

Brain- and Quantum Inspired Mathematical and Computational Models of Spiking Neural Networks for Deep Learning of Spatio-temporal Data

Helena Bahrami

A thesis submitted to

Auckland University of Technology

in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

30 June 2021

School of Engineering, Computer and Mathematical Sciences

Table of Contents

Table of Contents	I
List of Figures	VII
List of Tables.....	XIII
List of Abbreviations.....	XV
Attestations of Authorship	XVII
Acknowledgement.....	XVIII
Abstract	XIX
Chapter 1 Introduction	1
Abstract	1
1.1 Rationale and Motivation	1
1.2. Aims of Thesis and Research Questions	3
1.2.1 Research Questions	3
1.2.2. Contributions of This Study	5
1.3 Thesis Structure.....	7
Chapter 2 . A Review of Biological Spike-Based Learning in the Brain and the NeuCube Framework	9
Abstract	9
2.1. Introduction	9
2.2 Biological Learning in the Brain	9
2.3 Biological Memory in the Brain.....	14

2.3.1 Types of Memory	14
2.3.2 Brain Regions Involve in Formation of Different Memory Types.....	16
2.4. A Brief Overview of Spiking Neural Networks and the NeuCube Framework.....	19
2.4.1 Input Data Encoding Module	20
2.4.2 3D SNN Reservoir (SNNr) Module	23
2.4.3 Evolving Output Classification Module in NeuCube.....	34
2.5 Chapter Summary.....	39
Chapter 3 . Evolving Predictive Unsupervised Supervised deep learning algorithms for Spike Streams: EPUSSS	42
Abstract	42
3.1 Introduction	43
3.2 Previous Work.....	44
3.2.1 Predictive Models Using Spiking Neural Networks.....	44
3.2.2 Deep Learning in Spiking Neural Networks	46
3.3 The EPUSSS Learning Principles	50
3.4 EPUSSS Novel Characteristics	53
3.4.1 Biological Resemblance Characteristics	53
3.4.2 Computational Characteristics.....	55
3.5 The Proposed EPUSSS Learning Mechanisms.....	56
3.5.1 EPUSSS Local Learning Mechanism.....	56
3.5.2 EPUSSS Global Learning Mechanism	61
3.5.3 Lyapunov Energy Function	65

3.6 The Proposed EPUSSS Algorithm	67
3.6.1 EPUSSS Computational Complexity	70
3.7 Implementation and Experimental Analysis	70
3.7.1 The EPUSSS Framework Implementation	70
3.7.2 Experimental Analysis.....	72
3.8 Chapter Summary.....	89
Chapter 4 Using Chaos Theory to Develop a New Quantum-Inspired Evolutionary Algorithm: CQIEA	92
Abstract	92
4.1 Introduction	93
4.2 Quantum Mechanics Concepts.....	93
4.2.1 Principal Quantum Number.....	95
4.2.2 Azimuthal Quantum Number	95
4.2.3 Magnetic Quantum Number	97
4.2.4 Electron Spin Quantum Number	97
4.2.5 Quantum Wave Function.....	100
4.3 Quantum Computation	101
4.4 Chaos Theory.....	103
4.5 Previous Work.....	105
4.5.1 Quantum-Inspired Evolutionary Algorithms.....	105
4.5.2 Chaotic Optimisation Algorithms.....	107
4.6. The Proposed Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA).....	108

