet transform (HWPT), are denoted by 2^j where j is the decomposition level [7]. Therefore, the bandwidth for each sub-band is determined as,

$$f_b = \frac{f_h}{2^j}, \quad (10)$$

where f_h is the highest frequency component of the signal to be transformed, and f_b denotes the bandwidth of each sub-band in Hertz. The Bandwidth of the harmonic wavelet is given by $2\pi(n-m)$ as shown in Eq. (6). Therefore, the wavelet scaling parameters *n* and *m* must be chosen so that HWPT satisfies the following condition,

$$2\pi(n-m) = 2\pi f_b. \tag{11}$$

Then the harmonic wavelet packet coefficients can be found as [7],

$$h_{wpt}(j,l,k) = h_{wt}(m,n,k) \tag{12}$$

where $h_{wpt}(j,l,k)$ denotes the harmonic wavelet packet coefficients of l^{th} sub-band in j^{th} decomposition level, and *k* is the index of the coefficient. The scaling parameters for this computation are selected such that,

$$m = l \times \frac{f_h}{2^j} = l \times f_b \tag{13}$$

and,

$$n = (l+1) \times \frac{f_h}{2^j} = (l+1) \times f_b, \tag{14}$$

$$where\ l = 0,1,\cdots,2^j - 1.$$

In Eqns. (13) and (14), f_b denotes the bandwidth of each sub-band as defined in Eqn. (10). The energy content in each sub-band of harmonic wavelet packet decomposed EEG signals are used as a feature for seizure detection as this effectively captures the oscillatory nature of seizures. The energy at each sub-band of the EEG signal is computed using the harmonic wavelet packet coefficients $h_{wpt}(j,l,k)$ such that,

$$En(j,l) = \sum_{k=1}^{N} \left| h_{wpt}(j,l,k) \right|^2 \quad (15)$$

where $En(j,l)$ denotes the energy of the signal at l^{th} sub-band in j^{th} decomposition level. *N* denotes the total number of HWPT coefficients in each sub-band, and k denotes the index of each coefficient.

2.3 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are inspired by human and animal neural pathways such that the ANNs can mimic the ability to learn and adapt. The ANNs can be considered as universal approximators such that the ANNs have the ability to approximate a given function distribution. The structure of an ANN lightly resembles that of a biological neuronal cluster combining an array of simple processing units, called neurons [64].

The first modeling of an artificial neuron complete with a possible linear or non-linear activation function is given in [21]. Fig. 2 shows an example of a non-linear neuronal. Inputs to the neuron are labeled as, $x_0, x_1 \ldots, x_n$. The input x_0 in the model is a fixed input that is typically set to 1 and provides a weighted external bias to the neuron. Each input to the neuron is multiplied by corresponding weights and summed at the summing junction. The output of the neuron is obtained by transforming the weighted sum of inputs, S_i, via the activation function, $f(\,)$.

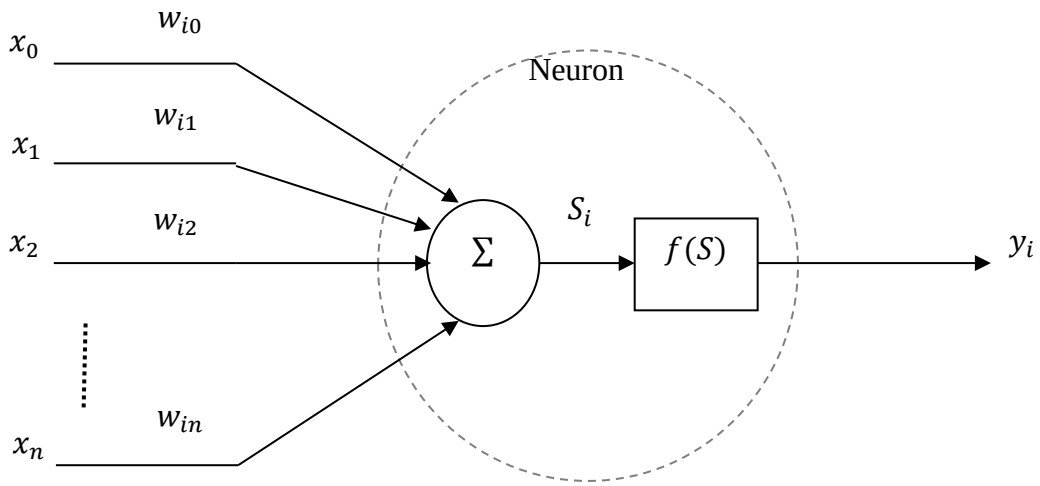

Fig. 2. An Artificial neuron model. $x_0, x_1 \ldots, x_n$ denote the bias and external inputs respectively. The corresponding synaptic weights are denoted by $w_{i0}, w_{i1} \ldots, w_{in}$, where i denotes the current neuron and n denotes the input number.

The model can be mathematically represented as:

$$y_i = f(S_i) = f\left(\sum_{j=0}^{n} w_{ij} \cdot x_j\right) \tag{16}$$

Commonly used linear and non-linear activation functions $f(\,)$ include *sigmoid*, *hyperbolic tangent*, *piecewise linear*, *rectified linear*, etc.

2.4 NEURAL NETWORK ARCHITECTURES

Neural networks are constructed by connecting multiple artificial neurons (such as the model shown in Fig. 2) in some organized fashion. Based on the direction of signal-flow, neural networks can be categorized as *feed-forward* and *recurrent*. In a *feed forward* network, the signal flows only in one direction. However, *recurrent* networks contain feedback connections, enabling signal flow in both directions.

2.4.1 MULTI-LAYERED FEED-FORWARD NETWORKS

Feed forward neural networks that connect neurons in a layered fashion are called multi-layered feed-forward networks (MLFF), or multi-layered perceptrons (MLP) [65]. An example 3-layer MLP is shown in Fig. 3. The layer between input and output layers is called the hidden layer. The neurons/nodes in this layer are referred to as 'hidden' as their outputs are not externally visible. Note that an MLP may contain multiple hidden layers. MLPs are regarded as the most popular type of ANN and have gone through extensive research. The MLPs have been shown to be able to approximate any sufficiently smooth function and are regarded as a universal approximator.

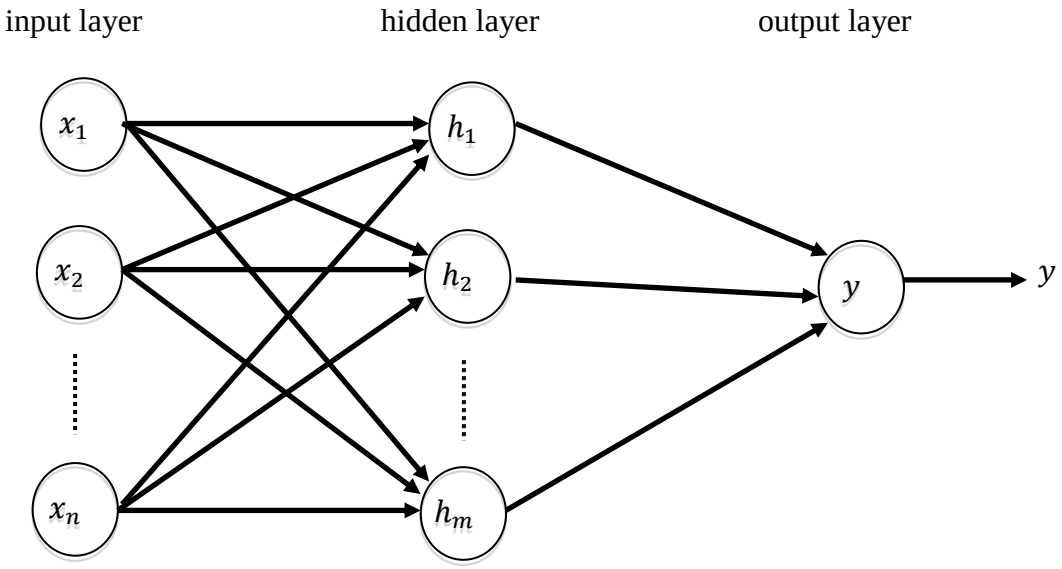

Fig. 3. A multi-layered feed-forward network

At its inception, successful training of MLPs was problematic due to the difficulties in computing the required derivatives. However, the introduction of the rule of back-propagation by Werbos [66], and other related studies [67] resulted in a tractable training algorithm for MLPs.

2.4.2 RECURRENT NEURAL NETWORK

Observe that the information flow of the neural network shown in Fig. 3 occurs in only one direction. However, the biological neural structures in the brain have been shown to exhibit a high level of recurrent behavior [68, 69]. This can be factored into the design of ANNs by considering feedback pathways. Such a neural network that contains one or more feedback connections among neurons is regarded as a recurrent neural network (RNN). In fact, the MLFF shown in Fig. 3 can be converted to an RNN by adding a feedback loop to a hidden layer or an

output layer neuron. Though adding a feedback connection may seem trivial, recurrency in an ANN has a profound effect on its training and performance.

Recurrent neural networks can be categorized into either time-delayed RNN or Simultaneous RNN, based upon the nature of the feedback connection.

2.4.2.1 TIME-DELAYED RECURRENT NEURAL NETWORK

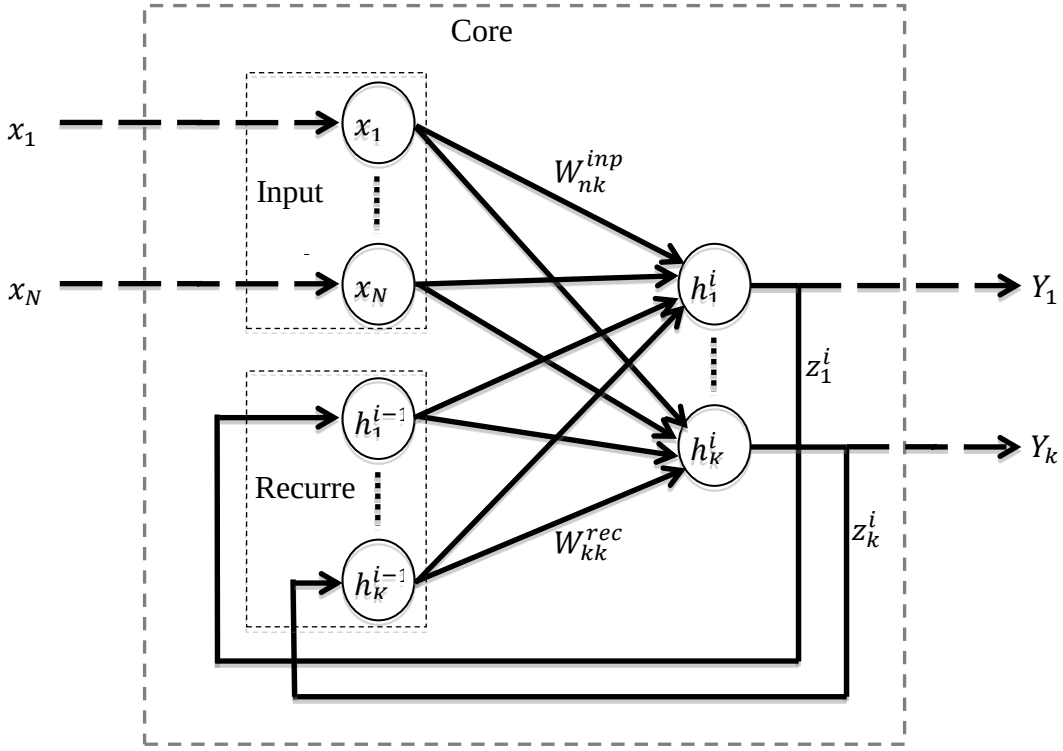

Fig. 4. Time-delayed recurrent (Elman) neural network

Fig. 4 shows a simple Elman time-delayed recurrent neural network (TDRNN) architecture. The Elman TDRNN resembles a feed forward network but with an additional feedback connection from the hidden layer to the input layer with a unit delay. The forward propagation equation for the RNN in Fig. 4 is:

$$Y_k^i = f\left[\left(\sum_{l=1}^{N} W_{lk}^{inp} X_l^i\right) + \left(\sum_{m=1}^{K} W_{mk}^{rec} Y_m^{i-1}\right)\right]' \tag{17}$$

where, Y_k^i is the output from the hidden node k at i^{th} time point in the data sequence. W_{lk}^{inp} is the weight connecting l^{th} input component, X_l^i at time i, with hidden unit h_k^i. Similarly, weight W_{mk}^{rec} connects the previous (recurrent) output component Y_m^{i-1} with the hidden unit h_k^i.

Note that eqn. (17) can be expanded backwards through the data sequence from $i-1$ until the initial element of the input sequence. This reveals that the current output Y^i of the RNN depends on the current input X^i as well as the context accumulated through the previous outputs. This way of processing, however, may not be quite efficient, especially if the whole data sequence is available, as the future context is unused. Moreover, the unidirectional model often fails to form proper responses to the first few inputs of a sequence due to lack of past context.

The bidirectional RNN (BRNN) [70] alleviates these problems by making use of the past as well as the future context of a data sequence. BRNN can be thought of as a straightforward extension of the classic RNN, where it maintains two different recurrent layers for each direction (one for processing from left to right (RNN^{d_1}), the other from right to left (RNN^{d_2})). BRNN then combines the outputs of RNN^{d_1} and RNN^{d_2} using an additional output layer. Fig. 5 shows the general composition of a BRNN.

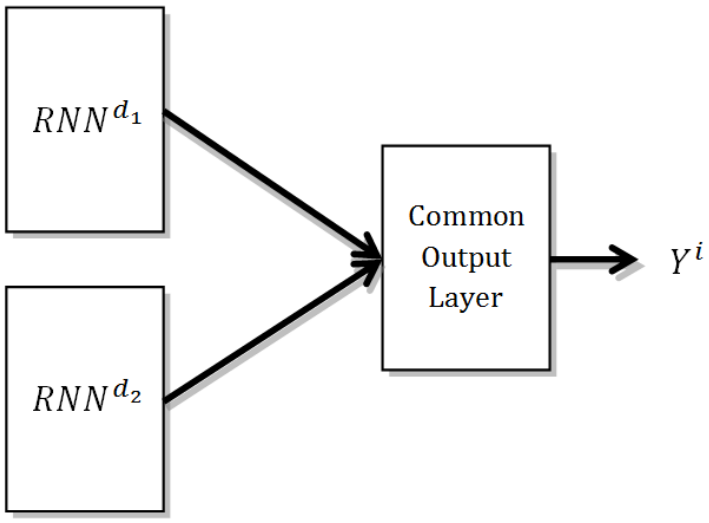

Fig. 5. Bidirectional RNN (BRNN) architecture

The RNN^{d_1} and RNN^{d_2} in Fig. 5 are recurrent neural networks such as the one shown in Fig. 4. The common output layer combines outputs from each directional RNN to produce the final output Y^i for the i^{th} element in the data sequence. Note that this bidirectional setup ensures constant contextual influence in processing any element i of the input data sequence.

2.4.2.2 LONG SHORT-TERM MEMORY NETWORK

The generic recurrent neural networks such as in Fig. 4 face a known problem of limited reach of context over time series data on the network output. This is due to the limited or decaying backpropagation error over long time periods of a given time-series [71]. This can be considered as a vanishing gradient problem over time, similar to the vanishing gradient problem that occurs over depth of a deep network architecture. Consequently, the long short-term

memory is developed to address this vanishing error signal, with the introduction of memory gates that control the flow of context over time [72]. Fig. 6 shows a signal flow diagram of an LSTM unit.

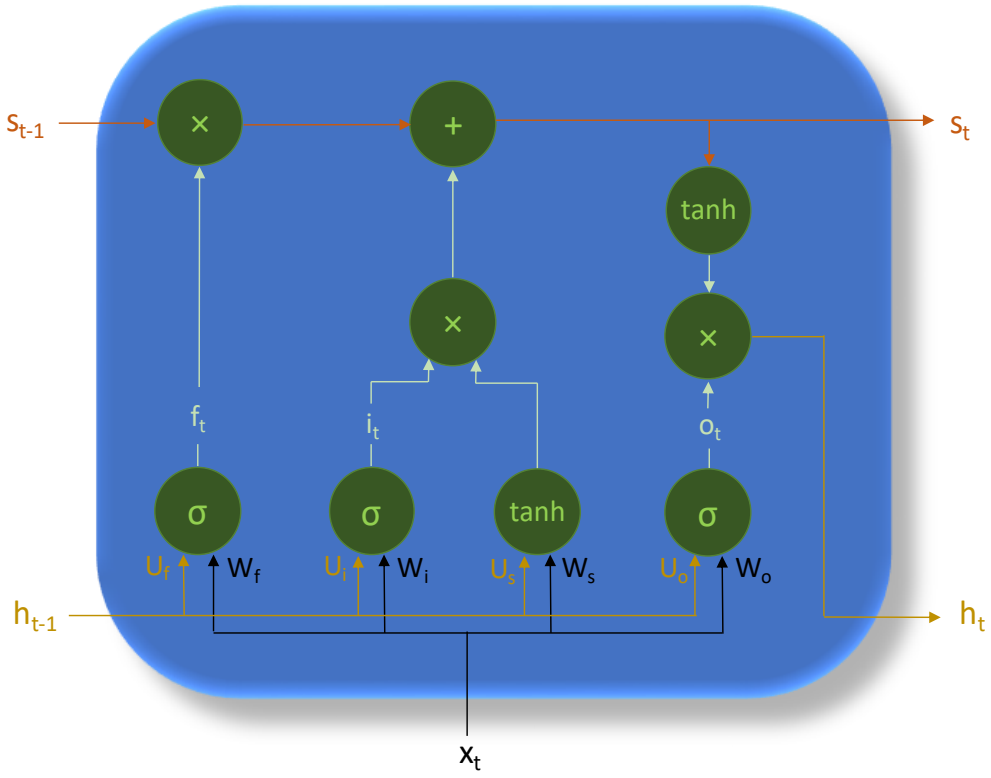

Fig. 6. Long Short-Term Memory Unit Signal Flow Diagram

The full operation of an LSTM unit for a single time step [73] is described in Eqns. (18) to (22).

$$i_t = \sigma(W_i x_t + U_i h_{t-1}), \tag{18}$$

$$f_t = \sigma\big(W_f x_t + U_f h_{t-1}\big), \tag{19}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1}), \tag{20}$$

$$s_t = f_t \odot s_{t-1} + i_t \tanh(W_s x_t + U_s h_{t-1}), \tag{21}$$

$$h_t = o_t \odot \tanh(s_t). \tag{22}$$

Typical input components for an LSTM unit at time step t includes the signal input x_t, hidden output of the previous time step h_{t-1}, and memory accumulated at the previous time step s_{t-1}. The input signal x_t and previous hidden signal h_{t-1} are combined in Eqns. (18) to (20) and passed through a sigmoid activation function to obtain i_t, f_t and o_t. These are known as the "gates" such that if the sigmoid output is near 0, the gate signals have the effect of inhibiting the propagation of the corresponding input signal. Accordingly, the input gate i_t is used to control the effect of the signal input. The forget gate f_t is used to clear the memory. The output gate o_t is used to clear the hidden output. The effect of the three gates i_t, f_t and o_t on the running memory s_t, and the hidden output h_t can be observed in Eqns. (21) and (22). This gate combination in LSTM helps preserve the long term and short term temporal relevance in time series of variable length [72].

2.4.2.3 SIMULTANEOUS RECURRENT NEURAL NETWORK

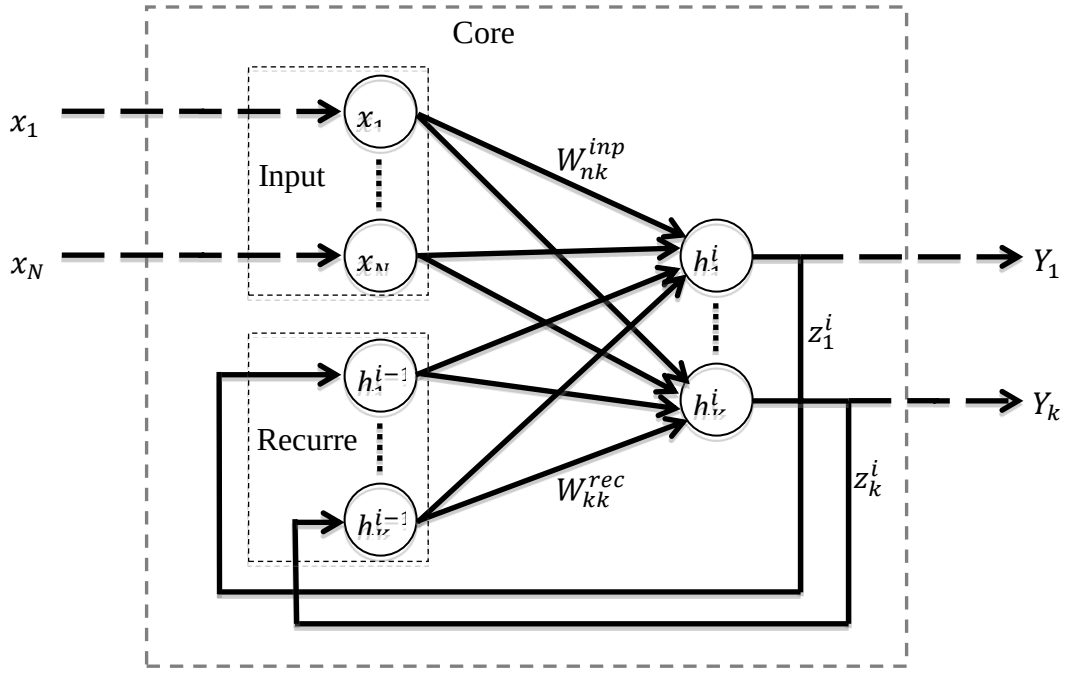

Fig. 7. Simultaneous recurrent network (SRN)

The main difference between TDRNN and simultaneous recurrent network (SRN) is the lack of time delay (Z^{-1}) in the feedback loops, as shown in Fig. 7. The SRN architecture therefore prompts the use of two time scales: external and internal (core) in its data processing scheme. The external time runs slowly, to the point that each output only corresponds to the latest input (such as in the case of a feed forward network). The internal time, however, runs much faster, allowing for the system to run multiple forward propagations until the outputs settle down to the final stable values before being read externally. The forward propagation equations for the forward propagation of an SRN are:

$$z_k^i = f\left[\left(\sum_{l=1}^{N} W_{lk}^{inp} X_l\right) + \left(\sum_{m=1}^{K} W_{mk}^{rec} z_m^{i-1}\right)\right] \quad for\ i = 1\ to\ C, \qquad z_m^0 = 0 \qquad (23)$$

$$Y_k = z_k^C, \qquad (24)$$

where, Y_k is the external output from the hidden node k that corresponds with the data sequence data sequence X. i denotes the core network forward propagation iterations that correspond with the internal time scale, which runs for C steps. W_{lk}^{inp} is the weight connecting l^{th} input component, X_l, with hidden unit h_k^i. Similarly, weight W_{mk}^{rec} connects the recurrent core output component z_m^{i-1} with the hidden unit h_k^i.

The 'settling' response of the SRN is similar to that of a non-linear dynamical system. Therefore, SRNs incorporate the flexibly of performing the tasks of MLPs as well as being particularly proficient in approximating feedback based dynamic systems [74]. Furthermore, the SRN forward propagation process resembles the multiple time scales present in the learning process of the human brain [75].

Note from equations (17) and (23) that forward propagation functions of both TDRNN and SRN involve current input as well as previous/core outputs in the current output of the network. This inhibits the use of standard back-propagation (BP) for training this network. Werbos et al. [76] devised an extension to the BP algorithm termed Back-Propagation Through Time (BPTT) in order to effectively train RNNs. Taking the DTRNN forward propagation function as an example, it can be shown that equation (17) can be expanded backwards through the data sequence from i-1 until the initial value of the sequence. The BPTT algorithm works in a similar manner, where it ‘unfolds’ the network through time. This unfolding process creates a pseudo feed forward network that consists of replicas of the original RNN with the feedbacks

now feeding forward to the following replica of itself. BPTT then uses the standard BP algorithm on the pseudo feed forward network to compute weight updates. However, unlike usual feed forward networks, the pseudo network shares weight among each replica. Therefore, the weights are updated simultaneously at the end, using the sum of all the derivatives corresponding to each weight. The same BPTT algorithm can be applied to SRN by 'unfolding' equation (23).

2.4.3 CELLULAR NEURAL NETWORK

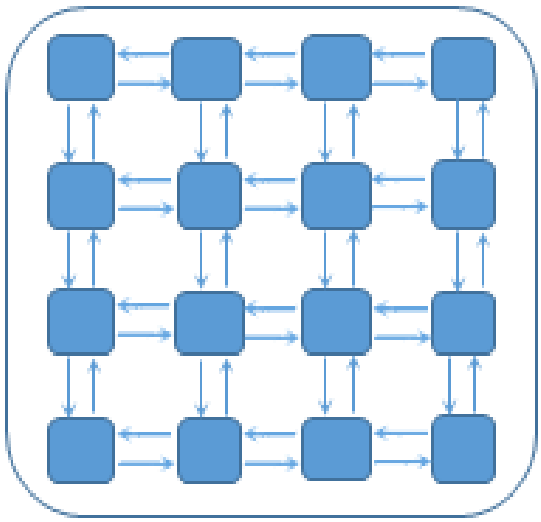

Fig. 8. Cellular neural network architecture. This cellular neural network consists of 16 cells in a 4×4 2D grid arrangement

Cellular neural networks (Cellular ANNs) are another type of ANN architecture that consists of cells that contain identical elements arranged in some geometric pattern. The first 2D Cellular ANN was proposed by Chua et al. in [77]. Each element (cell) of the Cellular ANN

could vary from a simple artificial neuron to a complex ANN. An example cellular architecture is shown in Fig. 8.

Cellular ANNs, such as the one shown in Fig. 8, can efficiently process input patterns that contain some inherent geometric structure, such as images. Each cell of the network processes the corresponding element (e.g. a pixel of an image, a time-series signal from a particular signal source) of the input pattern, enabling efficient distributed processing. Though each element may be processed independently by the corresponding cell, the interactions between neighboring cells allow for processing local geometric patterns in 2D input. The cellular structure also utilizes network weight sharing between each cell. This helps process large input patterns with relatively few training weights.

2.4.4 CELLULAR SIMULTANEOUS RECURRENT NETWORK

The cellular simultaneous recurrent network (CSRN) was first introduced by Pang et al. [78] in an attempt to use ANNs to approximate the solution to the Bellman equation in dynamic programming. The authors use the popular 'maze traversal' problem, a popular dynamic programming example, as the testing environment for their study. The author's findings suggested that generic ANNs such as MLPs are incapable of efficiently learning the 'maze traversal' problem. As a means to solve this problem, the authors successfully utilized an SRN in a cellular architecture, which they referred to as cellular SRN or CSRN.

2.4.4.1 CSRN ARCHITECTURE

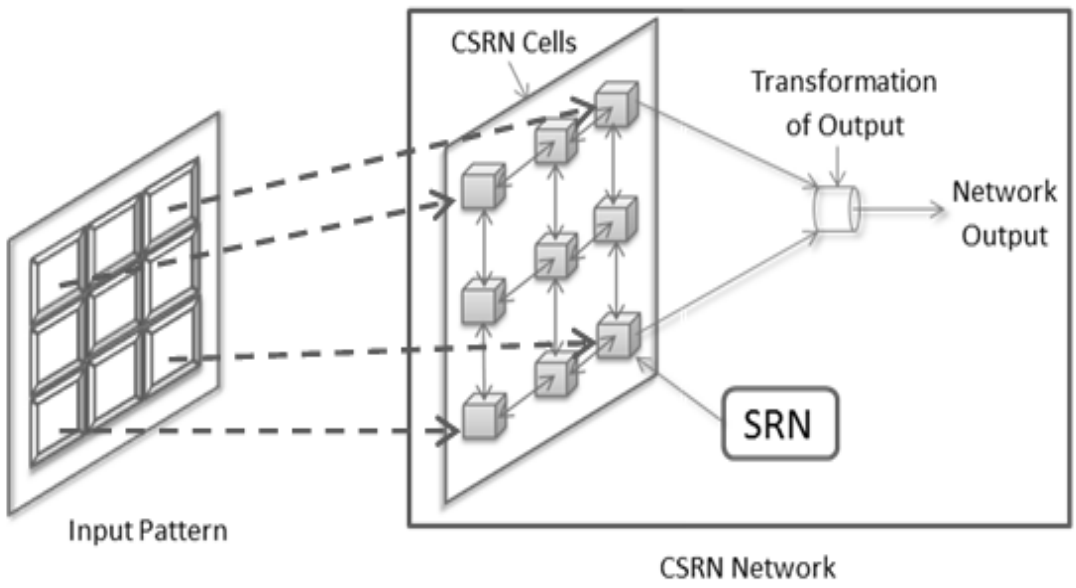

Fig. 9. CSRN external architecture with an example input pattern

Fig. 9 shows the external view of the CSRN. The cellular structure of the network matches that of a 2D input pattern such as a 2D maze or an image. Note that all the attributes of a Cellular ANN architecture (discussed in section 2.4.3) are preserved in the CSRN external architecture. Each cell of the CSRN communicates with its closest neighbors. As mentioned earlier, each cell in the CSRN external architecture contains an SRN as its core network. This core SRN architecture is shown in Fig. 10.

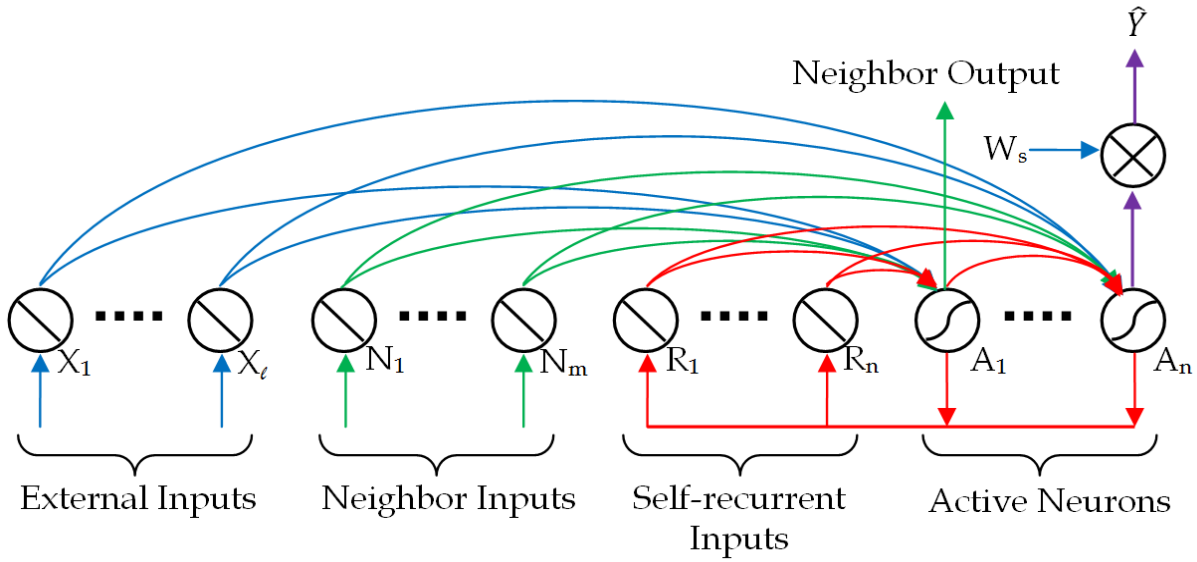

Fig. 10. The SRN architecture utilized as the core in CSRN

The SRN shown in Fig. 10 is slightly different from the usual layered ANN design. This type of architecture is largely known as a generalized multilayered perceptron, and was first introduced by Pang et al. in [78]. This SRN consists of l external inputs, m neighbor inputs originating from neighboring cells, and n number of self recurrent active neurons. The original SRN within the CSRN structure [78] contains 4 closest neighbor inputs and five active nodes. Output from the first active node A_1, known as the connector node is sent to the 4 closest neighbors. All active nodes are self-recurrent as shown in Fig. 10. In this SRN architecture, the inputs to each active node consist of the outputs from all previous nodes.

From a computational stand point, one could easily identify the similarities in the inner workings of the brain and the CSRN architecture. The human brain is a complex, non-linear parallel processing device. It organizes billions of neurons to perform specific tasks. The

organization of these neurons and the associated synaptic activity that is required to perform a specific task are learnt by experience. Neighboring neurons in the brain gather together, forming cell clusters. These clusters represent the basic building blocks of an entire system [69]. Neuroscientists have understood that the locally recurrent behavior plays a critical role in the higher functions of the brain [68, 69]. CSRN architecture mimics the parallel processing, neuronal clustering, and local recurrency of the brain, representing a strong biological basis for it.

2.4.5 CSRN TRAINING

This section summarizes the training algorithms commonly used for training the CSRN for dynamic programming and spatial domain image processing tasks.

2.4.5.1 CSRN TRAINING WITH BPTT

The original CSRN introduced by Pang et al. [78] utilize the back-propagation through time (BPTT) algorithm (briefly discussed in section 0) for training. This method is fundamental to training CSRN, as no other tractable method for the required derivative computation existed prior to BPTT. The cellular structure of CSRN discussed in section 0 requires BPTT to perform the unfolding process through all the cells at each 'iteration through time'. Therefore, the cellular nature of CSRN significantly increases the complexity of BPTT and results in a lower efficiency. Pang and Werbos [12] successfully trained CSRN with BPTT to perform maze traversal. They report that training a single 2-dimentional maze of size 7×7 with BPTT requires approximately 1000 epochs for convergence.

2.4.5.2 CSRN PARAMETER ESTIMATION THROUGH EKF

The Kalman filters, originally proposed by Kalman [79], are commonly used in signal processing applications. The Kalman filter essentially provides a computational means to recursively estimate future states of a system based on past observations. The original Kalman filters are linear recursive filters that estimate the state of a linear dynamic system. For estimation of nonlinear models, an extension of Kalman filters referred to as Extended Kalman Filer (EKF) is used. Parameter estimation through Kalman filters has been utilized in neural network training [80]. In this case, the neural network weights are regarded as the parameters to be estimated, and the neural network outputs are regarded as the observations of the system. The basic idea of EKF is to linearize the system model at each iteration prior to applying the standard Kalman filter [80]. The linear approximation of the nonlinear model is performed by computing the partial derivative matrices (Jacobian) of the model. Since BPTT computes the derivatives of CSRN, the same algorithm is used in EKF to obtain the required jacobian. Ilin et al. in [81] successfully trained CSRN with EKF for maze navigation. The authors report a large reduction in training time with EKF. For example, CSRN trained with EKF for a single maze converges within 15 to 30 epochs while that with BPTT requires approximately 1000 epochs. This large reduction in training time is regarded as a breakthrough in training CSRN, as it expanded the use of CSRN from mere maze navigation to more complex image processing applications.

However, there are a few major drawbacks in using EKF to train CSRN. Mainly, the Jacobian computation via BPTT can be quite complex due to the unique structure of the CSRN. This inhibits even minor structural changes to the network. Also, as the number of cells in CSRN is increased to account for more inputs, the size of the Jacobian matrix increases accordingly. Consequently, for a large number of network inputs, which is very common for large scale maze

and image data, computational complexity of computing Jacobian in EKF training becomes prohibitive [46]. Furthermore, the linear approximation of the nonlinear system model could introduce errors in the estimation process [80] of EKF, which may adversely affect convergence of the network. These drawbacks necessitate further research into finding a training algorithm that is better suited for training a complex universal approximator such as the CSRN.

Current Applications of CSRN

The original application of CSRN as discussed in section 0 is to solve the maze navigation problem by approximating the solution to the Bellman equation [78]. Further studies of CSRN for maze traversal tasks such as in [81, 82] demonstrated its capability to learn complex topological relations. Realizing that topological mapping tasks govern most fundamental image processing tasks, Anderson et al. [46, 83] introduce CSRN to learn image affine transformations. Apart from image affine transformations, a generalized version of CSRN has been shown to efficiently perform image gray-scale to binary conversion, and image low-pass filtering.

CHAPTER 3

EFFICIENT MACHINE LEARNING PIPELINE FOR REAL TIME EPILEPTIC SEIZURE DETECTION USING SCALP EEG

3.1 CHAPTER OVERVIEW

This chapter proposes a novel machine learning pipeline for large scale time series data processing. The proposed pipeline is applied as a patient-specific real-time automatic epileptic seizure onset detection method, using both scalp and intracranial EEG. The proposed technique obtains harmonic multiresolution and self-similarity-based fractal features from EEG for robust seizure onset detection. A fast wavelet decomposition method, known as harmonic wavelet packet transform (HWPT), is computed based on Fourier transform to achieve higher frequency resolutions without recursive calculations. Similarly, fractal dimension (FD) estimates are obtained to capture self-similar repetitive patterns in the EEG signal. Both FD and HWPT energy features across all EEG channels at each epoch are organized following the spatial information due to electrode placement on the skull. The final feature vector combines feature configurations of each epoch within the specified moving window to reflect the temporal information of EEG. Finally, Relevance Vector Machine (RVM) is used to classify the feature vectors due to its efficiency in classifying sparse yet high dimensional datasets. The proposed method is evaluated using two publicly available long term scalp EEG (dataset A) and short-term intracranial and scalp EEG (dataset B) databases. The results demonstrate that the proposed method is effective with both short and long-term EEG signal analyzes recorded with either scalp or intracranial modes respectively. Finally, the use of less computationally intensive feature extraction

techniques enables faster seizure onset detection when compared to similar techniques in the literature indicating potential usage in real-time applications.

3.2 LITERATURE REVIEW

This section provides a brief literature review of hand engineered feature extraction methods used on EEG time-series data for seizure detection.

3.2.1 FEATURE EXTRACTION BASED ON WAVELET TRANSFORM

Early attempts have led to the development of detection systems with accuracies within 76-90% and false detection rates within 1-0.71 h^{-1} [84, 85]. This suggests the need for more computationally efficient methods capable of extracting oscillatory features over different frequencies and self-similar EEG patterns for more effective seizure detection.

Wavelet transform for processing EEG has been pursued in automatic seizure detection due to its ability to obtain non-stationary time-frequency analysis of signals. Saab et al. [60] have introduced a seizure detection method for long-term scalp EEG based on simple Duabachies wavelet transform, achieving a sensitivity of 76%, false detection rate of 0.34 h^{-1}, and median detection delay of 10 s. Similarly, many studies [4, 61, 85, 86] have utilized wavelet and wavelet packet transorm (WPT) based features with varying success in automated seizure detection. Though Wavelet based features have been effective in characterizing seizure events, generic WPT algorithms may be computationally expensive due to recursive computation. Similarly, other typical time-frequency analysis methods such as short-time Fourier transform are also used for seizure detection [87].

3.2.2 SELF-SIMILARITY FEATURES WITH FRACTAL DIMENSION

Fractal dimension has been used in time series analysis of signal complexity in physiological signals [88-91]. Accardo et al. [89] have studied the ability of fractal dimension (FD) to characterize EEG recordings corresponding to different physiopathological conditions. The authors use Higuchi's [92] algorithm to compute FD of EEG, and show that non-linear fractal analysis offers better temporal resolution than Fourier analysis. Polychronaki et al. [93] have performed a comparison of FD estimation algorithms such as Katz's [94], Higuchi's [92], and K-nearest neighbor [95] for epileptic seizure detection. The authors show that K-nearest neighbor offers the best seizure detection performance. The FD estimation algorithms are generally faster with minimal computational cost and provide high seizure detection accuracy. This makes FD attractive for potential real-time applications. However, the high seizure detection latency and false positive rates associated with FD computation suggest that FD features alone may not be adequate for robust seizure detection [93].

3.2.3 FEATURE CLASSIFICATION WITH MACHINE LEARNING

Several recent studies highlight the importance in examining the functional brain connectivity in successful seizure detection and localization [96, 97]. Mierlo et al. [96] discussed the effectiveness in multivariate analysis of EEG to capture the changes in the spatial domain at the onset of a seizure for early detection. Furthermore, the evolution of spatial connectivity patterns in the brain at the onset of a seizure may also provide valuable cues for early detection [98]. The importance of this has led to several studies that utilize network based approaches to capture such connectivity patterns, as reviewed by Yaffe et al. [97]. They highlight the

importance of considering the temporal and spatial evolution of seizure activity for improved accuracy and early detection.

Most detection methods in the literature use ML techniques such as artificial neural networks (ANN) as the final step to classify extracted features into the seizure and non-seizure events. Yuan et al. [99] have presented an EEG classification using an extreme learning machine and non-linear features [100] with an overall 96.5% accuracy. Shoeb et al. [59] also present a patient specific automated seizure detection method that utilizes a support vector machine (SVM) for feature classification. The method achieves 96% sensitivity with a mean delay of 4.6 s and a median false detection rate of 0.083 h^{-1} when evaluated using the CHB-MIT scalp dataset [101]. Santaniello et al. [102] propose a 'dynamic' seizure detector that utilizes sequential hidden Markov model estimations of the singular value in long-term intracranial EEG. The model adapts to changing neural activity over time and obtains a mean false positive rate of 1.39 h^{-1} and an average delay of 9.6 s while tuned for 100% sensitivity. In another study, Acharya et al. [103] have evaluated 7 different classifiers in detecting epileptic seizure onset. The highest classification accuracy of 98.1% is recorded using a fuzzy classifier. Many of these classifiers require estimating initial network parameters such as the number of free parameters, active neurons, and activation function for ANN and parameters related to hyperplane computation for SVM.

3.2.4 SEIZURE DETECTION WITH EEG TIME SERIES DATA

EEG can be segmented into intracranial and scalp recordings, based on the location of electrode placement. Intracranial EEG is recorded using electrodes placed invasively under the scalp. Scalp EEG is recorded non-invasively and is therefore more popular and more widely used

in clinical applications [86]. However, scalp EEG is sensitive to noise and tends to accumulate artifacts. Therefore, additional artifact reduction techniques and algorithms are used to reduce false detections in analysis. Though attempts have been made to develop algorithms for seizure detection using both scalp and intracranial EEG, a reliable automated algorithm with low computational complexity for real-time application is still needed. Applications based on generic wavelet decompositions and filter banks require a better compromise between computational complexity, efficiency, and accuracy.

3.3 PROPOSED REAL-TIME EPILEPTIC SEIZURE DETECTION PIPELINE

Most epileptic seizures are associated with oscillatory patterns [60, 84, 104]. Therefore, distinctive oscillatory patterns in EEG channels can be used as an indicator of an ictal state. However, EEG signals can also contain non-seizure related oscillatory patterns stemming from various activities of the brain. This inhibits the sole use of oscillatory patterns for seizure detection. Therefore, it is necessary to consider other discriminative features to increase the accuracy of seizure detection. Several studies [93, 105] have shown that seizure activity alters self-similarity characteristics in EEG, establishing fractal estimations as a probable feature in seizure detection. Spatial and temporal properties of seizure onset and progression also help to detect a seizure at an early stage. The proposed seizure detection algorithm based on fractal dimension and harmonic wavelet packet transform considers the oscillatory patterns as well as the self-similarity characteristics of seizure EEG signals.

3.3.1 EEG DATA

This study uses two datasets, named Dataset A and Dataset B, for evaluation of the proposed methods. The first dataset (Dataset A) is CHB-MIT EEG database [101]. Dataset A consists of long-term bipolar referenced EEG recordings from pediatric patients with intractable seizures. The 23 subjects in the database consist of 5 males between ages 3 to 22 years and 17 females of ages between 1.5 to 19 years. The case 'chb24' is a later addition to the database with an unspecified age and gender. Case 'chb21' is a second EEG recording of subject 'chb01' obtained after 1.5 years. Due to the large time gap between the recordings, 'chb21' is considered as a separate case. For each subject, the long-term EEG is recorded in continuous segments of 1 to 4 hours. All EEGs are sampled at 256 Hz at 16-bit quantization. Most cases contain 23 bipolar EEG signals derived from electrodes placed according to the International Federation of Clinical Neurophysiology 10-20 placement system. Therefore, only the recordings that contain the 23 EEG channels (FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FZ-CZ, CZ-PZ, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, T8-P8, P8-O2, P7-T7, T7-FT9, FT9-FT10, FT10-T8, T8-P8) are used in the analysis. The EEG recordings of ‘chb16’ are excluded from the analysis due to the unusually short seizure duration with an average ictal length of only 8.6 s. Table 2 summarizes the patient specific information such as the amount of EEG used in the study, the number of seizure events, and seizure duration of Dataset A. The seizure duration is presented for individual seizure events if the total number of events per patient is less than or equal to 4. Otherwise, the mean duration with standard deviation (stdev) is presented.

Table 2. Long term EEG dataset (Dataset A)

Patient	EEG Used (h)	Number of Seizures	Seizure Duration (sec) mean ± stdev, or individual	Patient	EEG Used (h)	Number of Seizures	Seizure Duration (sec) mean ± stdev, or individual
chb01	20	7	63.1 ± 30.5	chb12	10	14	40.7 ± 23.5
chb02	15	3	82, 81, 9	chb14	24	8	21.1 ± 8.7
chb03	20	7	57.4 ± 8.4	chb15	32	20	99.6 ± 53.6
chb04	15	4	49, 111, 102, 116	chb17	15	3	90, 115, 88
chb05	19	5	111.6 ± 9.4	chb18	17	6	52.9 ± 14.4
chb06	24	10	14.1 ± 2.8	chb19	14	3	78, 77, 81
chb07	32	3	86, 96, 143	chb20	20	8	36.8 ± 6.2
chb08	20	5	183.8 ± 49.2	chb21	20	4	56, 50, 81, 12
chb09	32	4	64, 79, 71, 62	chb22	15	3	58, 74, 72
chb10	24	7	63.9 ± 17.2	chb23	20	7	60.6 ± 32.6
chb11	11	3	22, 32, 752	chb24	22	16	31.9 ± 18.4

The second dataset (Dataset B) is an EEG dataset shared freely by the University of Bonn [100]. The EEG recordings in this database are divided into five sets (A-E). Each set consists of 100 segments of artifact free single channel EEG recorded for 23.6 s. Sets A and B are scalp EEGs recorded from five healthy volunteers where A is recorded with 'eyes open' and B is recorded with 'eyes closed' conditions. Sets C, D, and E consist of intracranial EEG (IEEG) recordings obtained for pre-surgical evaluation from five patients with temporal lobe epilepsy. Set D consists of EEGs from the epileptogenic zone and set C from the hippocampal formation

of the opposite hemisphere. EEGs of both C and D are recorded in inter-ictal intervals. EEGs in set E are recorded at seizure activities from all recording sites showing ictal activity. All the recordings are digitized and amplified using a 128 channel system with the average common reference method. The original sampling rate for all EEGs in Dataset B is 173.61 Hz. We up-sample the EEG segments to 256 Hz for seamless analysis using the pipeline developed in this work.

3.3.2 FEATURE EXTRACTION

The scalp EEG data is filtered using a band-pass filter between 3 Hz and 32 Hz since most seizure activity at ictal state occurs within this frequency range [60, 84]. On the contrary, the IEEG in dataset B is filtered between 3 Hz and 80 Hz. The EEG activity from 0-3 Hz is typically associated with frequent non-ictal sleep patterns [60] rather than seizure activity. A zero phase 4th order Butterworth filter is utilized to filter EEG data from all EEG channels. The EEG signals are segmented prior to feature extraction in such a way that the spectral and spatial features are captured accurately. Since the desired frequency range is less than 30 Hz, segmenting EEGs into very small epochs is unnecessary. Consequently, an epoch window length of 2 s is considered in this study as this represents the minimum stationary interval for EEGs [106]. Furthermore, for EEGs sampled at 256 Hz, the 2 s epoch window offers a desirable number of samples per epoch that provides sufficient decomposition depth for an effective wavelet analysis [60, 86]. Changes in EEG at ictal state may occur in multiple frequency components. The spectral features are captured by performing HWPT at decomposition level 5, with (L = 2 s) non-overlapping epochs of each EEG channel. This decomposes the signal epoch into frequency sub-bands of bandwidth 4 Hz. HWPT decomposition level 5 is chosen as the best

compromise between accuracy and processing time using a preliminary experiment with a subset of dataset B as shown in Fig. 11. Note the observation from Fig. 11 can be directly applicable for both scalp and intracranial EEG as the epoch duration does not change. The energy of each sub-band spanning from 4 Hz to 32 Hz (for scalp EEG) or 4 Hz to 80 Hz (for IEEG) is calculated as shown in Eq. (15). The fractal analysis is also performed simultaneously on 2 s epochs of each EEG channel. The spectral and fractal features are then concatenated to form a feature vector of size *M*.

Spatial disperse patterns at the onset may provide important information in differentiating a seizure from other brain activity. Therefore, spatial features must be incorporated in the seizure

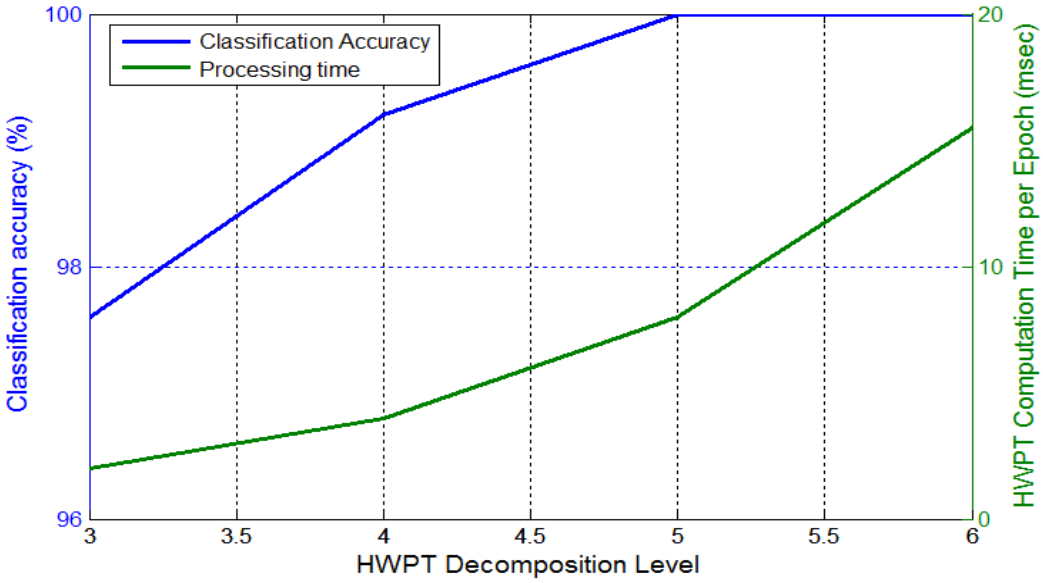

Fig. 11. Seizure classification accuracy (subset of dataset B) and processing time vs. HWPT decomposition level. Level 5 gives the best compromise between seizure detection accuracy and processing time.

analysis. To capture both spectral and spatial features of EEG, feature extraction is performed across all EEG channels. The *M* spectral features extracted from each EEG channel on *L* s epoch

at time $t = T$, are concatenated together to form a combined feature vector $\boldsymbol{X_T}$. If the EEG data is recorded using N number of channels, the combined feature vector contains $M \times N$ elements. Though $\boldsymbol{X_T}$ captures both spectral and spatial information of EEG, it does not contain information about the progression of a seizure. Progression information is important as the goal is to detect an epileptic seizure as early as possible. Medical personnel typically monitor abnormal activity in an EEG over 6 to 10 s before declaring a seizure event [59, 107]. As such, the temporal information is encoded into the feature vectors as follows: (1) Declare a moving window W of length 6 s, (2) Segment the EEG data within this window into 3 non-overlapping (L = 2 s) epochs, and (3) Perform spectral and spatial feature extraction on each of the EEG epochs as described above. (4) Move the window W by 1 s (5 s overlap) and repeat (1) to (4). Consider that window W is captured at time $t = T$. Then, the three feature vectors in W are given as, $\boldsymbol{X_T, X_{T-L}, X_{T-2L}}$, where L is 2 s non-overlapping epoch. Concatenate these feature vectors to form the final feature vector $\boldsymbol{F_T}$ as follows,

$$\boldsymbol{F_T} = [\boldsymbol{X_T},\ \boldsymbol{X_{T-L}},\ \boldsymbol{X_{T-2L}}] \tag{25}$$

Therefore, the final feature vector F_T contains $M \times N \times 3$ elements. The window W is set for 6 s as a compromise between seizure detection accuracy, delay, and processing speed. A similar feature vector formation method is introduced by Chan et al. [108] for seizure onset detection using offline intracranial EEG. Shoeb et al. also [59] show that this method successfully captures temporal information with scalp EEG.

3.3.3 FEATURE CLASSIFICATION

The classification of feature vectors to seizure and non-seizure classes is performed by an RVM classifier. A Radial basis function is chosen as the kernel function to generate non-linear decision boundaries. The RVM is trained with seizure feature vectors derived from the first 20 s of seizure events recorded from the patient under evaluation [18] so that the seizure onset is accurately characterized. The performance is analyzed using patient-specific leave one out cross-validation scheme. Dataset A typically contains 1hour segments of patient specific EEG that contains seizure and inter-ictal activity. Consider EEG recordings of one patient contain S number of seizure events. The classifier is trained using EEG segments which typically amount to a minimum of 10 hours. The training data consists of $(S-1)\times 20$ seconds of seizure EEG along with ~10 hours of non-seizure inter-ictal EEG segments. The testing is done on the EEG segment containing the seizure record that is withheld from the training set including the surrounding inter-ictal data that typically exceeds ~1 hour. This is repeated S times so that each seizure record along with all associated inter-ictal data is tested once. With this strategy, the testing and training sets are never overlapped. In Dataset A, the duration of a seizure event varies, even for the same patient. Consequently, we determine that two seizure notifications within a 150 s interval belong to the same seizure event.

The overall pipeline of the proposed seizure detection algorithm is shown in Fig. 12.

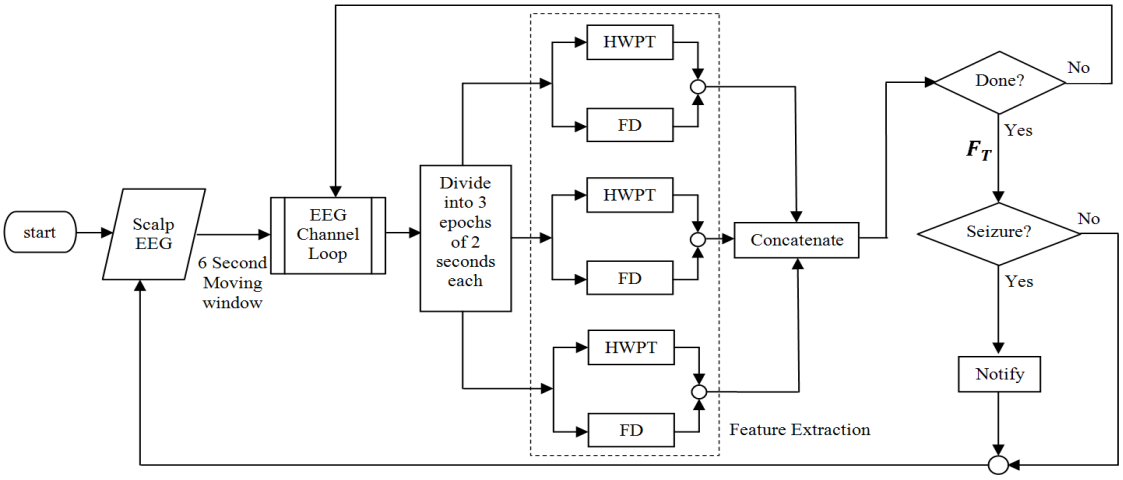

Fig. 12. Pipeline of the proposed automated seizure detection algorithm.

3.4 RESULTS AND DISCUSSION

The performance of the proposed automated seizure detector is analyzed in terms of three different criteria: (1) seizure detection sensitivity, (2) number of false positives (specificity), and (3) seizure detection delay (latency). Sensitivity is the percentage of seizures correctly detected in the test dataset. Specificity is computed by the number of times the algorithm misclassifies a feature vector as a seizure event per hour. Latency is the time delay (in s) between the seizure detected by the algorithm and the seizure onset marked by an expert.

The analysis using Dataset A is presented and discussed in sections 3.4.1 to 3.4.4. The results and discussion of the analysis using Dataset B are presented in section 3.4.5. The proposed algorithm is implemented using MATLAB 8 with a workstation: Intel Xeon 3GHz and 32 GB of RAM.

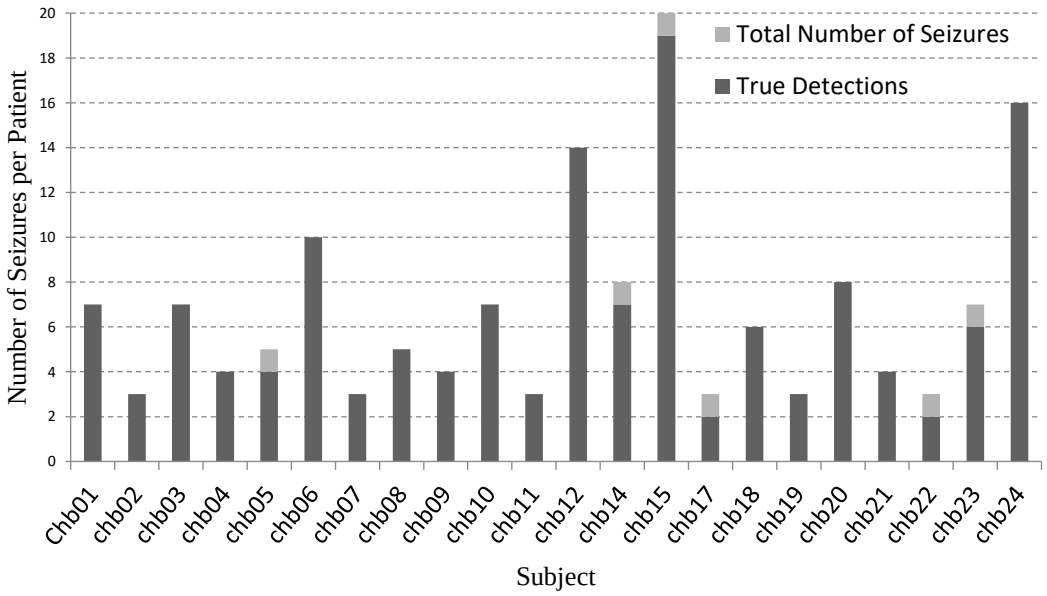

Fig. 13. Patient specific seizure detection results

3.4.1 SEIZURE DETECTION SENSITIVITY

The proposed method successfully detects 144 of 150 seizure events. This amounts to a detection sensitivity of ~96%. Fig. 13 shows the patient-specific seizure detection results with the total number of seizure events used for testing per patient and corresponding successful detections. The algorithm has misclassified only six seizure events specifically from patients 'chb05', 'chb14', 'chb15', 'chb17', 'chb22', and 'chb23'.

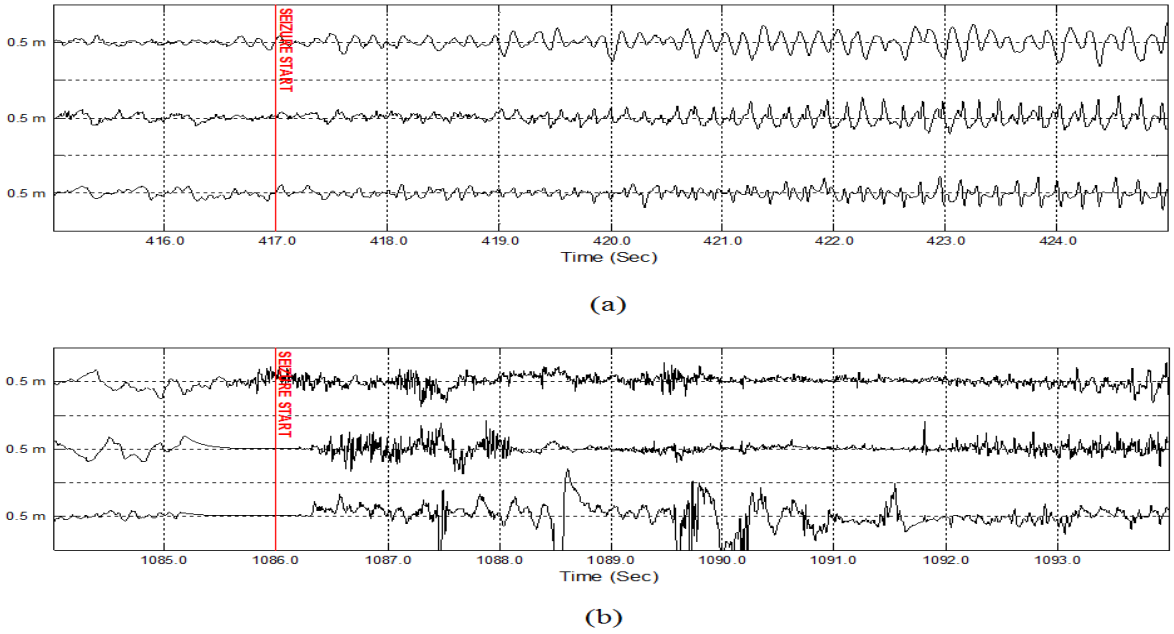

Fig. 14. (a) An example EEG data segment of a patient 'chb05' used for training. (b) EEG segment of the same patient misclassified by the trained classifier.

Fig. 14 (a) shows an EEG segment of an example seizure event commonly seen for patient ‘chb05’, while Fig. 14(b) shows the uncommon misclassified seizure event of ‘chb05’. Close observation of Fig. 14 suggests that the seizure characteristics of the two events are quite different from one another. For example, Fig. 14 (b) shows many high-frequency fluctuations and sudden spikes at the onset. In contrast, a common seizure event of 'chb05' as in Fig. 14 (a) shows low-energy rhythmic activity at the onset which gradually intensifies as it progresses.

The seizure detector inherently assumes that characteristics of seizure EEG do not change over different events of the same patient. Therefore, since the detector is largely trained with EEG events like that of Fig. 14 (a), the probability of misclassifying a seizure with significant deviations of characteristics such as in Fig. 14 (b) is rather high.

3.4.2 SEIZURE DETECTION LATENCY

As the goal of this study is to detect the onset of seizures, the detection latency is critical [109]. The EEG segmentation and feature extraction methods discussed in Section 3.3.2 are specifically utilized to capture the formation and evolution of seizure events with the hopes of reducing the detection latency. The proposed algorithm achieves a median seizure detection latency of 1.34 s across all EEG patient data with more than 70% of all seizures being detected with less than or equal to 2 s after the onset. Average seizure detection latency along with the minimum and maximum recorded latencies per patient is shown in Fig. 15.

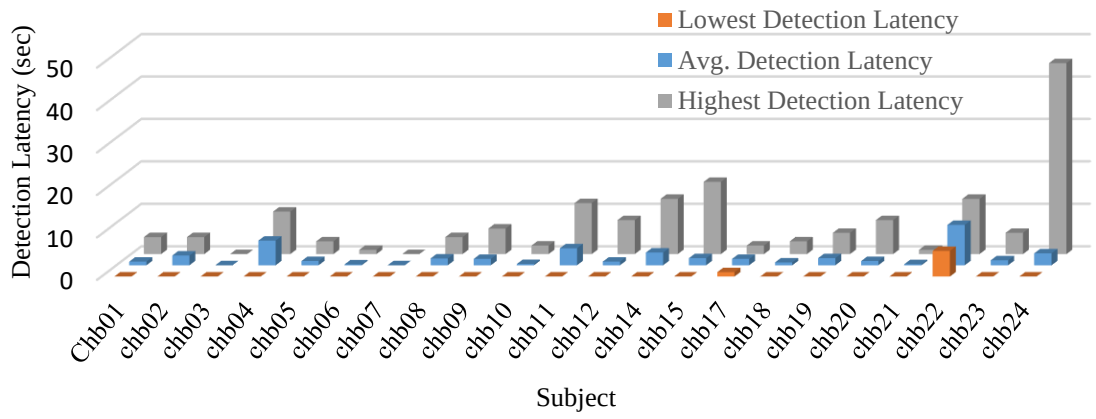

Fig. 15. Patient specific average seizure detection latency with minimum and maximum recorded values

Instances of 0 second minimum latency are achieved in most cases except for ‘chb17’ and ‘chb22’. The consistently low average latencies across the patients show that the instances of high latency recordings are quite rare. This is especially true in the case of ‘chb24’ in which one seizure event is classified at a latency of 45 s while all other events are classified at 0 sec. Along

with sensitivity, the latency also depends on representative seizure onset characteristics in the training dataset. A slow starting seizure with its oscillatory and fractal characteristics that develops over time or a seizure onset with a large number of artifacts can easily induce a delay in classification. The seizures associated with 'chb22' records the largest average detection delay of 9.5 s, while 91% of patient wise average detection delays are less than or equal to 4 s.

3.4.3 SEIZURE DETECTION SPECIFICITY

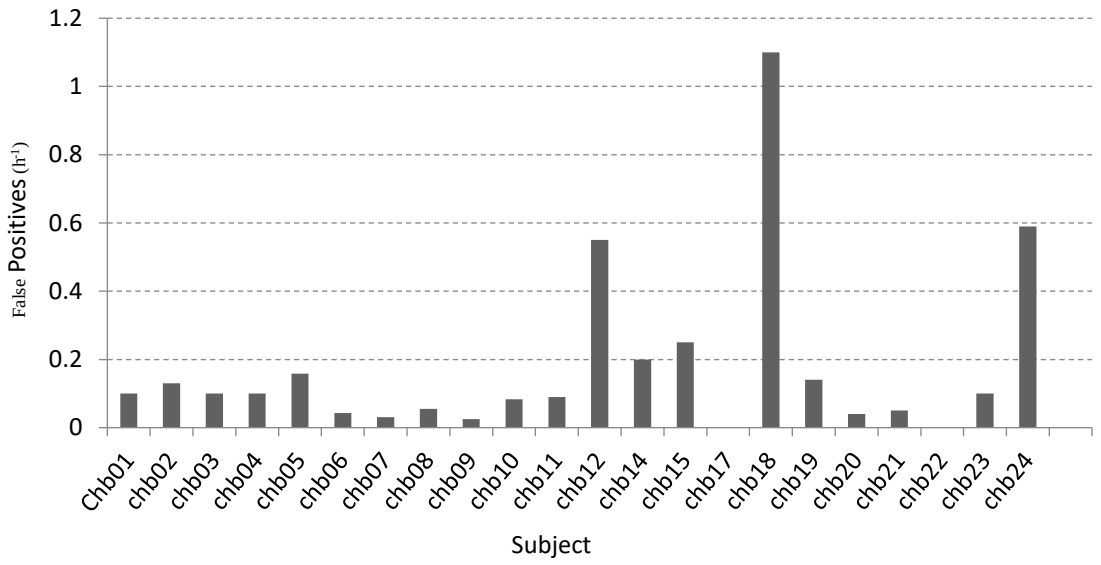

Fig. 16. Patient specific false positive score of the proposed seizure detection algorithm

A good seizure detector must achieve maximum seizure detection sensitivity while minimizing false detections. Even with very high sensitivity, a large number of false alarms can be equally problematic for the patient as well as the medical staff in real life applications. Fig. 16 summarizes the number of false seizure detections per hour for each subject. The patient-wise

specificity (h^{-1}) is computed as the total number of false detections that occur in cross-validation steps, divided by the total duration of EEG (in hours) used in the patient-wise cross-validation analysis. The median false detection rate across all subjects is 0.1 h^{-1}. Note that the algorithm records a very low false positive rate of less than 0.2 for approximately 70% of patients as shown in Fig. 16. However, for EEG data of some subjects, such as in the case of 'chb18', the algorithm produces relatively high false positive rates. The cause of false positives in seizure detection can be explained as a problem of over sensitivity. As discussed in section 3.3.2, the algorithm utilizes the oscillatory and self-similarity (fractal) characteristics of seizure EEG recordings as features for seizure detection. However, it is possible for recordings to contain non-seizure EEG segments exhibiting oscillatory and fractal characteristics similar to a seizure event in a subject.

3.4.4 PERFORMANCE COMPARISON WITH OTHER METHODS

The proposed algorithm is geared towards seizure detection using long-term continuous recordings of scalp EEG, such as those included in Dataset A. Therefore, this section compares the performance of our algorithm with relevant methods in the literature that also analyzes long term EEGs. Several seizure detection methods are chosen based on the criteria that the algorithms obtain seizure detection utilizing long-term scalp EEG data. However, this must be regarded as a qualitative comparison, as the majority of these methods except for the study of Shoeb et al. [59, 110] utilize EEG datasets that are not publicly available and not tested on Dataset A. The performance metrics are summarized in Table 3.

Table 3. Seizure detection algorithm performance comparison using Dataset A

Seizure Detection Algorithms	**Data used**	**EEG Channels**	**Method**	**Sensitivity (%)**	**Specificity (False Detections h^{-1})**	**Detection Delay (Sec)**
Saab et al. [60]	652 h of EEG from 28 Patients with 126 seizures	Bipolar 24 or 32 channels	Features: Wavelet transform (WT), Amplitude, Energy, Variance. Classification: Bayes	78	0.86	9.8 (median)
Shoeb et al. [59]	844 h of EEG from 23 patients with 163 seizures	Bipolar 18 or 23 channels. (From Dataset A)	Features: windowed filter bank, Spectral energy, spatial features, temporal feature vectors. Classification: SVM	96	0.083	3 (median) 4.6 (mean)
Zandi et al. [86]	76 h of EEG from 14 patients with 63 seizures	Bipolar 15 channels	Features: windowed WPT, spectral Energy, Combined seizure index. Classification: CUSUM Thresholding	90.5	0.51	7 (median) 8.02 (mean)
Zandi et al. [4]	236 h of EEG from 26 patients with 79 seizures	Bipolar 18 channels	Features: windowed WPT, Regularity index, Energy index, Combined seizure index. Classification: CUSUM Thresholding	91	0.33	7 (median) 9 (mean)
Kuhlmann et al. [61]	525 h of EEG from 21 patients with 88 seizures	Bipolar 16 channels	Features: WT, Amplitude, Energy, Variance measures, Cross-correlation, Relative derivative Classification: Bayes	81	0.60	16.9 (median)
Hopfengartner et al. [87]	25,278 h of EEG from 159 patients with 794 seizures	CAR electrodes on standard 10-20 and 10-10 montages	Features: Moving window STFT, averaged and integrated power. Classification: Adaptive thresholding	87.3	0.22	-
Fergus et. al [111]	171 seizure and non-seizure blocks of EEG (of 60 sec duration) extracted from dataset A	Bipolar 23 channels (From Dataset A)	Features: PSD, peak frequency, median frequency, correlation dimension, PCA, LDA methods. Classification: Tested with multiple classifiers. KNN classifier shows best performance	93%	(94%)	-
Proposed Algorithm	440 h of EEG from 23 patients with 150 seizures	Bipolar 23 channels	Features: HWPT, FD, spatial and temporal feature vector arrangement Classification: RVM	96	0.1	1.34 (median) 1.89 (mean)

The proposed algorithm achieves a seizure sensitivity of 96.2%, which is comparable to Shoeb et al. [59] and is significantly higher than the other similar seizure detection algorithms presented in Table 3. The proposed algorithm further shows comparable results in median specificity. However, the seizure detection delay is significantly lower than the other seizure detection algorithms in Table 3. Note that the dataset used in [59] is used in this study; hence, the findings in that paper can be used for direct comparison with this work. The considerably lower latency can be largely attributed to the acute performance of HWPT (further analyzed under Section 3.4.6) aided by overlapped sliding window based processing of the proposed method. The sensitivity is slightly higher than that of [59] while the median false positive rate is comparable. The proposed algorithm gains a more than 50% improvement over the seizure detection latency.

3.4.5 PERFORMANCE EVALUATION USING SHORT TERM EEG

We further evaluate the generalization of the proposed technique for classifying short term patient non-specific intracranial and scalp EEGs using Dataset B. Set E in this dataset represent ictal state intracranial EEGs, while sets A-D contain seizure free scalp and intracranial EEGs. The task is to detect seizure EEG (set E) amongst the seizure free segments. Several recent studies have utilized different combinations of this dataset for performance evaluation. For example, Samiee et al. [112] formulate six different classification tasks: 1) Classification of E from a combination of E and A. 2) Classification of E from E and B. 3) Classification of E from E and C. 4) Classification of E from E and D. 5) Classification of E from E, A and C. 6) Classification of E from A, B, C and D. We evaluate the proposed method following the task 6 in [112] as it is the most challenging classification problem involving this dataset.

Table 4. Seizure detection algorithm performance comparison using Dataset B (short term EEG)

	Ocak et al. [113]	Samiee et al. [112]	Guo et al. [114]	Tzallas et al. [115]	Song et al. [116]	Proposed method
Accuracy	96.2%	98.1%	98.3%	97.7%	97.5%	99.8%
Sensitivity	98%	99.2%	99%	99%	99%	99%
Specificity	94.3%	93.8%	95.5%	94.2%	96%	100%

The proposed pipeline for classification is slightly altered to process short-term EEGs. First, all EEG segments are re-sampled to 256 Hz. Then, each segment is epoched using the same method explained in Section 3.3.2. This results in ~18 epochs per EEG segment. All EEGs in Dataset B are filtered using a band-pass filter between 3 Hz to 80 Hz as intracranial recordings may contain vital seizure components in higher frequencies. The upper cutoff of 80 Hz is chosen to adhere to the Nyquist criterion. Accordingly, HWPT decomposition is also performed within the same bandwidth. The fractal feature extraction procedure is unchanged. Though the pipeline is designed to perform epoch based classification, entire EEG segments in Dataset B are intended to be classified for seizure availability. Therefore, we use a two-step process in which epoch based classification on each EEG segment is performed first, and then the whole segment is classified based on its majority epoch classifications. The analysis is performed using the 5 fold cross-validation procedure. Note that the short-term nature of Dataset B prohibits a detection latency analysis. Table 4 shows the performance of the proposed method and presents a comparison with other seizure detection techniques in the literature that utilize the same dataset.

The proposed algorithm achieves 99.8% classification accuracy with a single false negative, and consequently offers 100% specificity and 99% sensitivity. This performance is notably higher than that of other seizure detection techniques in the literature for the same dataset. Similar to the scalp EEG, subsequent IEEG analysis with a narrower frequency band (3 Hz to 32 Hz) also yields the same seizure classification accuracy of 99.8%. This shows the robustness of the proposed algorithm in both scalp and intracranial seizure detection and classification. Note the artifact free nature of dataset B may also have contributed to improved performance in IEEG classification. Overall, the results with dataset B suggest that the proposed technique offers effective seizure detection performance using non-patient specific and artifact-free EEG recordings. Additionally, with sets D and E being IEEG, we show that the proposed method can be applied across both recording paradigms seamlessly.

Furthermore, we utilize Dataset B to investigate the benefits of using RVM over a more common classifier such as SVM for classification. The RVM in the proposed pipeline is replaced with an SVM classifier, while the rest is unchanged. The pipeline with SVM classifier achieves the same 99.8% accuracy as the RVM. However, RVM offers important advantage in training efficacy. The SVM requires ~1047 s per training fold, while the RVM needs ~552.7 s for the same. This advantage in training efficacy enables the use of RVM for large and high dimensional data classification tasks that are common in seizure detection with long term EEG.

3.4.6 COMPARATIVE EVALUATION OF FEATURE IMPORTANCE

The performance of the proposed seizure detection method relies on two main features of EEG such as fractal dimension estimates and HWPT spectral energy estimates. To investigate the influence of each of these features on the combined seizure detection method, we perform

seizure detection using each feature separately and compare the results with the combined method. The feature importance evaluation on the long term EEG is obtained using a subset of data selected from Dataset A containing 26 seizure events recorded from 4 subjects such as ‘chb1’, ‘chb3’, ‘chb5’, and ‘chb7’. These subjects are chosen for the availability of adequate and a similar number of seizure events and includes a challenging event that is misclassified in the main analysis. The results are obtained using the leave one out cross validation method. Analysis is also conducted on the short-term EEG with all of Dataset B, using the processing method described in Section 3.4.5. Table 5 shows the seizure detection performance of each of these features on Dataset A and Dataset B, respectively.

Table 5. Evaluation of feature importance

	Dataset A			Dataset B
	Sensitivity (%)	Latency (sec)	False Positive rate (h^{-1})	Accuracy (%)
FD Estimates	97	5.86	0.243	99.6
HWPT spectral estimates	97	0.39	0.254	99.2
Combined Features	97	0.514	0.1	99.8

The results for Dataset A show that the seizure detection sensitivity is 97% for FD, HWPT and the combined FD and HWPT features respectively. This illustrates that both FD and HWPT can successfully characterize the oscillatory patterns of seizure EEG. However, the

difference in performance can be seen in terms of latency and false positive rate. The FD feature based detection results in a detection delay of 5.86 s while HWPT based detection delay is just 0.39 s. This shows that HWPT characterizes the onset of a seizure event much better than FD for this dataset. The combined features result in a slightly higher latency of 0.514 s than that of HWPT. However, this minor drawback for combined FD and HWPT features is offset by the notable reduction of the false positive rate to 0.1 h^{-1}. The false positive rates of FD and HWPT based detection are comparable to each other at 0.243 h^{-1} and 0.254 h^{-1} respectively. However, combining FD and HWPT features for detection results in an excellent false positive rate percentage reduction of 59%.

On the other hand, the seizure detection accuracy for Dataset B offers 99.6%, 99.2%, and 99.8% for FD, HWPT, and combined features respectively. This is because Dataset B consists of artifact-free EEG segments as opposed to the raw EEGs in Dataset A. The results illustrate that EEG artifacts due to interference may cause many of the false detections in Dataset A. Seizure latency is not applicable for the short-term EEG Dataset B with the analysis described in section 3.4.5.

Though widely used in seizure detection methods, generic wavelet decomposition techniques are computationally complex and time-consuming, especially if the number of EEG channels is high. This may complicate the real-time seizure detection ability of many algorithms. In contrast, the HWPT method exploits the fast Fourier transform for signal decomposition which is significantly less cumbersome. Similarly, the fractal estimation utilized in this study is also computationally simple and less involved. The seizure detection algorithm designed in this study is fully implemented in MATLAB, and the time taken for HWPT of a 2 s signal epoch is ~0.0003 s on average. Fractal estimation requires an average time of 0.004 s. As the time taken

for the classification step is negligible (1.5×10^{-5} s), the total run-time of the algorithm on a 23 channel EEG segment of 6 s (the moving window used in the proposed method) is 0.4286 s. In comparison with HWPT, classic wavelet packet decomposition on a 2 s signal epoch on the same hardware and software takes ~0.024 s on average. This shows a substantial run-time advantage in using HWPT over classic WPT methods; this is an advantage of the proposed method over comparable studies that use classic WPT in the seizure detection task. Therefore, our proposed technique is highly sensitive and useful for real-time implementation.

3.5 SUMMARY

This chapter proposes a novel algorithm for epileptic seizure detection with scalp EEGs that utilize a wavelet decomposition method, known as HWPT, and FD estimation. The procedure of feature extraction and the formation of feature vectors are designed such that spectral, fractal, spatial, and temporal information of seizure EEGs are captured in the feature vectors. The proposed algorithm has a sensitivity of 96% with a median false positive rate of 0.1 $h^{-1}$and an average detection delay of 1.89 s for the long term EEG Dataset A. Analysis of short-term scalp and intracranial EEGs in Dataset B yields a 99.8% seizure detection accuracy. These results suggest that the seizure detection performance for the proposed method is better than other methods published in the literature. The results also demonstrate that the proposed method is effective with both short- and long-term EEG signal analysis recorded with either scalp or intracranial modes respectively. Further analysis of FD and HWPT suggest that the combination of these features offers distinct improvements in seizure detection latency and false positive rates.

The results also show that the proposed algorithm is computationally efficient. The HWPT is performed using fast Fourier transform and is much faster than other wavelet transform methods. The FD estimation also involves low computational complexity compared with other methods. Consequently, the run-time of the proposed algorithm of approximately 0.43 s to process a 6 s epoch of 23 channel scalp EEG highlights the potential use for real-time applications. These findings are published in [17].

The proposed ML pipeline shows improved performance and speed compared to state-of-the-art models. However, the pipeline still suffers from the problems inherent to classical ML methods. For instance, the HWPT based feature extraction is hand engineered upon observation that the seizure EEG may contain oscillatory patterns through time. Similarly, FD features are hand selected to capture self-similarity characteristics observed in EEG. However, as evidenced in section 3.4.1, sudden deviations in these observations may yield erroneous results. These limitations of ML motivate us to utilize deep recurrent learning models that are capable of feature learning for time series processing.

Moreover, we implement HWPT and FD feature extraction methods to capture time-frequency, and self-similar characteristics. However, observe that the features are computed per each EEG channel separately and aggregated for final classification. This setup is quite rigid and is unable to account for the EEG channel locality in processing multi-channel data. Additionally, increment of the number of channels (increasing the dimensionality of input data) results in rather large increase in computational cost (observe run time per channel versus all channels together). This motivates us to investigate efficient spatially distributed neural network architectures for multi-source time series data processing applications.

Consequently, in the next chapter, we change our focus to investigate a unique neural network architecture known as cellular simultaneous neural network. CSRN architecture is shown to be quite adept at processing topological relevance in spatially distributed data in an efficient manner. However, the complexity of the structure makes training the CSRN quite challenging. Therefore, in the next chapter we address the issues faced in training CSRN for complex topological image processing tasks.

CHAPTER 4

COMPARISON OF CONSTRAINED AND UNCONSTRAINED LEARNING FOR CELLULAR SIMULTANEOUS RECURRENT NETWORK IN IMAGE PROCESSING

4.1 CHAPTER OVERVIEW

This chapter investigates the efficacy of training algorithms for the neural network architecture called Cellular Simultaneous Recurrent Neural network (CSRN). This network is considered as the platform to develop a novel architecture for large-scale multisource time series data processing. The CSRN architecture is shown to be quite adept at learning certain complex image processing tasks such as affine image transformations. The unique cellular structural composition of CSRN allows approximation of the complex topological mapping involved in the image transformations. However, training CSRN for topological image processing is a challenging task because of the complexity of the network and the sheer size of the image data. Several representative training algorithms such as Back-propagation Through Time (BPTT), Extended Kalman Filtering (EKF), and Particle Swarm Optimization (PSO) have been used to train CSRN for different applications. However, the literature does not show a systematic approach for choosing an appropriate learning algorithm for training a complex universal approximator such as the CSRN in generalized image processing applications. Consequently, this work obtains generalization of the network architecture for image processing and simultaneously performs a systematic comparison of the CSRN network to solve complex topological image mapping problems. Specifically, a comprehensive comparison among three state-of-the-art learning techniques such as EKF, Unscented Kalman Filter (UKF) and PSO in training the CSRN are investigated. It is shown that introduction of an unconstrained Jacobian

free UKF learning algorithm alleviates the high computational cost otherwise associated with calculating Jacobian for EKF. Our result also shows that Jacobian-free training algorithms such as UKF and PSO for unconstrained optimizers perform better than a Jacobian-based algorithm such as EKF in image data processing.

4.2 LITERATURE REVIEW

The Artificial Neural Networks (ANNs) can be considered as universal approximators such that the ANNs can approximate a given function distribution. However, generic feed-forward MLPs perform poorly in approximating highly non-linear dynamical systems with multiple feedbacks [65, 117]. On the other hand, Recurrent neural networks (RNNs) and their variants such as simultaneous recurrent networks (SRNs) incorporate the flexibly of performing the tasks of MLPs as well as being particularly proficient in approximating feedback based dynamic systems [74, 118, 119]. Moreover, a composite cellular version of SRN known as the CSRN is shown to be adept at performing complex tasks such as long-term optimization, reinforced learning and topological mapping efficiently [120].

4.2.1 TYPICAL TRAINING METHODS USED FOR CSRN

An efficient learning algorithm is necessary to train any universal approximator for implementing a given function. Back-propagation based gradient decent is one of the first algorithms introduced to train feed-forward neural networks and MLPs. The introduction of RNNs with inherent feedback connections rendered the generic back-propagation algorithm unusable. For training RNNs, a modified version of the back-propagation algorithm known as Back-propagation Through Time (BPTT) is introduced. BPTT was also utilized to train more

complex universal approximators such as SRN and CSRN [78]. However, BPTT proved to be inefficient for complex tasks such as approximating Bellman's cost function [78]. To improve the efficiency of training complex universal approximators, Extended Kalman Filter (EKF) was introduced as an alternative to BPTT to train CSRN [120]. The EKF training offered several orders of magnitude improvement in time over that of BPTT for solving a maze traversal problem [81]. The EKF is an extension of the original Kalman filter [79] that is applicable for state predictions in non-linear systems [80] by computing the first order linear approximation of the non-linear derivative functions (known as Jacobian). However, for complex universal approximators such as CSRN, the Jacobian calculation can be computationally expensive, especially for large scale image data processing applications.

Unscented Kalman filter (UKF) alleviates the drawbacks of EKF and offers more accurate state estimation of non-linear systems. UKF utilizes the unscented transform, a method of analyzing a random variable that traverses a non-linear system, for its state estimation process [80]. Even though several studies have proposed the use of UKF to train ANNs and RNNs [80, 121], none have been reported to train a cellular RNN such as the CSRN. The UKF is especially desirable for complex universal approximators as it only requires the forward propagation function of the network during the training process and therefore avoids Jacobian computation. Likewise, another Jacobian free learning technique known as Particle Swarm Optimization (PSO) [122] can be another candidate for training universal approximators such as a CSRN [123].

4.2.2 FUNCTION APPROXIMATION AND OPTIMIZATION CAPABILITY OF CSRN

One of the earliest attempts to study the versatility of CSRN was reported by Ilin et al. [120] wherein the authors showed two representative tasks such as maze traversal and

connectedness problem. Demonstrating the versatility and generalizability of CSRN, a few of our prior studies [46, 47] showed that CSRN can also handle complex topological mapping tasks commonly observed in image processing. In addition, complex function approximation tasks like image transforms can be considered as challenging test cases for a universal approximator. Therefore, building on our preliminary work [46], this paper proposes three representative image analysis tasks such as image binarization, spatial filtering and Affine transformations to demonstrate and compare the function approximation and optimization capability of CSRN. As such, we obtain a generalized CSRN structure that is suitable for a multitude of spatial domain image processing tasks. Furthermore, this work introduces a CSRN based spatial domain image processing pipeline. The pipeline systematically compares the performance of three state-of-the-art learning algorithms: one with Jacobian (e.g., EKF) and the other two without Jacobian (UKF and PSO) computation for large scale image data processing. The results show that all three learning techniques can train CSRN to solve the image processing tasks with varying efficacy.

4.3 PROPOSED TRAINING ALGORITHMS FOR CSRN BASED IMAGE PROCESSING

This section outlines the training algorithms utilized in the generalized CSRN model for image processing. Each algorithm is implemented and evaluated by training the CSRN for a variety of image processing tasks.

4.3.1 CSRN TRAINING WITH EXTENDED KALMAN FILTER (EKF)

The EKF models the training of an ANN as a parameter estimation process. This state-space model of the non-linear ANN system can be described using the following state transition and observation equations,

$$W_{k+1} = W_k + \gamma_k, \tag{26}$$

and,

$$Y_{k+1} = F(W_k, u_k) + \eta_k. \tag{27}$$

where W_k and γ_k are the system state (neural network weights in this case) and process noise at iteration k respectively. In (27), F is the forward propagation function of the neural network, while u_k is the neural network input, and η_k is the measurement noise (typically zero mean white noise) at iteration k. Y_{k+1} is the observation at $k + 1$. The EKF training algorithm is shown in Algorithm 1. Further details on EKF based training of CSRN can be found in [120].

Algorithm 1 CSRN Training Procedure with EKF

1. Initialization: Initialize CSRN weights, W and compute initial weight mean, μ_k and covariance, P_k. Initialize process and measurement covariance, R_k and Q_k respectively.
2. For each training step k: Compute the output of the network, Y_k and the error between the target and network output (Innovation) a_k.
3. Utilize BPTT to compute the Jacobian, C_k
4. Compute Kalman Gain, K_k
5. Compute the new weight mean estimate, μ_{k+1} and covariance estimate, P_{k+1}.

4.3.2 CSRN TRAINING WITH UNSCENTED KALMAN FILTER (UKF)

Like EKF, the problem of training an ANN using UKF is also posed as a parameter estimation problem. The system state transition and measurement equations are respectively given as follows,

$$w_{k+1} = w_k + \varepsilon_k, \tag{28}$$

and,

$$y_k = F(w_k, u_k) + \delta_k, \tag{29}$$

where, ε_k in (28) and δ_k in (29) correspond to the zero mean additive white process and measurement noise with covariance Q_k and R_k respectively. The system state (Network weights) is denoted as w_k at iteration k and the non-linear function (Network forward propagation) is denoted as F. The neural network input at iteration k is u_k. The UKF utilizes system state statistics of the previous iteration, network input and target of the current time step to estimate the current system state. The UKF based training algorithm is given in Algorithm 2. Note this process eliminates the need for a Jacobian calculation. A more detailed description specifically on using UKF in training CSRN can be found in [47].

Algorithm 2 CSRN Training Procedure with UKF

1. Initialization: Initialize CSRN weights, W and compute initial weight mean, μ_k^w and covariance, P_k^w. Initialize process and measurement covariance, R_k and Q_k respectively.

2. For each training step k: Perform Bayesian prediction step to obtain, $\bar{\mu}_{k+1}^w$ and $\bar{P}_{k+1}^w$

3. Compute the sigma points, $\boldsymbol{Z}_{k+1}$

4. Pass each sigma point from $\boldsymbol{Z}_{k+1}$ through CSRN forward-propagation function to obtain measurement updates, $\boldsymbol{Y}_{k+1}$

5. Compute updated estimate statistics, μ_{k+1}^y, P_{k+1}^y and $P_{k+1}^{w,y}$ using transformed sigma points $\boldsymbol{Y}_{k+1}$

6. Compute Kalman Gain, K_{k+1}

7. Compute the new weight mean estimate, μ_{k+1}^w and covariance estimate, P_{k+1}^w.

4.3.3 CSRN TRAINING WITH PARTICLE SWARM OPTIMIZATION (PSO)

Training a neural network with *W* number of weights can be considered as a search for an optimal location in a *W* dimensional weight space, with regards to the Network cost function output, called the fitness value. The fitness value is usually the sum-squared error or the mean squared error between the known target and actual Network outputs of a training set. The PSO algorithm generates a swarm of 'particles' that traverse through this weight space under swarm behavioral performance criteria in search of the optimal location [124]. All the particle locations are evaluated with regards to the function at each movement step. The PSO based training algorithm is given as Algorithm 3.

Algorithm 3 CSRN Training Procedure with PSO

1. Initialization: Initialize a swarm of particles with random positions, $X_i(0)$ and velocities, $V_i(0)$ in the weight (W) space. Initialize particles best fitness value, p_i and location, X_i^p, swarms best p_g and X_g.

2. For each particle, evaluate the fitness function (i.e. the error between target and network output), f_i.

3. For each training step k: Compare each particles fitness value, f_i with its best fitness value, p_i. If $f_i < p_i$, set $p_i = f_i$ and $X_i^p = X_i(k)$

4. Compare each particles p_i with swarms best p_g. If any $p_i < p_g$, set $p_g = p_i$, and $X_g = X_i(k)$.

5. Compute new velocity:

$$V_i(k+1) = w.V_i(k) + c_1.rand_1\left(X_i^p(k) - X_i(k)\right) + c_2.rand_2\left(X_g(k) - X_i(k)\right),$$

and new position:

$$X_i(k+1) = X_i(k) + V_i(k+1),$$

where w, c_1, c_2 are momentum, cognitive acceleration, and social acceleration terms and functions $rand_1$ and $rand_2$ are uniformly distributed random vectors

4.4 GENERALIZED SPATIAL DOMAIN IMAGE PROCESSING WITH CSRN

This section introduces the proposed generalized image processing pipeline based on the CSRN and discusses three state-of-the-art learning algorithms used in this study.

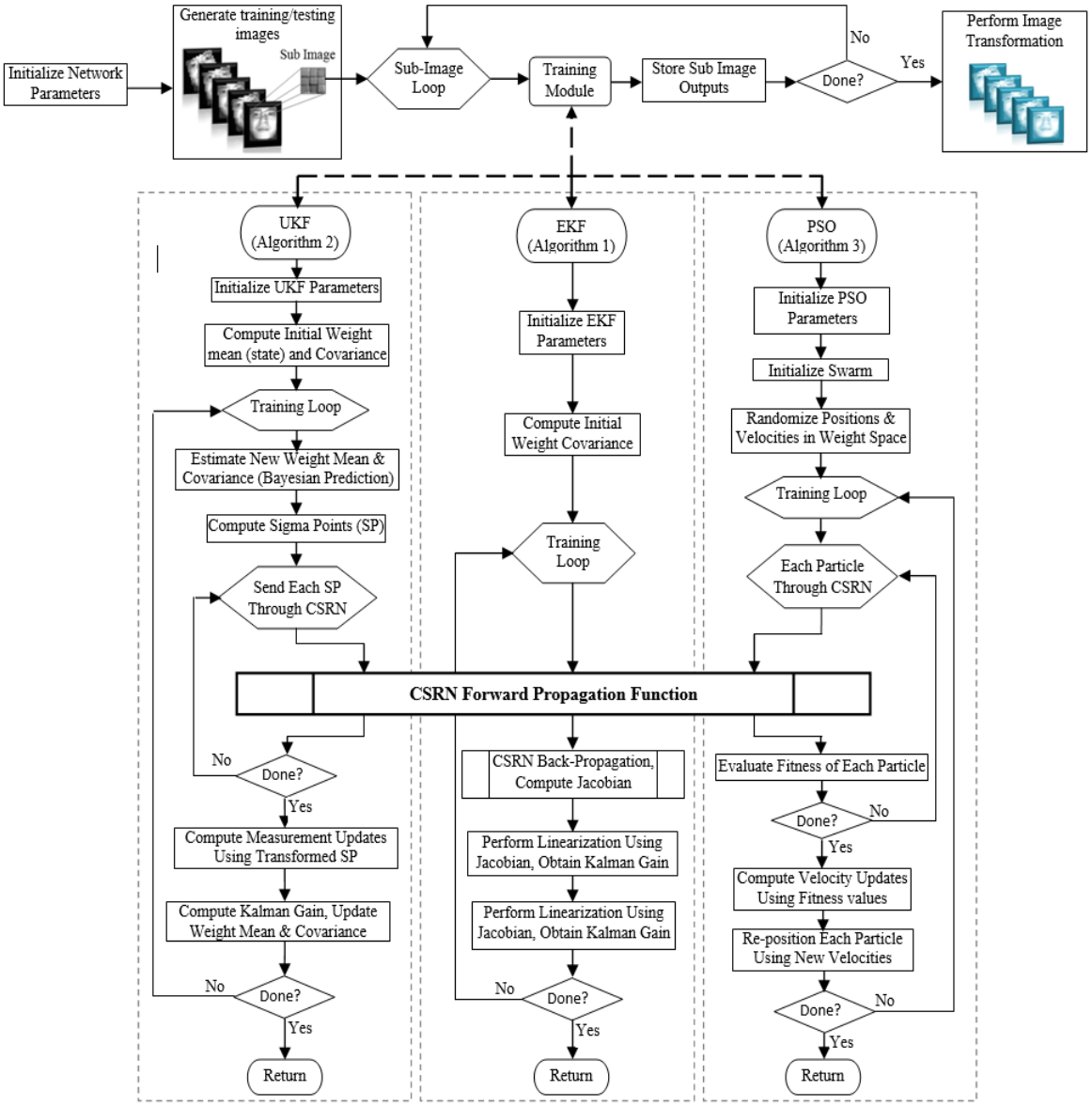

Fig. 17. Pipeline developed for comparison of training algorithms for generalized image processing

The flow diagram in Fig. 17 shows the proposed pipeline for training CSRN for image processing. This flow diagram includes the three state-of-the-art learning methods such as EKF, UKF and PSO identifying the common and independent processing blocks for each. The diagram

also includes the sub-image processing paradigm introduced by Anderson et al. [46] which adds the capability to process larger and complex gray-scale images. This is a generalized platform that may allow a quick comparison of competitive training algorithms for any given complex universal approximator to perform a complex task such as large-scale image processing. This pipeline implements three types of spatial domain image processing tasks: *1)* greyscale to binary conversion, *2)* spatial filtering, *3)* affine transformations.

4.4.1 BINARY IMAGE PROCESSING

Greyscale to binary conversion, also referred to as image binarization, is a simple image processing task that transforms each pixel of a greyscale image to binary depending upon the intensity of the pixel relative to a threshold value. Greyscale to binary conversion of an $N \times M$ image I_1 is given as,

$$I_2(x,y) = \begin{cases} 0 & if \;\; I_1(x,y) < TH \\ 1 & if \;\; I_1(x,y) \geq TH \end{cases} \qquad (30)$$

$$for\; x = 1\; to\; N, y = 1\; to\; M$$

where (x, y) represents the current location of the filter mask and *TH* is the threshold value and $I_2(x, y)$ is the resulting binary image.

4.4.2 IMAGE FILTERING

Spatial filtering operates directly on the pixels of an image [125] in a neighborhood surrounding the pixel of interest, referred to as the filter window. The response of the filter at a

given pixel location is the discrete convolution of the filter mask with the underlying sub-image. This filtering operation on an $N \times M$ image I_3 can be represented as,

$$I_4(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i,j) \cdot I_3(x+i, y+i), \quad \text{for } x = 1 \text{ to } N \text{ and } y = 1 \text{ to } M \tag{31}$$

where the filter mask, $w(i,j)$ is of size $m \times n$, and $a = \frac{m-1}{2}$, and $b = \frac{n-1}{2}$. $I_4(x,y)$ is the resulting filtered image.

4.4.3 IMAGE AFFINE TRANSFORMATIONS

Image Affine transformation is a topological mapping technique, where the locations of the intensity values of an input image are mapped to new locations of an output image. The Affine transformation is given as,

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x_o \\ y_o \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{32}$$

where (x_n, y_n) and (x_o, y_o) are the spatial coordinates of the corresponding pixel locations in transformed and original images, respectively. Rotation is performed by the 2×2 rotation matrix with an angle of "θ". "t_x" and "t_y" represent the translation along x and y axis. "S_x" and "S_y" represent the scaling parameter in x and y directions. More details on image Affine transformations can be found in [126].

Equation (32) suggests that in order to perform affine transformations using CSRN, several inputs must be provided for each of its SRN. They are (i) pixel location (the "x" and "y"

locations of a pixel within the image) and (ii) transformation parameter (he amount of transformation to be performed). This may be the translation, rotation or scaling parameter. The output of the network will be the pixel's (new) location in the transformed image.

The neighbor inputs for each cell of CSRN are taken as the four closest neighbors for all the image processing operations in this study. The first active neuron is connected to the neighbors, i.e. output from A_1 (Fig. 10) is sent to the four nearest neighbors. Five active neurons are used in each SRN for this work. The output of the final active neuron A_n is multiplied by a scaling weight, Ws, to produce the cell's final output, $\hat{Y}$.

4.4.4 COST FUNCTION FORMULATION FOR GENERALIZED IMAGE PROCESSING WITH CSRN

In order to use CSRN for any image processing tasks, the CSRN learning must be capable of approximating the cost function for a specific task. Consider the cost-function for affine transformation of an image. Let I_5 represent the transformed image obtained using the affine transformation Eq. (32) and I_5^i represent the i^{th} pixel in the transformed image. Let I_T^i represent the target (true) pixel location of the transformed image. The error between each pair of target and transformed pixel location is given as follows,

$$e_i = I_T^i - I_5^i. \tag{33}$$

The total error in the transformed image is given by,

$$E_i = \frac{1}{2}\sum_{i=1}^{l} e_i^2, \tag{34}$$

where l is the number of pixels in the image. The corresponding error for an entire set of N images is given as,

$$E = \sum_{n=1}^{N} \left[\frac{1}{2} \sum_{i=1}^{l} e_i^2 \right]. \quad (35)$$

Equation (34) represents the general cost- function for affine transformation (Eq. (32)). Minimizing this cost function also minimizes the Euclidean distance between the computed pixel locations and their target locations.

4.4.5 A QUALITATIVE COMPARISON BETWEEN CONSTRAINED AND UNCONSTRAINED TRAINING ALGORITHMS

Choosing an appropriate learning algorithm for a complex universal approximator such as CSRN can be challenging for efficient training. Table 6 summarizes a qualitative comparison between these three state-of-the-art learning methods. The UKF and PSO algorithms do not involve back-propagation of the error gradient and, hence, computationally expensive Jacobian computation for a large and complex network. This enables UKF and PSO to be applied to any neural network structure easily. For EKF and other Jacobian based learning algorithms, a small change in the structure of an ANN requires deriving the back-propagation equations from the beginning, which can be computationally expensive. This computational complexity and the rapid expansion of the Jacobian matrix with network size constrain the usage of EKF with the CSRN.

Table 6. Comparison of constrained and unconstrained optimizers for training universal approximators

Comparison criteria	Training Algorithm		
	EKF	UKF	PSO
Optimization method	Statistical	Statistical	Biologically inspired
Jacobian computation	Yes	No	No
Forward-propagation	Yes (Once)	Yes (Multiple)	Yes (Multiple)
Back-propagation	Yes (Once)	No	No
Adaptability for different ANNs	Limited	High	High
Computational complexity	$O(n^2)$ [80]	$O(n^2)$ [80, 127]	$O(nk)$ [128]

On the other hand, both UKF and PSO have their own drawbacks. A potential drawback of using UKF for ANN training lies in the sigma point sampling and the measurement update calculation. If the number of trainable weights is n, then UKF draws $2n + 1$ sample points. Each sample point travels through the neural network forward propagation function, one at a time. Therefore, training a neural network with a large amount of weights with UKF may not be efficient. The disadvantage of PSO is that it may not always guarantee a global solution and the goodness of solution depends on the number of particles involved. Conversely, increasing the number of particles also increases the training duration of CSRN. The computational complexity

of EKF and UKF in parameter optimization is similar at $O(n^2)$ where n is the number of parameters (the number of weights in training NN). The computational complexity of PSO $(O(nk))$ is highly dependent on the number of particles k used.

4.5 RESULTS AND DISCUSSION

This section discusses results on the efficacy of constrained (e.g., EKF) versus unconstrained (e.g., UKF and PSO) learning for the CSRN in spatial domain image processing. These learning algorithms are evaluated on three image processing tasks such as, image binarization, filtering, and Affine transformation. These three image processing tasks represent increasingly complex pixel-wise operations. The following subsections provide the efficacy of the three representative learning algorithms for each image processing task.

YaleB face dataset [129] is used for all the experiments conducted in this study. Table 7 lists the evaluation metrics related to image comparison, function approximation, and speed. All simulations are performed on a workstation with an Intel i7-2630QM 4-core 2.00 GHz CPU and 8.0 GB of RAM.

Table 7 performance evaluation metric for image processing

Metric	Description
J_{SSE}	Sum Squared Error between the target function and network output. This evaluates how well the network learns. J_{SSE} varies with image size.
J_{MSE}	Mean Squared Error between the target function and network output. Computed by normalizing J_{SSE} by image size and the number of training images.
J_{ACC}	Percentage of cell outputs which exactly match known target function values. It evaluates the accuracy of the function.
IM_{ACC}	Percentage of matching pixels between the target image and the output image.
IM_{CR}	Correlation ratio between the target image and the output image. It evaluates the similarity between target and output image. IM_{CR} closed to 1 indicates a higher similarity between the two images.
T_{TR}	Training time (secs) – time required to train the CSRN.

4.5.1 BINARY IMAGE PROCESSING

Greyscale to binary conversion is a spatial domain pixel-wise operation that does not affect the surrounding pixels. This operation does not involve information sharing among neighboring pixels; therefore' the connections with the neighboring cells in CSRN are not required. In this experiment, the CSRN is trained using 11 facial images with a threshold value $\theta = 0.4$ chosen as an example. The target images for training CSRN are generated using

MATLAB©'s greyscale to binary conversion function. The CSRN is tested using a different set of 11 test images from the same dataset. An example test image is considered as the primary test case and is shown in Fig. 18.

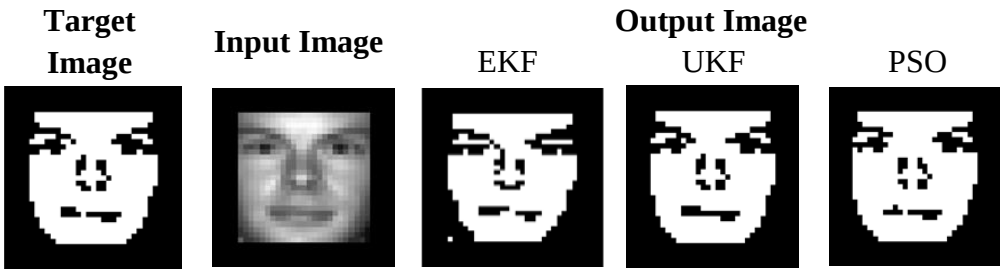

Fig. 18. Greyscale to binary conversion results (Primary Test Cases).

Fig. 18 illustrates that the output images obtained by training the CSRN with three different learning algorithms are very similar to the target image. This is confirmed by the proposed performance metrics IM_{ACC} = 98.3% and IM_{CR} = 96.4%; IM_{ACC} = 99.10% and IM_{CR} = 98.0%; IM_{ACC} = 99.35% and IM_{CR} = 99.0% using EKF, UKF and PSO, respectively. The results show that all three learning methods can train CSRN to successfully perform greyscale to binary transformation with comparable accuracies.

4.5.2 IMAGE FILTERING

In this experiment, we implement a low pass filter (LPF) using CSRN in order to investigate the performance of the learning methods in spatial filtering. Like the greyscale to binary conversion, spatial filtering is performed at the individual pixel level. Therefore, a simpler

CSRN structure like the one used for greyscale to binary conversion is used. The CSRN is trained using the same 11 facial images as used in the greyscale to binary conversion experiment with filter coefficient, $\theta = \frac{1}{9}$. The target images are generated using MATLAB©'s standard image filtering function. The CSRN is tested using the same 11 facial images used as the test samples in the greyscale to binary conversion. An example outcome on a test image sample is shown in Fig. 19.

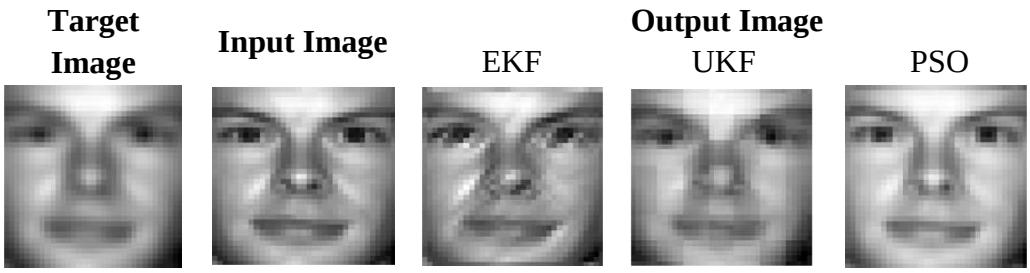

Fig. 19. Results for LPF implementation.

Fig. 19 shows that the output images produced by the CSRN trained with three different learning algorithms are very similar to the target image. This is quantitatively confirmed by the performance metric such as IM_{CR} = 93.0%; IM_{CR} = 96.6%; and IM_{CR} = 97.0% using EKF, UKF and PSO, respectively. Among the three learning algorithms, UKF and PSO show better performance in the low pass filtering application when compared to EKF.

4.5.3 IMAGE AFFINE TRANSFORMATIONS

This section evaluates the efficacy of both learning techniques for CSRN in a more complex image processing task such as affine transformation of facial images in greyscale. The

computation complexity of CSRN increases with image size. Specifically, with the current state-of-the-art EKF-based CSRN training, the computation of Jacobian becomes increasingly complex with large images [83]. In order to lessen the computation complexity of the Jacobian in EKF, each input image is segmented into 5×5 non-overlapping sub-images, and a separate CSRN is assigned for each sub-image [46]. Following the steps in Fig. 17, the CSRN is first trained, and the transformation is obtained by first segmenting the test image into 5×5 sub-images and then sending each sub-image through the corresponding CSRN.

The transformed sub-images are recombined to produce the final output. In order to make a fair comparison between the training algorithms, UKF and PSO are also utilized in a similar manner to train the CSRN. A 35×35 greyscale facial image is considered for performing the Affine transform. The image transformation results for the primary test cases (PTC: $t_x = 10$, $\theta = 16°,\ S_x = S_y = 0.73$) are shown in Fig. 20 and Table 8.

The Affine translation of greyscale images is performed through $0\ to\ 10$ pixels using 11 training images. The examples of translation on the test image are shown in Fig. 20. Table 8 shows that CSRN trained with EKF, UKF, and PSO achieves $IM_{ACC} = 100\%, 96.41\%$ and 100% for translation, respectively. The approximation accuracy (J_{ACC}) for the translation function is 100% by both EKF and PSO, while UKF shows a slightly lower accuracy of 92.24%.

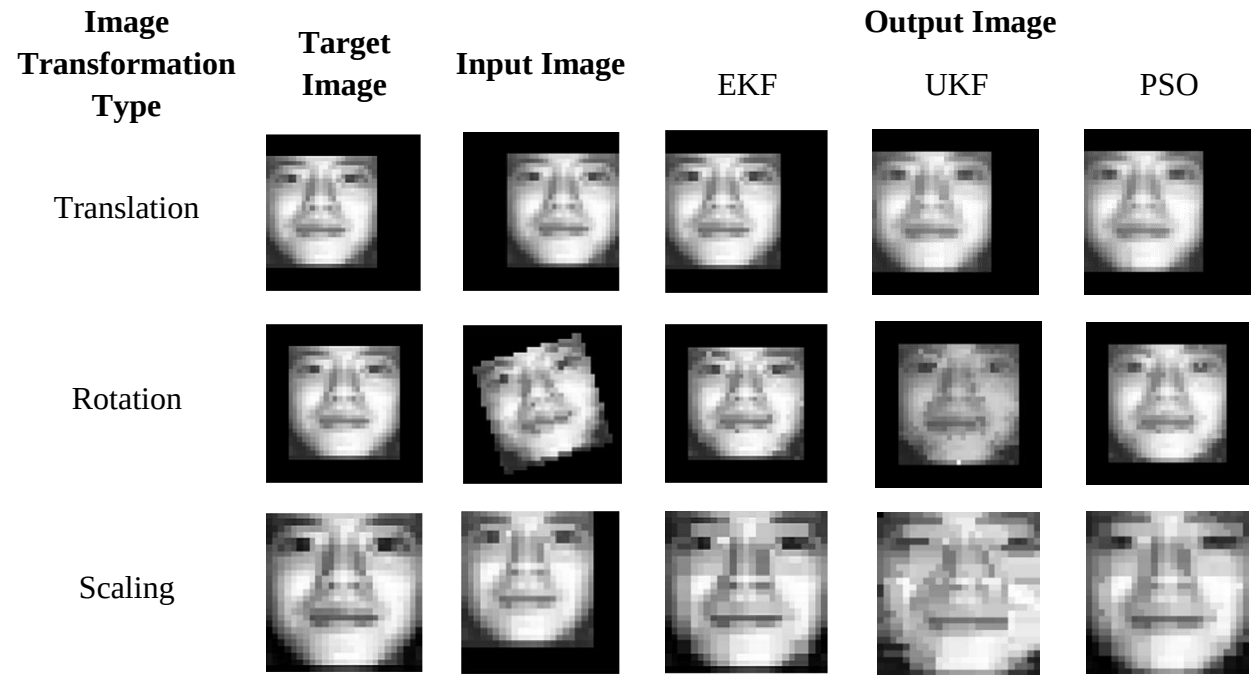

Fig. 20. Greyscale image affine transformation results (Primary Test Cases).

Affine rotation on greyscale images is performed in the range $\theta = 0° \; to \; 20°$ in 2° steps using 11 training images. For rotation, the CSRNs trained with EKF, UKF, and PSO yield $IM_{ACC} = 94.9\%,\ 89.22\%$ and 96.64%, respectively. The CSRNs trained with EKF, UKF and PSO also yield $J_{ACC} = 96.8\%,\ 85.06\%$, and 94.40%, respectively in approximating the rotation function.

The experiment for the scaling of greyscale images is limited to training images of scale factor $S_x = S_y = 0.5$ to 1.0. Seven training images with scale factors $S_x = S_y = 0.52,\ 0.6,\ 0.68,\ 0.76,\ 0.84,\ 0.92$ and 1 have been used. For Affine scaling, the CSRNs trained with EKF, UKF, and PSO achieve $IM_{ACC} = 68.3\%, 36.32\%$ and 66.24%, respectively. The accuracies in approximating the scaling function have been found as $J_{ACC} = 96.8\%, 51.52\%$ and 91.36% for CSRNs trained with EKF, UKF and PSO, respectively.

Table 8 shows that the EKF and PSO train CSRN with a higher function approximation accuracy (J_{ACC}) than that of UKF. Note that the above experiments utilize the sub-image processing scheme which is developed specifically for EKF-based learning of large images. As mentioned earlier, Jacobian computing becomes difficult with large image sizes which in turn drastically degrades the EKF performance. In contrast, the performance of UKF and PSO is largely unaffected by the size of the input image.

Table 8. Summary of greyscale image transformation with CSRN

Metric (Units)	Greyscale Transformation								
	Translation (PTC: $\theta' = 10pix$)			Rotation (PTC: $\theta' = 16^0$)			Scaling (PTC: $\theta' = 0.73$)		
	EKF	UKF	PSO	EKF	UKF	PSO	EKF	UKF	PSO
J_{ACC} (%)	100	92.24	100	96.8	85.06	94.40	96.8	51.52	91.36
J_{MSE}	0.0	0.01	0.0	0.003	0.024	0.01	0.005	0.08	0.01
IM_{ACC} (%)	100	96.41	100	94.9	89.22	96.64	68.3	36.32	66.24
IM_{CR}	1.0	1.0	1.0	0.9979	0.99	1	0.9734	0.91	0.98
T_{TR} -total (sec)	835	1277.86	560.79	887	1337.55	791.18	1060	816.46	582.13

We also compare the training time for all three learning algorithms. The training time is obtained to study the performance of UKF and PSO when trained with larger *full size* grey scale images. The rotation is chosen as an example Affine transform considering its complexity. The results are shown in Table 9 and Fig. 21.

Table 9. Training time vs. Image size for EKF, UKF, and PSO based training of CSRN

Image Size	Training Time (Sec.)		
	EKF	UKF	PSO
25x25	232.58	410.05	1050.19
35x35	1030.01	845.375	2564.00
45x45	3950.74	1581.575	13986.81
55x55	10670.78	3011.69	29299.93

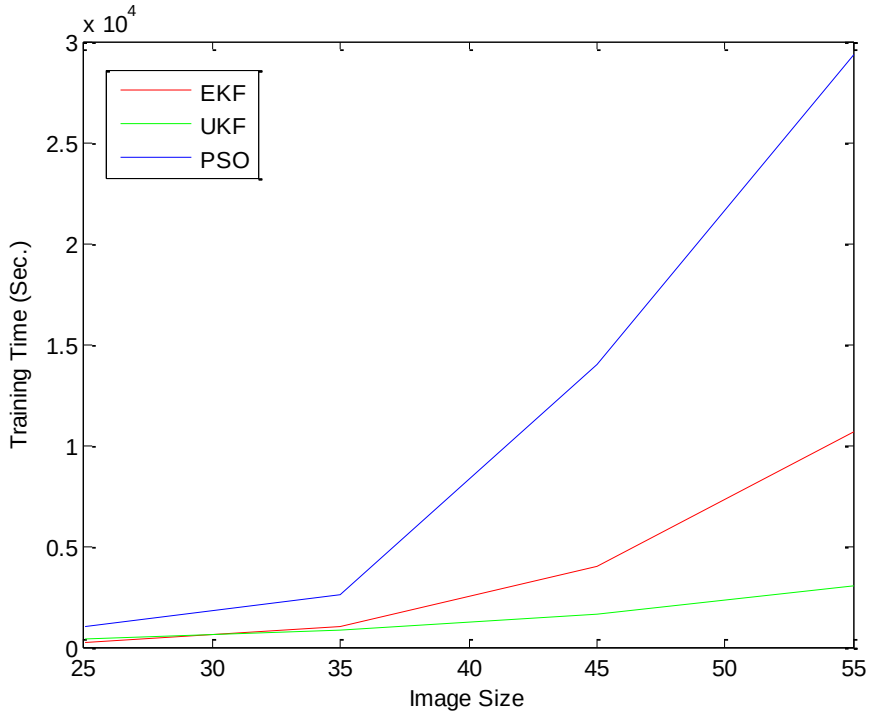

Fig. 21. Training time vs. Image size using EKF, UKF, and PSO for image Affine transform

All three training algorithms are used with the same set of greyscale facial images and trained for 50 epochs. Table 9 shows that EKF training time is much lower than that of both UKF and PSO for smaller sized images (25x25). As the image size increases, the training time for UKF becomes much lower than that of EKF and PSO. Compared to EKF and PSO, the training time for UKF grows at a slower rate with respect to the input image size as shown in Fig. 21. As an example, for an image size of 55x55, the training time for EKF is about 3.5 times than that of UKF whereas, the training time for PSO is 10 times that of UKF. This shows that UKF can handle larger size input samples more efficiently than EKF and PSO for a universal approximator such as CSRN. For the case of UKF, the increase in the input size only increases the number of cells. Since the number of weights does not increase (because of the weight sharing property of CSRN), it does not alter the complexity of the optimization process of UKF.

Note the training time for the UKF algorithm depends on the number of weights associated with the network. The reason behind higher training time for EKF is explained above. For PSO, the value of the cost function grows with the size of the input which in turn increases the time for each particle to calculate its fitness value. As a result, the PSO algorithm takes more time to reach the global convergence.

The results suggest that EKF offers better performance when the input image size is small. On the other hand, UKF shows better performance for larger images in terms of training time. PSO shows better convergence for both small and large input images. Our results also show that on average the training time of UKF is about 3 times faster than EKF and 8 times faster than that of PSO for larger image samples. In conclusion, for large scale applications, a constrained learning method such as EKF may not be a good choice for training complex

universal approximators. Unconstrained (e.g., Jacobian free) learning techniques such as UKF and PSO may be better choices for a universal approximator in large scale complex applications.

4.6 SUMMARY

This chapter first obtains a generalized structure of CSRN with variable external and neighbor connections for spatial domain image processing applications. We then compare three representative training algorithms for generalized image processing. A sub-image processing scheme is developed for the ease of Jacobian computation in an EKF training algorithm for large-scale images. In order to make a fair comparison, the sub-image processing is also used for UKF and PSO training algorithms respectively. In general, results show that all three learning techniques are capable of training CSRN for the spatial domain image processing tasks. However, EKF and PSO show better performance than UKF in terms of image accuracy, function accuracy, and training time for a smaller sub-image. Further, UKF outperforms EKF and PSO for larger images, especially in terms of the training time. Therefore, while EKF and PSO are desirable for low dimensional input applications, UKF is optimal for applications with larger input patterns. Moreover, UKF and PSO present clear advantages over EKF due to the added versatility from Jacobian free learning for CSRN. These findings and further experiments conducted on CSRN are published in [44, 48-50].

This chapter illustrates the experiments conducted on the CSRN and its training in highly complex topological image processing tasks. The results demonstrate the unique capability of the cellular structure in processing spatial relationships in data. This is beneficial for our task of processing large-scale time series data such as EEG that may include signal-source locality features. Additionally, the ability of CSRN to share the trainable weights among its cells is also

of importance to us as we strive to develop an efficient architecture with limited use of trainable parameters for our task. This chapter also addresses the difficulties encountered in training a complex cellular architecture by experimenting on several state-of-the-art learning algorithms. In summary, we establish the advantages of a cellular architecture in processing complex multidimensional inputs, and we address the problems in training cellular architecture. Using the cellular structure as our base, in the next chapter we develop a novel efficient recurrent neural network architecture that is capable of processing large-scale time series data with superior efficiency.

CHAPTER 5

DEEP CELLULAR RECURRENT NEURAL NETWORK FOR EFFICIENT TIME-SERIES ANALYSIS

5.1 CHAPTER OVERVIEW

Efficient processing of large-scale time series data is an intricate problem in machine learning. Conventional sensor signal processing pipelines with hand engineered feature extraction often involve huge computational costs with high dimensional data. Deep recurrent neural networks have shown promise in automated feature learning for improved time-series processing. However, generic deep recurrent models grow in scale and depth with increased complexity of the data. This is particularly challenging in the presence of high dimensional data with temporal and spatial characteristics. Consequently, this chapter proposes a novel deep cellular recurrent neural network (DCRNN) architecture to efficiently process complex multi-dimensional time series data with spatial relevance. The cellular recurrent architecture in the proposed model allows for location-aware synchronous processing of time series data from spatially distributed sensor signal sources. Extensive trainable parameter sharing due to cellularity in the proposed architecture ensures efficiency in the use of recurrent processing units with high-dimensional inputs. The proposed DCRNN architecture is evaluated using two time-series datasets: a multichannel scalp EEG dataset for seizure detection and a machine fault detection dataset obtained in-house. The results suggest that the proposed architecture achieves state-of-the-art performance while utilizing substantially fewer trainable units when compared to comparable methods in the literature.

5.2 LITERATURE REVIEW

This section provides a brief literature review of the typical time series processing methods. These methods vary in terms of the methodology and the functionality. Most feature-based methods usually follow a two step-process: 1) time series feature Extraction and selection, and 2) feature classification. Consequently, numerous feature extraction methods and classification methods have been introduced with feature-based methods. However, the overall performance of feature-based methods heavily depend on the hand engineering of appropriate feature extraction and selection for a given task.

5.2.1 FEATURE BASED TECHNIQUES IN TIME-SERIES PROCESSING

Typical pattern recognition applications oftentimes involve classification or regression of input data that is static in time. However, most real-world data obtained through a set of observations almost always exhibit changes with time. Though in some cases, the change of observations in time can be ignored, certain applications that particularly deal with changes across time require an additional temporal dimension to be incorporated in the pattern recognition process. Moreover, tasks such as monitoring multi-channel EEG for seizure detection and complex machine health monitoring may require recognition of patterns that extend in both spatial and temporal dimensions. Computational models that are specifically capable of capturing complex patterns in time and space are required to process such multi-dimensional time series data. One of the most challenging steps in constructing a ML model for complex time series analysis is an appropriate feature extraction scheme that effectively captures the patterns across time and spatial dimensions. These representative features may be a set of simple statistics of the time series data such as mean, variance, skewness, kurtosis, largest peak,

and number of zero crossings. [14]. More descriptive features such as autoregressive coefficients [15], frequency power spectral features [16], and features derived from time-frequency analysis such as wavelet transform [4, 16], wavelet packet transform [4, 17], filter banks [18], and self-similarity features [13], and further engineered versions of these may also be considered to obtain a more discriminatory representation of data. However, one of the main problems associated with feature engineering is that the efficacy of such features essentially depend on the data, and the application. Therefore, the performance of a ML pipeline depends on the hand selection of a subset of features or extraction of a set of new features based on the domain expertise. Feature learning with artificial neural networks (ANN) largely alleviates this problem by progressively learning the best possible discriminatory feature from data.

5.2.2 DEEP NEURAL NETWORKS FOR TIME SERIES PROCESSING

The availability of powerful computational tools and training methods have enabled deep neural networks to solve many difficult recognition problems in robotics [22], object recognition [23], and text recognition [24]. The typical feed-forward neural networks are predominantly used in processing data that is static in time due to its inability to process temporal relations owing to the limited forward information processing capability. Recurrent neural network (RNN) [25] or a time-delay neural network (TDNN) is a variant of ANN with the added capability of information aggregation through feedback connections. RNNs such as Elman and Jordan architectures [26-28] process time-series by reading samples sequentially in time, and the feedback connections aid in retaining valuable information through time steps. Further improvements to the feedback units in retaining memory through longer time-sequences are tasked to Long Short-term Memory (LSTM) [29] units, and Gated Recurrent Units (GRU) [30]. Large-scale deep versions of

recurrent neural networks have been successfully utilized in multiple domains [31-34]. Few works suggest the use of deep CNN and/or deep LSTM networks for processing EEG [130-132]. These typically involve an additional feature extraction step such as Fourier spectrum computation prior to the application of CNN for improved compatibility. The deep CNN is primarily used as a feature extractor while a LSTM layer is applied subsequently for temporal processing. However, current state-of-the-art deep models suffer from a major limitation. The depth, complexity, and number of trainable parameters associated with these models grow proportionally to the complexity of the input dimensionality and the given task. This problem is further exacerbated in recurrent learning models as the additional feedback links demand even more trainable parameters. Therefore, such architectures can increase prohibitively in the presence of large-scale, multi-source time-series data such as EEG.

Furthermore, the deep CNN and LSTM methods still largely ignore the spatial relevance in large scale time series data for some applications (e.g., EEG and machine fault detection) where space location information is of interest. The time series data recorded from different components in a machine health diagnosis, and fault detection system such as in [133] may hold spatial correlation based on the locality of the components. Specifically, the particle accelerator facility of Jefferson Nation Labs contains multiple cavities situated serially on cryomodules [133]. Multiple RF signals from each cavity are recorded for monitoring the operating conditions [133]. Automated detection and classification of faults in this system involves efficient processing of time series data obtained from each cavity. Interestingly, Thodoroff et al. [131] propose an image-based representation combining Fourier spectral features from individual EEG electrodes into a single image based on the 2D projection of the EEG montage. This representation maintains the spatial locality of individual EEG electrodes to exploit the spatial

relevance of seizure EEG. However, this is still processed using a large-scale multi-layer CNN and LSTM combined architecture that suffers from large computational costs for the networks.

In order to address the general lack of computationally efficient methods for processing time series data that also maintain spatial relevance, this study proposes a novel DL architecture called deep cellular recurrent neural network (DCRNN). The DCRNN is inspired by the cellular neural network architectures [42-44] that are shown useful for real time image processing [43] and approximating the dynamic programming tasks [45]. The typical cellular architecture spans the area of a 2D input such as an image, overlapping each pixel with a corresponding cell in the network [44, 45]. Such cellular architectures enable distributed processing of information while maintaining synchronized communication with the neighboring cells. The cellular architecture also promotes extensive sharing of tunable parameters by placing identical neural structures in each cell [44]. We utilize this unique cellular sub-architecture in designing our DCRNN architecture for multi-dimensional time series data processing. The cellularity of the proposed architecture allows for processing sensor signals obtained from individual sources. The grid-like placement of cells in turn enables communication with the neighboring cells, which allows learning spatial characteristics based on the locality of sensor signal sources. We also gain extensive trainable weight sharing by placing identical recurrent neural models within each cell. Moreover, the cellularity enables straightforward expansion of architecture for changes in the number of input sources, with only negligible increments to the number of trainable weights.

5.3 THE DEEP CELLULAR RECURRENT NEURAL NETWORK FOR LARGE-SCALE TIME SERIES PROCESSING

This section illustrates the details of the proposed deep cellular recurrent neural network (DCRNN) architecture and the associated input formulation and training procedure along with complexity analysis.

5.3.1 DCRNN ARCHITECTURE FOR TIME SERIES PROCESSING

The DCRNN model combines the versatility of cellular neural processing with the recurrent LSTM for flexible time series processing. The proposed architecture is shown in Fig. 22.

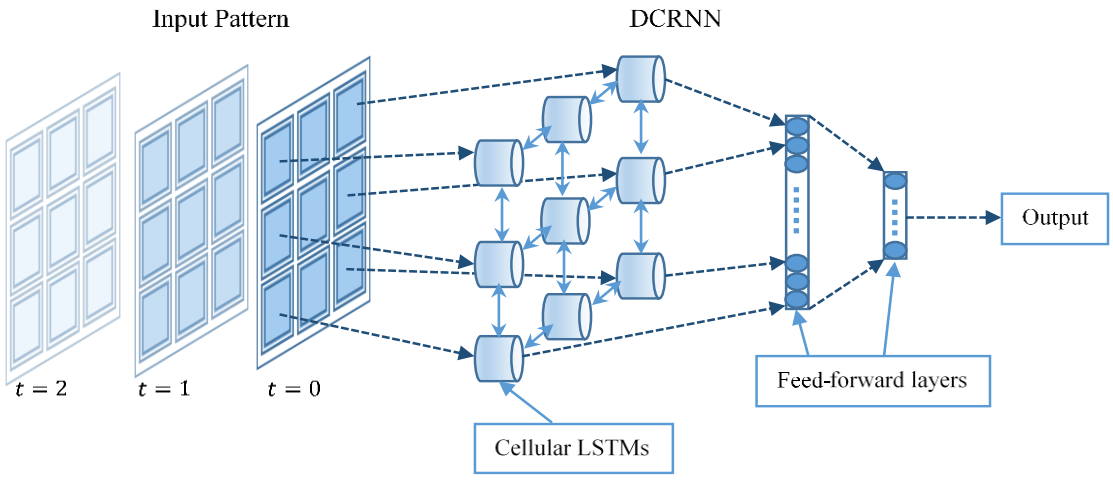

Fig. 22. Proposed DCRNN architecture. Each cell in the cellular sub-architecture hold a configurable LSTM network. Final outputs of each cell is aggregated and passed through a feed-forward network followed by classification.

Note that the cellular front end of the proposed architecture is expanded to overlap the multi-source 2D input pattern as illustrated in Fig. 22. This enables the LSTM network core in each cell to process the time series generated from the corresponding signal source simultaneously. The LSTM core network within each cell can be configured as needed for a particular task. However, we constrain the LSTM core architecture to be identical for each cell to ensure maximum trainable weight sharing. This novel DCRNN model, therefore, offers versatility of cellular neural processing combined with flexible time series processing of recurrent LSTM while keeping the spatial location information of the input sensor signal.

It is also evident from Fig. 22 that communication paths exist between a given cell and its four neighboring cells. The neighborhood information processing occurs at each time step. For instance, consider the cell j, k of the cellular grid of size $J \times K$ is processing a time series at time step t. Along with the input of time series at t, we configure an additional path to the core architecture coming from the neighbors $((j-1,k),(j+1,k),(j,K-1),(j,k+1))$ outputs obtained at time $t-1$. In order to accommodate this additional neighbor information path in a 2D cellular setting, we augment the LSTM equations (18) – (22) in section 2.4.2.2 by taking the DCRNN core at cell j, k as follows:

$$i_{j,k,t} = \sigma\left(W_i x_{j,k,t} + W_{Ni} N_{j,k,t} + U_i h_{t-1}\right), \tag{36}$$

$$f_{j,k,t} = \sigma\left(W_j x_{j,k,t} + W_{Nf} N_{j,k,t} + U_f h_{t-1}\right), \tag{37}$$

$$O_{j,k,t} = \sigma\left(W_o x_{j,k,t} + W_{No} N_{j,k,t} + U_o h_{t-1}\right), \tag{38}$$

$$S_{j,k,t} = f_{j,k,t} \odot S_{j,k,t-1} \odot tanh\left(W_s x_{j,k,t} + W_{Ns} N_{j,k,t} + U_s h_{t-1}\right), \quad (39)$$

$$h_{j,k,t} = O_{j,k,t} \odot \tanh\left(S_{j,k,t}\right) \quad (40)$$

where,

$$N_{j,k,t} = \left[h_{j-1,k,t-1}, h_{j+1,k,t-1}, h_{j,k-1,t-1}, h_{j,k+1,t-1}\right]. \quad (41)$$

Note that the previous time-step hidden output information of the four closest neighbors is used as an additional input signal $N_{j,k,t}$ for the LSTM network at each cell. With a $G \times 1$ dimensional hidden output per cell, we typically assign just one neuron output (the G^{th}element) as the output for neighbors. Though this is configurable, we have found that a single neighbor output per cell is sufficient for adequate performance.

The cellular configuration makes it necessary to hold cell specific intermediate, final hidden and memory outputs as shown in Eqns. (36) to (40). However, maintaining identical LSTM settings for each cell allows sharing of trainable parameters. Though only shown for a single LSTM layer, the cell core architecture can be expanded for multiple layers or bidirectional processing as necessary. The final outputs at time step T of each cell $h_{j,k,T}$ are aggregated to obtain the feature vector H. Subsequently, the feature vector H is passed through the feed-forward sub-net to obtain the final output as follows:

$$FF = \sigma\left(W_{ff} H + b_{ff}\right), \quad (42)$$

$$\bar{y} = softmax\left(W_{\bar{y}} FF + b_{\bar{y}}\right), \quad (43)$$

Given the ground truth classification as y the classification error E is computed using the Mean Squared Error based loss-function:

$$E = \frac{1}{2}\|y - \bar{y}\|_2^2; \tag{44}$$

The training of the network is performed by obtaining partial derivatives of feed-forward weights $\Delta W_{\bar{y}}$ and ΔW_{ff} using the standard back-propagation algorithm and ΔW_c using back-propagation through time across all cells. The detailed training procedure of the proposed DCRNN architecture is shown in Algorithm 4.

Algorithm 4: Training procedure of DCRNN

Initialization:

■ Set cellular sub-structure parameters $W_c \in \{W_i, W_j, W_o, W_s, W_N\}$ and feed-forward parameters $W_{ff}, W_{\bar{y}}$ with random values

Training:

for each epoch
  for each sample/batch
    for each time step $t \leftarrow 0\ to\ T$
      1. propagate through the cellular sub-net:

        for each cell $j, k \leftarrow 0\ to\ J\ and\ 0\ to\ K$:
          ▪ Obtain corresponding signal input $x_{j,k,t}$ and input from neighbors $N_{j,k,t}$
          ▪ Compute LSTM memory $S_{j,k,t}$ and hidden output $h_{j,k,t}$ using Eq.(6)-(10)
        end
      end
      2. Propagate through feed-forward sub-net:

        ▪ Aggregate $h_{j,k,T}$ from each cell j, k to form H
        ▪ Compute output $\bar{y}$ using Eq.(12),(13)
        ▪ Compute E using Eq. (14)

      3. Perform DCRNN back-propagation:

        ▪ Use standard back propagation to obtain $\Delta W_{\bar{y}}$, and ΔW_{ff}
        ▪ Use BPTT to obtain ΔW_c:
          for each time step $t \leftarrow T\ to\ 0$
            for each cell $j, k \leftarrow J\ to\ 0\ and\ K\ to\ 0$
              $\Delta W_c = \Delta W_c + \nabla_{W_c} E$
            end
          end
      4. Update the Weight parameters:

        ▪ $W_{\bar{y}}(new) = W_{\bar{y}}(old) - \alpha * \Delta W_{\bar{y}}$
        ▪ $W_{ff}(new) = W_{ff}(old) - \alpha * \Delta W_{ff}$
        ▪ $W_c(new) = W_c(old) - \alpha * \Delta W_c$
  end
end

5.3.2 COMPLEXITY ANALYSIS OF THE PROPOSED DCRNN ARCHITECTURE

This section analyzes the complexity of the proposed DCRNN architecture compared to a state-of-the-art deep LSTM (DLSTM) network of similar depth in terms of the $Big\ O$ notation. One clear advantage for DCRNN is the extensive use of weight sharing in the cellular recurrent sub-architecture as shown in Fig. 22 . This is evident especially when the DCRNN is used to process time series data with multiple signal sources spread in space, such as EEG. Consider a time series data sample at time-step t with $J \times K$ individual signal sources spread in a 2D space. The total number of parameters (N_{DCRNN}) of the DCRNN architecture is given by,

$$N_{DCRNN} = \underbrace{(n_{CLSTM} \times m)}_{\text{(LSTM weights in a cell)}} + \underbrace{\left(J \times K \times n_{ff}\right)}_{\text{(feed-forward weights)}} + \underbrace{c \times n_{ff}}_{\text{(classifier)}} \tag{45}$$

whereas, the required number of parameters (N_{DLSTM}) of a deep LSTM with similar depth is given by,

$$N_{DLSTM} = (n_{LSTM} \times m \times J \times K) + \left(n_{LSTM} \times n_{ff}\right) + c \times n_{ff} \tag{46}$$

Considering the fact that the LSTM network contains multiple trainable weights as shown in Eq. (18) to (22), the upper bound of the required number of parameters for the generic deep LSTNM (DLSTM) in presence of the above data is $O(n_{LSTM} \times m \times J \times K)$ where m denotes the dimensionality of the data in a single signal source. Conversely, the cellular architecture with weight sharing manages to process the same data with just $O(n_{CLSTM} \times m)$ complexity. Further note that typically $n_{LSTM} \gg n_{CLSTM}$ due to the large sensor signal input dimensionality faced by

the generic DLSTM architecture. In contrast, the DCRNN requires a very small amount of recurrent LSTM core units as the cellular architecture processes data from each sensor signal source separately.

5.3.3 MULTI-CHANNEL EEG PROCESSING WITH DCRNN

As discussed in Section 5.2, multi-channel scalp EEG data exhibits the characteristic of time series with spatial locality. The spatial locality may specifically be of interest in automated EEG signal processing as EEG signals collected at different locations in the brain represent specific seizure activity [131]. Accordingly, we utilize a multi-channel scalp EEG dataset known as the CHB-MIT EEG database [134]. This dataset consists of long-term multi-channel EEGs recorded from multiple pediatric patients with intractable seizures. More importantly, the scalp

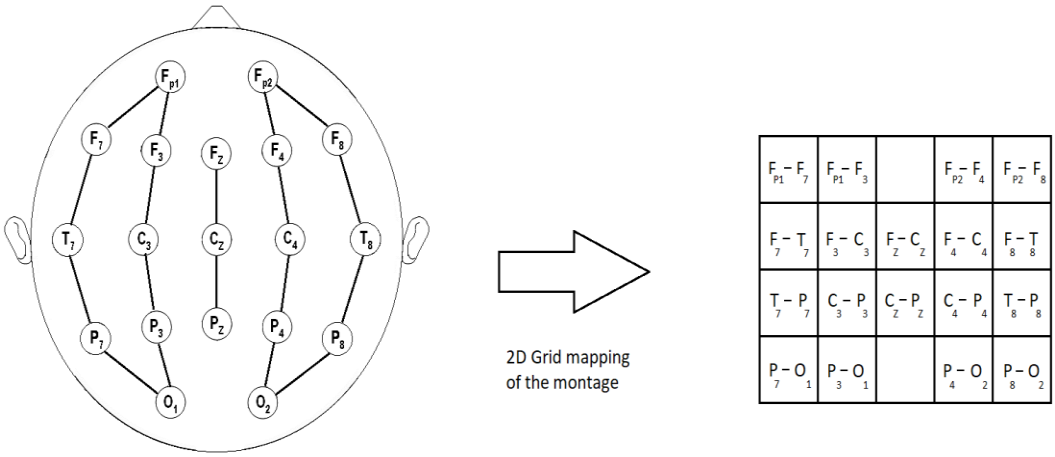

Fig. 23. The 2D grid mapping of the long-term bipolar EEG montage used in the CHB-MIT scalp EEG Dataset. The 2D grid approximation is a typical input to the proposed DCRNN.

EEG setup used in most cases contains 23 bipolar EEG signals recorded from individual electrodes placed according to the International Federation of Clinical Neurophysiology 10-20 system.

For effective processing of the EEG with spatial orientations intact, we map the EEG montage with 18 representative bipolar channels onto a 2D grid setting for better visualization as shown in Fig. 23. Note that the raw EEG signals localized as shown in Fig. 23 match with a 2D spatial input arrangement required for the proposed DCRNN architecture. In this study we utilize the mapping shown in Fig. 23. To obtain an input grid arrangement of size $J = 4$ and $K = 5$. Note the mapping in Fig. 4 is scalable so that any additional signal sources (channels) may be easily accommodated by rearranging the specified grid. This simply expands the cellular arrangement of the DCRNN correspondingly without additional complexity due to weight sharing. We utilize this dataset arrangement with the proposed DCRNN architecture to perform automated seizure detection.

5.3.4 MACHINE FAULT DETECTION WITH DCRNN

In order to investigate the versatility of the proposed DCRNN architecture, we utilize a second dataset for machine fault detection. The dataset is derived from a database maintained by the Jefferson National Laboratory based on the hardware specific faults encountered in the particle accelerator facility. A brief description of the hardware arrangement is as follows. The Continuous Electron Beam Accelerator Facility (CEBAF) at Jefferson Laboratory incorporates multiple cryomodules with superconducting radio frequency (SRF) cavities. Each cryomodule contains eight such cavities connected serially. A fault that occurs in any of these cavities disrupts the experimentation at the CBAF facility. More information on the facility, the

hardware, and the associated data can be found in [133]. In summary, multiple radio frequency (RF) signals are recorded from each SRF cavity in each cryomodule, and a database of recording with cavity faults are maintained for further study.

We utilize this database for automated multi-class fault detection with the proposed DCRNN. The cavities are arranged in a serial fashion within the cryomodule. We select 5 representative RF time series signals per cavity based on expert recommendation. We subsequently map the eight cavities and corresponding RF signals in a 2D grid layout as shown in Fig. 24. With this mapping, the 5 time series data from each cavity are separated in rows while the serial cavity arrangement is preserved in columns. This ultimately obtains a grid of size $J = 5$

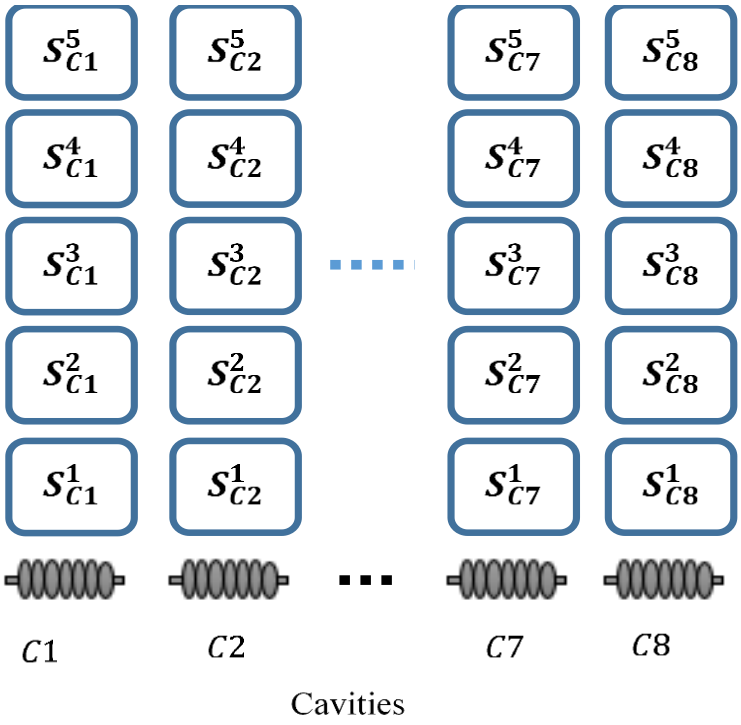

Fig. 24. The 2D time-sequence grid arrangement of the Jefferson Lab cavity fault detection database.

and $K = 8$, and an efficient 2D arrangement for the proposed DCRNN architecture.

5.3.5 NETWORK PREPARATION

This section summarizes the input specific configuration of the proposed DCRNN architecture.

First, the DCRNN architecture is prepared for analysis with the EEG dataset as follows. We first implement the cellular recurrent architecture based on the EEG input mapping shown in Fig. 23. In the cellular sub-net, we implement a bidirectional LSTM architecture with 5 LSTM units in each direction. Note that the bidirectional LSTM architecture is made identical in all cells to allow sharing of trainable weights. The outputs from the bidirectional architecture are aggregated across all cells and passed to the first feed-forward layer consisting of 50 neurons. The final classification layer is configured for two class classifications (seizure vs. non-seizure EEG) with softmax activation. The other feed-forward layers utilize sigmoid activation as discussed in Eq. (43). With this setup, each 1 second segment of EEG is classified as either normal or a seizure EEG.

Subsequently, the DCRNN architecture is reconfigured for the machine fault detection data analysis as follows. We implement the cellular recurrent architecture to complement the data mapping arrangement in Fig. 24. Accordingly, the cellular sub-architecture contains 40 individual cells in 5×8 configuration. Within each cell, we setup a unidirectional LSTM architecture consisting of 5 LSTM units. Like EEG, the LSTM sub-sub-architecture is made identical in each cell to ensure full weight sharing. Final outputs of LSTMs in each cell are aggregated and processed through a feed-forward layer consisting of 100 neurons following Eq. (42). The final classification layer is configured for a 5 class classification task with softmax activation. We classify each of the ~600 waveform events based on the corresponding fault class.

5.4 RESULTS AND DISCUSSION

In this section, we evaluate the performance of the proposed DCRNN model using two time series datasets such as CHB-MIT scalp EEG [134], and the Jefferson Lab machine fault dataset [133]. We specifically select these datasets from different application domains to evaluate the scalability of the proposed DCRNN architecture in multiple application domains.

5.4.1 MULTI-CHANNEL SCALP EEG DATASET

As discussed in section 5.3.3, the CHB-MIT scalp EEG dataset consists of a long-term bipolar referenced multi-channel EEG recorded from pediatric patients with epileptic seizures. We utilize EEG data from 20 patients containing 124 separate seizure events for the analysis. The EEG is recorded in continuous segments of 1 to 4 hours. All EEG time series signals are sampled at 256 Hz. The seizure events within the long-term EEG segments are annotated by an expert [134]. We perform patient specific seizure detection using the proposed DCRNN model.

The EEG preparation for analysis is as follows. We extract and segment all available raw seizure EEGs into 1 second segments. We subsequently segment the non-seizure EEG into 1 second segment and perform randomized sampling to obtain a patient specific dataset of seizure and non-seizure EEG. The dataset is then prepared with the mapping procedure specified in section 5.3.3 for analysis. Note that we simply normalize the raw EEG without any additional pre-processing or feature extraction for this analysis. The patient specific dataset is finally utilized in a 5-fold cross validation procedure to obtain the performance of the proposed architecture.

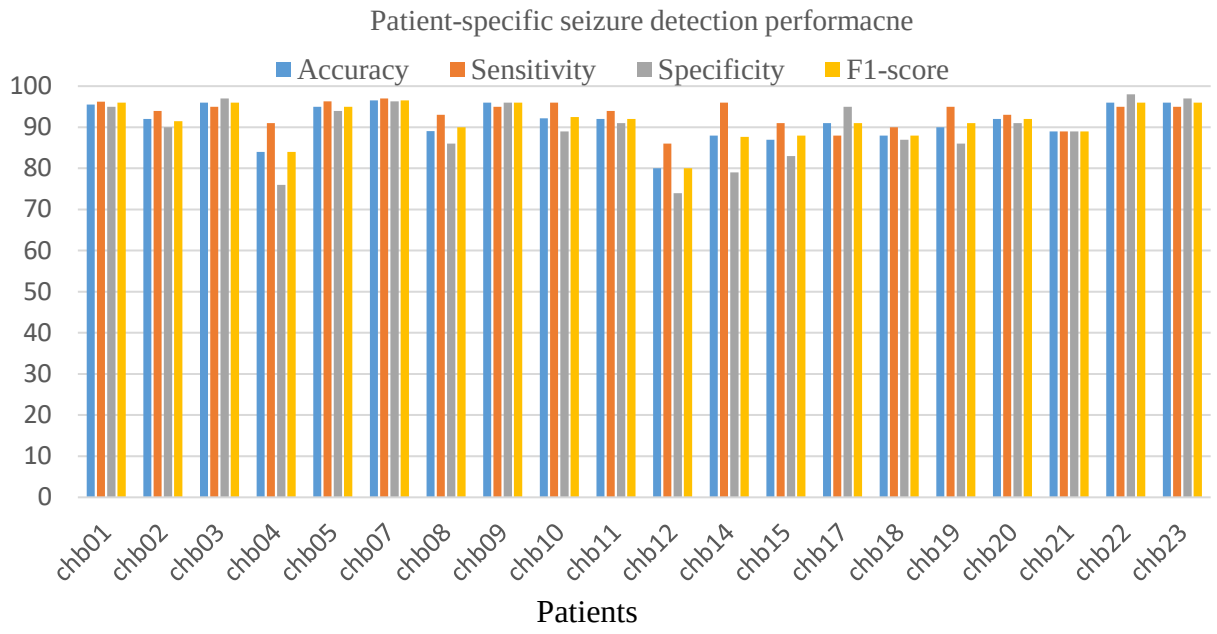

Fig. 25. Summary of patient specific seizure detection performance of the proposed DCRNN model

Fig. 25 summarizes the patient specific EEG classification results obtained with the DCRNN architecture. According to Fig. 25, seizure detection accuracy for most patients are well over 90%. Specifically, the DCRNN achieves an average accuracy of 91.3% with a median of 92.1%. However, when seizure detection criterion is considered, sensitivity score plays a more important role. This is because in a realistic setting, one would expect to correctly identify all seizure events even at the cost of relatively higher false positive numbers. Considering this, we show that the proposed architecture achieves an average sensitivity value of 94% with a median sensitivity of 95%. The DCRNN model still manages to maintain a median specificity value of

90.5%. The proposed model also achieves mean and median F1 scores of 91.4% and 92.25% respectively.

Table 10 compares the seizure detection performance of the proposed DCRNN model with other studies in the literature.

Table 10. Performance comparison of the proposed DCRNN model with other methods on seizure detection with scalp EEG

Methods	Sensitivity (%)	Specificity (%)	Accuracy (%)	Conventional Feature Engineering and Classification	Automated Feature Learning and Classification
Yuan et al. [135]	93.8	94.8	94.9	Yes	No
Subasi et al. [136]	93.10	92.8	93.1	No	Yes
Shoeb et al. [59]	96	-	-	Yes	No
Vidyaratne et al. [17]	96	-	-	Yes	No
Khan et al. [137]	83.6	100	91.8	Yes	No
Fergus et al. [138]	93	94	-	Yes	No
Yao et al. [139]	87.3	86.7	87	No	Yes
Park et al. [140]	80.8	91.7	85.6	No	Yes
DCRNN	**94**	**90**	**91.3**	**No**	**Yes**

Table 10 compares the seizure detection performance of the proposed DCRNN model with other studies in the literature. Table 10 shows that the proposed architecture manages to achieve comparable seizure detection performance to other state-of-the-art methods in the literature. Specifically, the sensitivity of seizure detection is only slightly lower than the methods in [59] and [17] and higher than that of [138], all of which utilize the same dataset. However, we point out that these methods use several complex feature extraction, feature selection, and classification methods in the associated seizure detection pipelines. For example, the seizure detection algorithm in [17] uses fractal dimension and harmonic wavelet packet transform on each EEG signal to extract features and subsequently utilizes relevance vector machine (RVM) classifier for seizure classification. The method in [59] constructs a filter bank to extract features from each EEG signal followed by classification with SVM. The method in [138] uses several features-based power spectral density (PSD) measures and temporal statistics of EEG time series data followed by feature selection methods such as principal component analysis (PCA) and linear discriminant analysis (LDA). The study then experiments with several classification models such as linear discriminant classifier, quadratic discriminant classifier, polynomial classifier, etc. and obtains the best sensitivity of 93% with a K-nearest neighbor (KNN) classifier. Yao et al. [139] propose a deep recurrent learning approach for seizure detection using the same dataset. The architecture includes 15 recurrent layers with different time-scale hierarchies that are composed of 128, 200, and 250 hidden recurrent units for each 5 layer block, respectively. Similarly, Park et al. [140] utilize a 7 layer CNN architecture containing both 1D and 2D convolutional filters to process a multi-channel EEG dataset for seizure detection.

In contrast, the proposed DCRNN contains only 5 bidirectional LSTM units in the recurrent hidden layers of each cell. With cellular weight sharing, we maintain the same number

of units among all cells that process corresponding channels. This comparison shows the highly superior computational efficiency of the proposed architecture. In summary, the proposed architecture performs efficient feature learning and classification simply utilizing minimally pre-processed EEG. Moreover, time series processing with LSTM is performed within the cellular sub-net, which allows for simultaneous processing of each EEG channel while considering the locality of electrodes on the scalp. Minimal pre-processing with automatic feature learning and efficient use of trainable weights make DCRNN desirable for multi-channel EEG processing applications.

5.4.1 MACHINE FAULT DETECTION DATASET

The Jefferson Labs machine fault detection dataset includes approximately 600 samples of cavity waveform data acquired from the particle accelerator system. Each sample contains 17 RF waveforms recorded from each of the 8 SRF cavities. Each waveform contains ~1.6 seconds (8196 time samples) of data that includes system failure due to a certain fault event. The dataset is inspected and categorized into 5 known fault types by an expert. An example waveform extracted from cavity 1 is shown in Fig. 26.

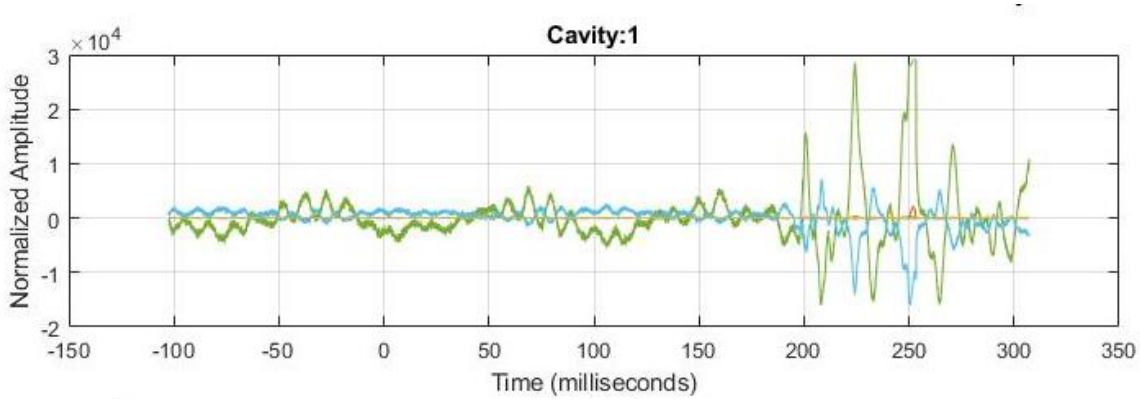

Fig. 26. Example RF waveforms extracted from cavity 1.

We prepare the machine fault data for analysis as follows. We select the 5 most significant RF waveforms for analysis based on visual analysis by an expert. We subsequently normalize the waveforms based on the z-score normalization technique. Even though the RF waveforms are sampled at a very high rate, we observe that the actual fault event is a relatively low frequency event. Therefore, we perform aggressive down sampling of the selected waveforms by a factor of 20 to obtain time series data of approx. 410 time samples. The data is subsequently arranged based on the mapping introduced in section 5.3.4 and visualized in Fig. 24. The dataset is utilized in a 10-fold cross validation process to obtain the performance of the proposed DCRNN architecture.

Table 11. Performance comparison of the proposed DCRNN model with other methods on machine fault detection dataset

Method	Number of recurrent units in each layer	10-fold accuracy ± standard deviation
AR features + SVM	-	90% ± 4%
AR features +RF	-	91.5% ± 2%
AR features + LR	-	87.4% ± 4.8%
Deep LSTM	256-256	88.83% ± 2.4%
DCRNN	**5-5**	**89.1% ± 2.7%**

In order to compare the performance of DCRNN on the fault classification dataset, we construct bidirectional LSTM architecture with two 256 LSTM units each followed by a feed forward layer of 512 neurons and a 5 class classification layer. Additionally, we set up a machine learning framework for the fault classification task. For this, we perform feature extraction on 5 selected waveforms utilizing autoregressive (AR) analysis. Accordingly, we obtain a 6-dimensional feature vector per waveform to construct a 240 ($6\ features \times 5\ waveforms \times 8 cavities$) element feature vector for each data sample. We subsequently perform 10-fold cross validation analysis using classifiers such as Logistic regression (LR), support vector machine (SVM), and Random Forrest (RF). The 10-fold cross validation performance of the proposed architecture along with comparison with other methods are shown in Table 11.

As shown in Table 11, between the two DL models, the proposed DCRNN offers comparable accuracy. However, note the large difference in hidden LSTM units used for the

recurrent layers in both deep LSTM and DCRNN. This is due to the cellular processing feature that maintains the location information for sensor signal in DCRNN as illustrated in Fig. 22. Therefore, the input dimensionality of the sensor signal per cell is quite small and will require a much smaller number of LSTM units per cell. Moreover, since the LSTM architecture is shared among cells, the number of trainable parameters does not grow. The ROC curve of DCRNN for multi-class processing is shown in Fig. 27.

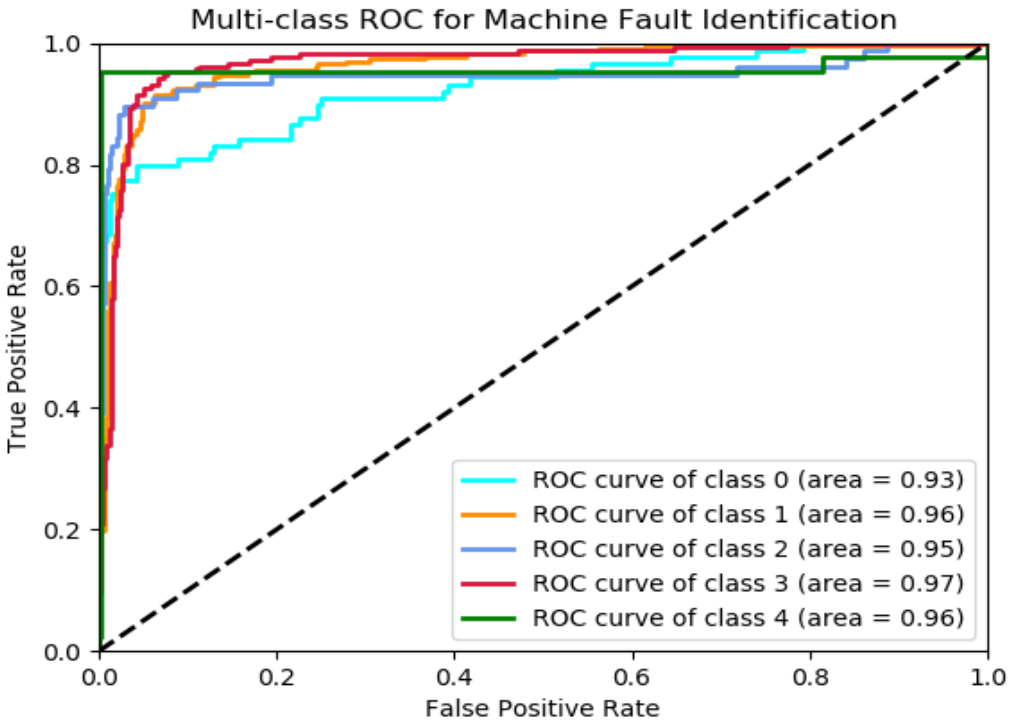

Fig. 27. ROC performance curve of DCRNN for multi-class machine fault detection.

Though the ML based methods in Table 11 perform slightly better than those of the proposed DCRNN model, we point out that the associated pipeline requires autoregressive

feature extraction from each RF waveform of each cavity. This may be a tedious and computationally intensive process, especially if the number of waveforms or cavities is higher. The proposed DCRNN architecture is quite helpful in this regard as it simply requires expanding the cellular grid to accommodate the increased input sources. The spatial domain trainable weight sharing functionality of cellular architecture would contain the computational complexity as analyzed in section 5.3.2.

5.5 SUMMARY

This chapter proposes a novel deep cellular recurrent neural network (DCRNN) architecture for efficient processing of large-scale time-series data with spatial relevance. The DCRNN model consists of a cellular recurrent sub-network that operates in 2D space to enable efficient processing of time series data while considering multiple signals from spatially distributed sensors. The cellular architecture processes data from each localized sensor signal source individually in a synchronized manner. This 2D distributed processing approach enables minimum use of recurrent LSTM units within each cell due to the locally reduced input dimensionality. Moreover, time series data obtained from spatially distributed sensor systems such as multi-channel EEG may hold importance in the locality of the sensor signal for many associated tasks. The cellular architecture of the proposed DCRNN preserves the locality of the distributed sensor signals by mapping itself onto the 2D space. The inter-cellular weight sharing property further improves the efficiency of the proposed model. The performance of the proposed DCRNN model is evaluated using two large-scale time series datasets obtained from biomedical and machine fault analysis domains. The results show that the proposed architecture

achieves state-of-the-art performance with respect to comparable ML and DL methods while utilizing significantly fewer recurrent processing units and trainable parameters.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

The overall goal of this dissertation is to propose a novel biologically inspired deep cellular recurrent neural network model for efficient processing of large-scale time series data with spatial information. We begin by investigating the efficacy of feature engineered machine learning models in time series analysis. Accordingly, as our first goal, we develop a novel, efficient ML pipeline to process a large-scale EEG dataset. The proposed pipeline utilizes computationally efficient feature extraction methods such as harmonic wavelet packet transform and fractal analysis to obtain state-of-the art performance among machine learning models on EEG processing for seizure detection. We also identify the potential limitations incurred by the feature engineering process and recognize the need for an efficient feature learning model for such large-scale time-series processing tasks. Therefore, we investigate a particular neural network architecture known as CSRN as a platform to build a novel recurrent architecture. Though quite adept at learning intricate tasks, the inherent structural complexity of the CSRN affects its training. Consequently, as our second goal, we perform a comprehensive investigation of training CSRN utilizing several constrained and unconstrained learning algorithms. Subsequently, as our final goal, we introduce a novel deep cellular recurrent architecture for efficient processing of large-scale time-series datasets. The overall outcomes of this dissertation are summarized in Table 12 and further discussed below.

Table 12. Summary of novel contributions

Chapter	Topic	Contributions	Results	Comments
3	Efficient ML pipeline for real time epileptic seizure detection using scalp EEG	• A novel machine learning pipeline for automated seizure detection • Feature engineering with harmonic wavelet (HW) analysis and fractal analysis (FD)	The proposed pipeline shows improved seizure detection performance compared to the state-of-the-art ML models	• The proposed method is significantly more computationally efficient with respect to other methods • Works with both scalp and intracranial EEG
4	Comparison of constrained and unconstrained learning for cellular simultaneous recurrent network (CSRN) in image processing	• Generalized structure of CSRN for a variety of spatial domain image processing • Introduced Jacobian free optimization for CSRN	The composite cellular architecture of CSRN is capable of learning complex topological image processing tasks using several optimization methods	• The cellular architecture aids in learning topological relevance of data effectively • Jacobian free optimization is appropriate for complex architectures with limited trainable parameters
5	Deep cellular recurrent neural network (DCRNN) for efficient time-series analysis	• Introduced a unique deep cellular recurrent architecture for large-scale time series processing	DCRNN model demonstrates comparable performance to state-of-the-art models while using substantially less computing resources	• DCRNN Processes both spatial and temporal relevance of data • The DCRNN requires significantly reduced training parameters achieve high performance

For the first goal, we propose a novel ML pipeline for efficient large-scale time series processing for patient-specific real-time automatic epileptic seizure onset detection. The proposed technique obtains the harmonic multiresolution and self-similarity-based fractal features from multi-channel EEG for robust seizure onset detection. Accordingly, a fast wavelet decomposition method, known as harmonic wavelet packet transform (HWPT), is computed based on Fourier transform to achieve higher frequency resolutions without the recursive calculations ubiquitous to generic methods. Similarly, we consider fractal dimension (FD) estimates in order to capture the self-similar repetitive patterns in the EEG signal. Subsequently, we organize both FD and HWPT energy features across all EEG channels at each epoch to capture the spatial information due to electrode placement on the skull. Moreover, we construct the final feature vector using a moving window analysis to capture the temporal information of EEG. Our experimental results using a large-scale multi-channel scalp EEG dataset and a small-scale intracranial EEG dataset show that the proposed pipeline outperforms the state-of-the-art methods on multi-channel EEG processing for automated seizure detection. Moreover, the results show that use of less computationally intensive feature extraction techniques such as HWPT and FD enables considerably faster seizure onset detection when compared to similar techniques in the literature. We also show that the superior speed indicates potential usage in real-time applications. We recognize the potential limitations of feature engineering and the difficulty in processing the spatial locality of large-scale time series such as EEG with traditional methods.

For the second goal, we investigate a unique artificial neural network architecture known as cellular simultaneous recurrent neural network (CSRN) and its training. Many practical applications such as large-scale spatial domain image processing represents extremely complex

topological mapping functions that typical ANN architectures are unable to approximate effectively. However, the distinctive cellular architecture makes CSRN particularly adept at such topological mapping tasks typically found in image data processing. However, training CSRN for topological image processing is a challenging task because of the complexity of the network and the sheer size of the image data. Several representative training algorithms such as Back-propagation Through Time (BPTT), Extended Kalman Filtering (EKF), and Particle Swarm Optimization (PSO) have been used to train CSRN for different applications. However, the literature does not show a systematic approach for choosing an appropriate learning algorithm for training a complex universal approximator such as the CSRN in generalized image processing applications. Consequently, we first develop a generalization of the network architecture for several topological image processing tasks. Then we perform a systematic comparison of the CSRN network training algorithms developed to effectively solve the topological image mapping problems. Consequently, we show that introduction of an unconstrained Jacobian free UKF learning algorithm alleviates the high computational cost otherwise associated with calculating Jacobian for EKF. Moreover, our results show that Jacobian-free training algorithms such as UKF and PSO for unconstrained optimizers perform better than a Jacobian-based algorithm such as EKF in large scale image data processing. We also note that the Jacobian free training methods are efficient on architectures with a limited number of trainable parameters.

Finally, for the third goal we characterize and develop a novel bidirectional deep cellular recurrent neural network (DCRNN) architecture for efficient processing of large scale high dimensional time series data obtained from spatially distributed multi-sensor systems. Specifically, we design our DCRNN architecture based on the cellular structure of the CSRN

investigated under our second goal in the fourth chapter. The cellular recurrent architecture in the proposed model allows for location-aware synchronous processing of time series data obtained from spatially distributed multiple sensor signal sources. Additionally, we show that the extensive trainable parameter sharing enabled by the cellular architecture ensures superior efficiency in the use of recurrent processing units with high-dimensional inputs. We also demonstrate the generalizability of the novel DCRNN architecture utilizing two large-scale multi-sensor time series datasets: a multichannel scalp EEG dataset for a binary seizure classification task, and a multi-class machine fault detection time series dataset. The results indicate that the proposed DCRNN architecture achieves state-of-the-art performance with respect to comparable ML and DL methods while utilizing substantially fewer recurrent processing units and trainable parameters.

Our plan for the DCRNN is to further expand its efficacy and scalability by promoting parallelized information processing. The cellular structure of DCRNN inherently promotes parallelized processing of information that is organized in 2D space. As discussed in Chapters 4 and 5, each cell in a cellular architecture processes data independently and in a synchronized manner. Therefore, cellular processing can essentially be performed in a distributed manner. However, in order to fully leverage the parallelized processing in cells, the implementation of the architecture may be realized in a dedicated multi-processor hardware system such as a graphics processing unit (GPU), or a field-programmable gate array (FPGA). The sparse use of trainable parameters may further assist the implementation of DCRNN in a dedicated, stand-alone, lightweight hardware system. Such implementation may substantially improve the processing time of DCRNN to allow for potential real-time and real-world use. Additionally, we plan to further extend the DCRNN architecture for multi-dimensional time-series processing

applications. Currently, the cellularity in DCRNN allows for flexible expansion of architecture in a 2D grid-like input environment. However, some large-scale time series data, such as fMRI may contain time-series signal-sources spread in a 3D volumetric space. The CDRNN may be expanded to obtain a 3D cellular architecture for processing such volumetric time-series data efficiently. Another area of investigation is in the ability of DCRNN for transfer learning. Most generic DL models are susceptible to abrupt changes in the input dimensionality. For instance, a reduction or an addition of an EEG channel changes the input dimensionality of the multi-channel EEG data. This may require a generic deep recurrent neural network model to be retrained to accommodate the change in the input. However, the cellular structure of DCRNN may simply expand by adding a new cell to accommodate the additional input. The weight sharing property of DCRNN may further aid in this process and may require only minimal adjustments. Further investigations into this property may reveal possibilities of using DCRNN for the intricate problem of missing data. In general, the findings of this dissertation suggest that future research in efficient deep learning may need to focus on architectures that enable decentralized processing of information to support the data analysis requirements of prevailing multi-sensor environments in modern computational intelligence.

REFERENCES

[1] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE transactions on intelligent transportation systems,* vol. 12, no. 2, pp. 596-614, 2010.

[2] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," 2006.

[3] M. Havaei *et al.*, "Brain tumor segmentation with deep neural networks," *Medical image analysis,* vol. 35, pp. 18-31, 2017.

[4] A. S. Zandi, G. A. Dumont, M. Javidan, and R. Tafreshi, "Detection of Epileptic Seizures in Scalp Electroencephalogram: An Automated Real-Time Wavelet-Based Approach," *Journal of Clinical Neurophysiology,* vol. 29, no. 1, pp. 1-16, 2012.

[5] E. Putin *et al.*, "Deep biomarkers of human aging: application of deep neural networks to biomarker development," *Aging (Albany NY),* vol. 8, no. 5, p. 1021, 2016.

[6] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, no. 6, pp. 1190-1203, 2012.

[7] R. Yan and R. X. Gao, "An efficient approach to machine health diagnosis based on harmonic wavelet packet transform," *Robotics and Computer-Integrated Manufacturing,* vol. 21, no. 4-5, pp. 291-301, 2005.

[8] G. Dorffner, "Neural networks for time series processing," in *Neural network world*, 1996: Citeseer.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[10] G. E. Hinton, "Learning multiple layers of representation," *Trends in cognitive sciences,* vol. 11, no. 10, pp. 428-434, 2007.

[11] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery,* vol. 26, no. 2, pp. 275-309, 2013.

[12] T. W. Liao, "Clustering of time series data—a survey," *Pattern recognition,* vol. 38, no. 11, pp. 1857-1874, 2005.

[13] L. Wang, X. Wang, C. Leckie, and K. Ramamohanarao, "Characteristic-based descriptors for motion sequence recognition," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2008, pp. 369-380: Springer.

[14] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based classification of time-series data," *International Journal of Computer Research,* vol. 10, no. 3, pp. 49-61, 2001.

[15] X. Wang, A. Wirth, and L. Wang, "Structure-based statistical features and multivariate time series clustering," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007, pp. 351-360: IEEE.

[16] F. Mörchen, "Time series feature extraction for data mining using DWT and DFT," ed: Univ., 2003.

[17] L. S. Vidyaratne and K. M. Iftekharuddin, "Real-Time Epileptic Seizure Detection Using EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 25, no. 11, pp. 2146-2156, 2017.

[18] A. Shoeb, H. Edwards, J. Connolly, B. Bourgeois, S. Ted Treves, and J. Guttag, "Patient-specific seizure onset detection," *Epilepsy & Behavior,* vol. 5, no. 4, pp. 483-498, 8// 2004.
[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, no. Mar, pp. 1157-1182, 2003.
[20] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.
[21] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* vol. 5, no. 4, pp. 115-133, 1943.
[22] S.-Y. King and J.-N. Hwang, "Neural network architectures for robotic applications," *IEEE Transactions on Robotics and Automation,* vol. 5, no. 5, pp. 641-657, 1989.
[23] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553-2561.
[24] M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Efficient feature extraction with simultaneous recurrent network for metric learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1195-1201.
[25] D. P. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc., 2001.
[26] C. Chatfield, *The analysis of time series: an introduction*. Chapman and Hall/CRC, 2003.
[27] T. Koskela, M. Lehtokangas, J. Saarinen, and K. Kaski, "Time series prediction with multilayer perceptron, FIR and Elman neural networks," in *Proceedings of the World Congress on Neural Networks*, 1996, pp. 491-496: Citeseer.
[28] D. T. Pham and D. Karaboga, "Training Elman and Jordan networks for system identification using genetic algorithms," *Artificial Intelligence in Engineering,* vol. 13, no. 2, pp. 107-117, 1999.
[29] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," 1999.
[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.
[31] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 2, pp. 1539-1548, 2017.
[32] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *2013 IEEE workshop on automatic speech recognition and understanding*, 2013, pp. 273-278: IEEE.
[33] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.
[34] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural networks,* vol. 18, no. 5-6, pp. 602-610, 2005.
[35] J. J. Nassi and E. M. Callaway, "Parallel processing strategies of the primate visual system," *Nature Reviews Neuroscience,* vol. 10, no. 5, pp. 360-372, 2009.
[36] T. Pasternak, J. W. Bisley, and D. Calkins, "Visual processing in the primate brain," *Handbook of psychology,* 2003.
[37] R. T. Born, "Visual processing: parallel-er and parallel-er," *Current Biology,* vol. 11, no. 14, pp. R566-R568, 2001.

[38] S. Zeki, "■ REVIEW: Parallel Processing, Asynchronous Perception, and a Distributed System of Consciousness in Vision," *The Neuroscientist*, vol. 4, no. 5, pp. 365-372, 1998.
[39] N. H. Yabuta, A. Sawatari, and E. M. Callaway, "Two functional channels from primary visual cortex to dorsal visual cortical areas," *Science*, vol. 292, no. 5515, pp. 297-300, 2001.
[40] M. S. Gazzaniga, *The cognitive neurosciences*. MIT press, 2004.
[41] G. Field and E. Chichilnisky, "Information processing in the primate retina: circuitry and coding," *Annu. Rev. Neurosci.*, vol. 30, pp. 1-30, 2007.
[42] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257-1272, 1988.
[43] L. O. Chua and L. Yang, "Cellular neural networks: applications," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1273-1290, 1988.
[44] L. Vidyaratne, M. Alam, J. K. Anderson, and K. M. Iftekharuddin, "Improved training of cellular SRN using Unscented Kalman Filtering for ADP," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 993-1000.
[45] X. Pang and P. Werbos, "Neural network design for J function approximation in dynamic programming," *arXiv preprint adap-org/9806001*, 1998.
[46] J. K. Anderson and K. M. Iftekharuddin, "Learning topological image transforms using cellular simultaneous recurrent networks," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1-9.
[47] L. Vidyaratne, M. Alam, J. K. Anderson, and K. M. Iftekharuddin, "Improved training of cellular SRN using Unscented Kaiman Filtering for ADP," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, 2014, pp. 993-1000.
[48] L. Vidyaratne, M. Alam, J. K. Anderson, and K. M. Iftekharuddin, "Constrained versus unconstrained learning in generalized recurrent network for image processing," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3310-3317.
[49] M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, *Cellular recurrent deep network for image registration* (SPIE Optical Engineering + Applications). SPIE, 2015.
[50] M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel hierarchical Cellular Simultaneous Recurrent neural Network for object detection," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-7.
[51] K. M. Iftekharuddin, M. Alam, and L. Vidyaratne, *Contemporary deep recurrent learning for recognition* (SPIE Defense + Security). SPIE, 2017.
[52] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, 2015, p. 89: Presses universitaires de Louvain.
[53] R. Esteller *et al.*, "Fractal dimension characterizes seizure onset in epileptic patients," vol. 4, pp. 2343-2346: IEEE.
[54] G. E. Polychronaki *et al.*, "Comparison of fractal dimension estimation algorithms for epileptic seizure onset detection," *Journal of neural engineering*, vol. 7, no. 4, p. 046007, 2010.
[55] D. Kaplan and L. Glass, *Understanding nonlinear dynamics*. Springer Science & Business Media, 2012.
[56] M. J. Katz and E. B. George, "Fractals and the analysis of growth paths," *Bulletin of mathematical biology*, vol. 47, no. 2, pp. 273-286, 1985.

[57] L. S. Liebovitch and T. Toth, "A fast algorithm to determine fractal dimensions by box counting," *physics Letters A,* vol. 141, no. 8-9, pp. 386-390, 1989.
[58] A. S. Zandi, M. Javidan, G. A. Dumont, and R. Tafreshi, "Automated real-time epileptic seizure detection in scalp EEG recordings using an algorithm based on wavelet packet transform," *IEEE Transactions on Biomedical Engineering,* vol. 57, no. 7, pp. 1639-1651, 2010.
[59] A. H. Shoeb and J. V. Guttag, "Application of machine learning to epileptic seizure detection," pp. 975-982.
[60] M. E. Saab and J. Gotman, "A system to detect the onset of epileptic seizures in scalp EEG," *Clinical Neurophysiology,* vol. 116, no. 2, pp. 427-442, 2// 2005.
[61] L. Kuhlmann *et al.,* "Seizure detection using seizure probability estimation: Comparison of features used to detect seizures," *Annals of biomedical engineering,* vol. 37, no. 10, pp. 2129-2145, 2009.
[62] D. E. Newland, "Harmonic wavelet analysis," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences,* vol. 443, no. 1917, pp. 203-225, 1993.
[63] K. M. Iftekharuddin, "Harmonic wavelet joint transform correlator: analysis, algorithm, and application to image de-noising," *Optical Engineering,* vol. 41, no. 12, pp. 3307-3316, 2002.
[64] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1998, p. 842.
[65] M. Minsky and P. Seymour, "Perceptrons," 1969.
[66] P. J. Werbos, *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. John Wiley & Sons, 1994.
[67] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling,* vol. 5, p. 3, 1988.
[68] C. Von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological cybernetics,* vol. 54, no. 1, pp. 29-40, 1986.
[69] H.-J. Chang and W. J. Freeman, "Parameter optimization in models of the olfactory neural system," *Neural Networks,* vol. 9, no. 1, pp. 1-14, 1996.
[70] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on,* vol. 45, no. 11, pp. 2673-2681, 1997.
[71] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019,* 2015.
[72] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.
[73] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," pp. 628-644: Springer.
[74] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *Artificial Neural Networks–ICANN 2006*: Springer, 2006, pp. 632-640.
[75] K. H. Pribram, *Origins: Brain and self organization*. Psychology Press, 1994.
[76] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE,* vol. 78, no. 10, pp. 1550-1560, 1990.
[77] L. O. Chua and L. Yang, "Cellular neural networks: theory," *Circuits and Systems, IEEE Transactions on,* vol. 35, no. 10, pp. 1257-1272, 1988.

[78] X. Pang and P. Werbos, "Neural Network Design for J Function Approximation in Dynamic Programming," vol. 1, ed: arXiv:adap-org/9806001, June 1998.
[79] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering,* vol. 83, no. 1, pp. 95-108, 1961.
[80] S. Haykin, Kalman Filtering and Neural Networks, Hoboken, NJ: John Wiley & Sons,, 2001, p. 1 online resource (302 p.). [Online]. Available: http://proxy.lib.odu.edu/login?url=http://search.ebscohost.com/direct.asp?db=iih&jid=VOF&scope=site.
[81] R. Ilin, R. Kozma, and P. J. Werbos, "Cellular SRN Trained by Extended Kalman Filter Shows Promise for ADP," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 2006, pp. 506-510.
[82] E. White, "Clustering for Improved Learning in Maze Traversal Problem," *arXiv preprint arXiv:0908.0939,* 2009.
[83] K. Anderson, K. Iftekharuddin, E. White, and P. Kim, "Binary image registration using cellular simultaneous recurrent networks," in *Computational Intelligence for Multimedia Signal and Vision Processing, 2009. CIMSVP '09. IEEE Symposium on*, 2009, pp. 61-67.
[84] J. Gotman, "Automatic recognition of epileptic seizures in the EEG," *Electroencephalography and Clinical Neurophysiology,* vol. 54, no. 5, pp. 530-540, 11// 1982.
[85] A. J. Gabor, R. R. Leach, and F. U. Dowla, "Automated seizure detection using a self-organizing neural network," *Electroencephalography and Clinical Neurophysiology,* vol. 99, no. 3, pp. 257-266, 9// 1996.
[86] A. S. Zandi, M. Javidan, G. A. Dumont, and R. Tafreshi, "Automated Real-Time Epileptic Seizure Detection in Scalp EEG Recordings Using an Algorithm Based on Wavelet Packet Transform," *Biomedical Engineering, IEEE Transactions on,* vol. 57, no. 7, pp. 1639-1651, 2010.
[87] R. Hopfengärtner *et al.*, "Automatic seizure detection in long-term scalp EEG using an adaptive thresholding technique: A validation study for clinical routine," *Clinical Neurophysiology,* vol. 125, no. 7, pp. 1346-1352, 7// 2014.
[88] M. Katz and E. George, "Fractals and the analysis of growth paths," (in English), *Bulletin of Mathematical Biology,* vol. 47, no. 2, pp. 273-286, 1985/03/01 1985.
[89] A. Accardo, M. Affinito, M. Carrozzi, and F. Bouquet, "Use of the fractal dimension for the analysis of electroencephalographic time series," (in English), *Biological Cybernetics,* vol. 77, no. 5, pp. 339-350, 1997/11/01 1997.
[90] A. Eke, P. Herman, L. Kocsis, and L. R. Kozak, "Fractal characterization of complexity in temporal physiological signals," *Physiological measurement,* vol. 23, no. 1, p. R1, 2002.
[91] P. C. Ivanov *et al.*, "Multifractality in human heartbeat dynamics," *Nature,* vol. 399, no. 6735, pp. 461-465, 1999.
[92] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Physica D: Nonlinear Phenomena,* vol. 31, no. 2, pp. 277-283, 1988.
[93] G. E. Polychronaki *et al.*, "Comparison of fractal dimension estimation algorithms for epileptic seizure onset detection," in *BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on*, 2008, pp. 1-6.
[94] M. J. Katz, "Fractals and the analysis of waveforms," *Computers in biology and medicine,* vol. 18, no. 3, pp. 145-156, 1988.

[95] P. Asvestas, G. K. Matsopoulos, and K. S. Nikita, "Estimation of fractal dimension of images using a fixed mass approach," *Pattern Recognition Letters,* vol. 20, no. 3, pp. 347-354, 3// 1999.

[96] P. van Mierlo *et al.*, "Functional brain connectivity from EEG in epilepsy: Seizure prediction and epileptogenic focus localization," *Progress in Neurobiology,* vol. 121, pp. 19-35, 10// 2014.

[97] R. B. Yaffe *et al.*, "Physiology of functional and effective networks in epilepsy," *Clinical Neurophysiology,* vol. 126, no. 2, pp. 227-236, 2// 2015.

[98] R. Yaffe *et al.*, "Brain state evolution during seizure and under anesthesia: A network-based analysis of stereotaxic eeg activity in drug-resistant epilepsy patients," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 5158-5161.

[99] Q. Yuan, W. Zhou, S. Li, and D. Cai, "Epileptic EEG classification based on extreme learning machine and nonlinear features," *Epilepsy Research,* vol. 96, no. 1–2, pp. 29-38, 9// 2011.

[100] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E,* vol. 64, no. 6, p. 061907, 2001.

[101] A. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation,* vol. 101, no. 23, pp. e215-e220, 06/13/ 2000.

[102] S. Santaniello, S. P. Burns, A. J. Golby, J. M. Singer, W. S. Anderson, and S. V. Sarma, "Quickest detection of drug-resistant seizures: An optimal control approach," *Epilepsy & Behavior,* vol. 22, pp. S49-S60, 2011.

[103] U. R. Acharya, F. Molinari, S. V. Sree, S. Chattopadhyay, K.-H. Ng, and J. S. Suri, "Automated diagnosis of epileptic EEG using entropies," *Biomedical Signal Processing and Control,* vol. 7, no. 4, pp. 401-408, 7// 2012.

[104] S. S. Spencer, P. Guimaraes, A. Katz, J. Kim, and D. Spencer, "Morphological patterns of seizures recorded intracranially," *Epilepsia,* vol. 33, no. 3, pp. 537-545, 1992.

[105] R. Esteller *et al.*, "Fractal dimension characterizes seizure onset in epileptic patients," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, 1999, vol. 4, pp. 2343-2346 vol.4.

[106] J. A. McEwen and G. B. Anderson, "Modeling the stationarity and gaussianity of spontaneous electroencephalographic activity," *Biomedical Engineering, IEEE Transactions on,* no. 5, pp. 361-369, 1975.

[107] S. P. Burns *et al.*, "Network dynamics of the brain and influence of the epileptic seizure onset zone," *Proceedings of the National Academy of Sciences,* vol. 111, no. 49, pp. E5321-E5330, December 9, 2014 2014.

[108] A. M. Chan, F. T. Sun, E. H. Boto, and B. M. Wingeier, "Automated seizure onset detection for accurate onset time determination in intracranial EEG," *Clinical Neurophysiology,* vol. 119, no. 12, pp. 2687-2696, 12// 2008.

[109] C. C. Jouny, P. J. Franaszczuk, and G. K. Bergey, "Improving early seizure detection," *Epilepsy & Behavior,* vol. 22, pp. S44-S48.

[110] A. Shoeb, A. Kharbouch, J. Soegaard, S. Schachter, and J. Guttag, "A machine-learning algorithm for detecting seizure termination in scalp EEG," *Epilepsy & Behavior*, vol. 22, Supplement 1, no. 0, pp. S36-S43, 12// 2011.
[111] P. Fergus, D. Hignett, A. Hussain, D. Al-Jumeily, and K. Abdel-Aziz, "Automatic Epileptic Seizure Detection Using Scalp EEG and Advanced Artificial Intelligence Techniques," *BioMed Research International*, vol. 2015, p. 17, 2015, Art. no. 986736.
[112] K. Samiee, P. Kovacs, and M. Gabbouj, "Epileptic Seizure Classification of EEG Time-Series Using Rational Discrete Short-Time Fourier Transform," *Biomedical Engineering, IEEE Transactions on*, vol. 62, no. 2, pp. 541-552, 2015.
[113] H. Ocak, "Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm," *Signal Processing*, vol. 88, no. 7, pp. 1858-1867, 7// 2008.
[114] L. Guo, D. Rivero, and A. Pazos, "Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks," *Journal of Neuroscience Methods*, vol. 193, no. 1, pp. 156-163, 10/30/ 2010.
[115] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic Seizure Detection Based on Time-Frequency Analysis and Artificial Neural Networks," *Computational Intelligence and Neuroscience*, vol. 2007, p. 80510, 12/05
[116] Y. Song, J. Crowcroft, and J. Zhang, "Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine," *Journal of Neuroscience Methods*, vol. 210, no. 2, pp. 132-146, 9/30/ 2012.
[117] M. L. Minsky and S. A. Papert, *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*. MIT press Boston, MA:, 1987.
[118] P. J. Werbos and P. Xiaozhong, "Generalized maze navigation: SRN critics solve what feedforward or Hebbian nets cannot," in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, 1996, vol. 3, pp. 1764-1769 vol.3.
[119] G. K. Venayagamoorthy and G. Singhal, "Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training MLP and SRN neural networks," *Journal of Computational and Theoretical Nanoscience*, vol. 2, no. 4, pp. 561-568, 2005.
[120] R. Ilin, R. Kozma, and P. J. Werbos, "Beyond feedforward models trained by backpropagation: A practical training tool for a more efficient universal approximator," *Neural Networks, IEEE Transactions on*, vol. 19, no. 6, pp. 929-937, 2008.
[121] B. Todorovic, M. Stankovic, and C. Moraga, "On-line learning in recurrent neural networks using nonlinear Kalman filters," in *Signal Processing and Information Technology, 2003. ISSPIT 2003. Proceedings of the 3rd IEEE International Symposium on*, 2003, pp. 802-805.
[122] J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. 4, pp. 1942-1948: Perth, Australia.
[123] T.-H. Kim and D. C. Wunsch, "Modified cellular simultaneous recurrent networks with cellular particle swarm optimization," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 2012, pp. 1-8: IEEE.
[124] X. Cai, G. K. Venayagamoorthy, and D. C. Wunsch Ii, "Evolutionary swarm neural network game engine for Capture Go," *Neural Networks*, vol. 23, no. 2, pp. 295-305, 2010.
[125] R. C. Gonzalez and R. E. Woods, "Digital image processing, 2nd," *SL: Prentice Hall*, 2002.
[126] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[127] E. A. Wan and R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153-158.
[128] M. S. Sohail, M. O. B. Saeed, S. Z. Rizvi, M. Shoaib, and A. U. H. Sheikh, "Low-Complexity Particle Swarm Optimization for Time-Critical Applications," *arXiv preprint arXiv:1401.0546*, 2014.
[129] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 643-660, 2001.
[130] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Computers in Biology and Medicine*, vol. 100, pp. 270-278, 2018/09/01/ 2018.
[131] P. Thodoroff, J. Pineau, and A. Lim, "Learning robust features using deep learning for automatic seizure detection," pp. 178-190.
[132] P. W. Mirowski, Y. LeCun, D. Madhavan, and R. Kuzniecky, "Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG," in *2008 IEEE Workshop on Machine Learning for Signal Processing*, 2008, pp. 244-249.
[133] A. Solopova *et al.*, "SRF cavity fault classification using machine learning at CEBAF," pp. 1167-1170: JACOW Publishing, Geneva, Switzerland.
[134] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215-e220, 2000.
[135] Q. Yuan, W. Zhou, Y. Liu, and J. Wang, "Epileptic seizure detection with linear and nonlinear features," *Epilepsy & Behavior*, vol. 24, no. 4, pp. 415-421, 2012.
[136] A. Subasi, "Automatic detection of epileptic seizure using dynamic fuzzy neural networks," *Expert Systems with Applications*, vol. 31, no. 2, pp. 320-328, 2006.
[137] Y. U. Khan, N. Rafiuddin, and O. Farooq, "Automated seizure detection in scalp EEG using multiple wavelet scales," pp. 1-5: IEEE.
[138] P. Fergus, D. Hignett, A. Hussain, D. Al-Jumeily, and K. Abdel-Aziz, "Automatic epileptic seizure detection using scalp EEG and advanced artificial intelligence techniques," *BioMed research international*, vol. 2015, 2015.
[139] X. Yao, Q. Cheng, and G.-Q. Zhang, "A Novel Independent RNN Approach to Classification of Seizures against Non-seizures," *arXiv preprint arXiv:1903.09326*, 2019.
[140] C. Park *et al.*, "Epileptic seizure detection for multi-channel EEG with deep convolutional neural network," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, 2018, pp. 1-5: IEEE.

VITA

Lasitha Vidyaratne

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

EDUCATION

- Old Dominion University, Norfolk, Virginia, USA GPA: 3.86

  Ph.D Candidate, Electrical and Computer Engineering, January 2012 - Present

- University of Nottingham, Semenyih, Malaysia

  Meng(Hons), Electronic and Communications Engineering, July 2009

ACADEMIC EXPERIENCE

- Old Dominion University, Norfolk, Virginia, USA
  - Graduate Research Assistant, January 2012 – Present
  - Teaching Assistant, - Spring 2019
- University of Moratuwa, Colombo, Sri Lanka
  - Instructor, Electrical and Telecommunications Engineering, March 2010 – December 2011

HONORS AND AWARDS

- Best Teaching Assistant award in 2016, Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, Virginia, USA

- Ranked 1st place in world for BraTS 2017 competition in brain tumor segmentation and survival prediction.
- Ranked 2nd place in world for MICCAI 2019 competition in brain tumor subtype classification.
- Top 10 finalists in the Old Dominion University inaugural 3-Minute Thesis (3MT) presentation competition, 2017.

PATENT

- Deep Cellular Recurrent Network for Efficient Analysis of Time-Series Data with Spatial Information (in preparation for submission)

GRANT PROPOSAL

- Authored several sections of a grant proposal for Thomas Jefferson National Accelerator Facility (Jefferson Lab)
    - Title: Machine Learning Based Cavity Fault Classification and Prediction
    - Start Date: 10/2019

PROFESSIONAL SERVICES

- Reviewing Activity
    - Serving regularly as a reviewer for IEEE Transactions on Neural Systems and Rehabilitation Engineering
    - Served as reviewer for Journal of Biomedical Research
    - Served as reviewer for IEEE Access

- Served as reviewer for Journal of integrated Design and Process Science
- Served as reviewer for International Joint Conference on Neural Networks 2018 and 2019

- Mentoring
  - Mentored junior graduate and undergraduate students in deep learning and digital signal/image processing research at ODU Vision Lab
  - Mentored visiting undergraduate research students from different U.S. Universities, awarded by the NSF Research Experiences for Undergraduates (REU) program.

LIST OF PUBLICATIONS

JOURNALS

- L. S. Vidyaratne, and K. M. Iftekharuddin, "Real-Time Epileptic Seizure Detection Using EEG," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 2146-2156, Nov. 2017.
- L. S. Vidyaratne, M. Alam, A. Glandon, and K. M. Iftekharuddin, "Deep Cellular Recurrent Network for Efficient Analysis of Time-Series Data with Spatial Information" (in preparation) for IEEE Transactions on Neural Networks and Learning Systems, 2019
- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Sparse Simultaneous Recurrent Deep Learning for Robust Facial Expression Recognition" in IEEE Transactions on Neural Networks and Learning Systems, vol. PP, no. 99, pp. 1-12, 2018.
- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel deep generative simultaneous recurrent model for efficient representation learning." in Neural Networks, vol. 107, pp.12-22, 2018.

- Z. Shboul, L. Vidyaratne, M. Alam, L. Pei, M. Elbakari, and K. M. Iftekharuddin, “Feature-guided Deep Radiomics for Glioblastoma Segmentation and Patient Survival Prediction” in Frontiers in Computational Neuroscience, vol. 13, pp. article 966, 2019.
- M. Alam, MD. Samad, L. Vidyaratne, A. Glandon, and K. M. Iftekharuddin, “Survey on Deep Neural Networks in Speech and Vision Systems” in arXiv preprint arXiv:1908.07656, 2019.

CONFERENCES

- L. Vidyaratne, M. Alam, J. K. Anderson, and K. M. Iftekharuddin, "Improved training of cellular SRN using Unscented Kaiman Filtering for ADP," in Neural Networks (IJCNN), 2014 International Joint Conference on, 2014, pp. 993-1000.
- L. Vidyaratne, A. Glandon, M. Alam and K. M. Iftekharuddin, "Deep recurrent neural network for seizure detection," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 1202-1207.
- L. Vidyaratne, M. Alam, J. K. Anderson and K. M. Iftekharuddin, "Constrained versus unconstrained learning in generalized recurrent network for image processing," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017, pp. 3310-3317.
- L. Vidyaratne, M. Alam, Z. Shboul, and K. M. Iftekharuddin. "Deep learning and texture-based semantic label fusion for brain tumor segmentation." In Medical Imaging 2018: Computer-Aided Diagnosis, vol. 10575, p. 105750D. International Society for Optics and Photonics, 2018.
- Z. A. Shboul, L. Vidyaratne, M. Alam, and K. M. Iftekharuddin, "Glioblastoma and Survival Prediction," Cham, 2018, pp. 358-368: Springer International Publishing.

- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel cellular recurrent deep network for image registration," in SPIE Optical Engineering+ Applications, 2015, pp. 95981B-95981B-10.
- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel hierarchical Cellular Simultaneous Recurrent Neural Network for object detection," in Neural Networks (IJCNN), 2015 International Joint Conference on, 2015, pp. 1-7.
- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Efficient feature extraction with simultaneous recurrent network for metric learning," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, 2016, pp. 1195-1201.
- M. Alam, L. Vidyaratne, T. Wash and K. M. Iftekharuddin, "Deep SRN for Robust Object Recognition: A Case Study with NAO Humanoid Robot", IEEE SoutheastCon, 2016.[1]
- M. Alam, L. Vidyaratne, and K. M. Iftekharuddin, "Novel hierarchical Cellular Simultaneous Recurrent Neural Network for object detection," *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, 2015, pp. 1-7.
- K. M. Iftekharuddin, M. Alam, L. Vidyaratne, "Contemporary deep recurrent learning for recognition," Proc. SPIE 10203, Pattern Recognition and Tracking XXVIII, 1020302 (1 May 2017);
- Glandon, A., S. Ullah, L. Vidyaratne, M. Alam, C. Xin, and K. M. Iftekharuddin. "Prediction of Spatial Spectrum in Cognitive Radio using Cellular Simultaneous Recurrent Networks." In 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, 2018.
- Bussey, Daniel, Alex Glandon, Lasitha Vidyaratne, Mahbubul Alam, and Khan M. Iftekharuddin. "Convolutional neural network transfer learning for robust face recognition in

NAO humanoid robot." In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-7. IEEE, 2017.

- Solopova, A., Carpenter, A., Iftekharuddin, K., Powers, T., Roblin, Y., Tennant, C. and Vidyaratne, L., 2019, June. SRF cavity fault classification using machine learning at CEBAF. In 10th Int. Partile Accelerator Conf.(IPAC'19), Melbourne, Australia, 19-24 May 2019 (pp. 1167-1170). JACOW Publishing, Geneva, Switzerland.