4.6.1 Chirikov Standard Map	113
4.6.2 The Proposed Quantum Chirikov Standard Map	115
4.6.3 The Proposed Chirikov Quantum Gate	116
4.6.4 The Proposed CQIEA algorithm	117
4.7. Benchmark combinatorial optimisation problems	120
4.7.1 Knapsack	120
4.7.2 Travelling Salesman Problem (TSP)	121
4.7.3 Continuous Optimization Benchmarks.....	121
4.8. Implementation and Experimental Analysis	122
4.9 Chapter Summary.....	128
Chapter 5 . A Novel Quantum Inspired Spiking Neural Network: QISNN.....	130
Abstract	130
5.1 Introduction	130
5.2. Previous Work.....	131
5.3. The Proposed Quantum Inspired Spiking Neural Network (QISNN).....	135
5.3.1 Quantum Local Learning	138
5.3.2 Quantum Neurons and Quantum Synapses	139
5.3.3 Quantum Global Learning	141
Chapter 6 . Quantum Inspired Associative Memory for Spiking Neural Networks (QIASM-SNN) on Spatio-temporal Data.....	147
Abstract	147
6.1 Introduction	148

6.2 Previous Work.....	148
6.3 Brain-inspired STAM based on NeuCube	150
6.3.1 Introduction	150
6.3.2 STAM on NeuCube Using EEG Data	151
6.3.3. Challenges For Further Development of STAM-SNN.....	153
6.4 The Proposed Quantum-Inspired STAM for SNN.....	154
6.4.1 The Grover's Algorithm	157
6.4.2 The CQIEA Pattern Retrieving Algorithm	161
6.5 Conclusion.....	163
Chapter 7 . Pre-processing and Feature Extraction from Brain Spatio-temporal EEG Data.	165
Abstract	165
7.1 Introduction	165
7.2 The Dementia EEG Recordings Pre-processing	168
7.2.1 Unifying EEG Channels	169
7.2.2 EEG Feature Extraction using Power Spectral Density	172
7.2.3 Balance Dementia EEG Dataset	177
7.3. Experimental Result of Applying EPUSSS Model to the Dementia Dataset	178
7.4. Using CQIEA to Improve Traditional Fuzzy Rough Set Feature Selection Method to Better Tackle the Curse of Dimensionality Problem.....	190
7.5. Conclusion	197
Chapter 8 . Conclusions and Recommendations for Future Work	199
Abstract	199

8.1. Main Contributions of The Thesis	199
8.1.1 Chapter 3 Contributions and Future Directions.....	201
8.1.2 Chapter 4 Contributions and Future Directions.....	204
8.1.3 Chapter 5 Contributions and Future Directions.....	205
8.1.5 Chapter 6 Contributions and Future Directions.....	207
8.1.6 Chapter 7 Contributions and Future Directions.....	208
8.2 Microtron Neuron Model	209
8.2.2 Other Directions for Further Research	213
References	214
Appendix A	230
Fuzzy Rough Set Feature Selection.....	230

List of Figures

Figure 1-1: The main contributions of my PhD research	6
Figure 1-2: A schematic representation of the thesis structure.....	7
Figure 2-1: The major parts of the human brain (Bhagat, 2007).	11
Figure 2-2: The neuronal synaptic connections (part of the above image is retrieved from https://sbiswasphd.com/alive%3A-the-blog/f/the-broken-mind-mending-the-melancholy) ...	12
Figure 2-3: Human memory types (Mastin, 2010)	15
Figure 2-4: Brain regions for storing memories (retrieved from https://qbi.uq.edu.au/brain-basics/memory/where-are-memories-stored).....	18
Figure 2-5: The overall NeuCube framework architecture (a simplified presentation- only one encoding algorithm is shown out of many possible and only one type of a classifier)	20
Figure 2-6: BSA encoding algorithm result for a sample input signal.	22
Figure 2-7: Pseudocode for BSA algorithm (Schrauwen and Campenhout, 2003).....	23
Figure 2-8: Small-world connectivity creates a local recurrent reservoir network within the NeuCube architecture.....	24
Figure 2-9: The shape of postsynaptic potential (PSP).....	26
Figure 2-10: A schematic diagram of a postsynaptic neuron behaviour which receives spikes from two presynaptic neurons (Gerstner and Kistler, 2002).....	27
Figure 2-11: Four types of neuron response dynamics to a stimulating current (Gerstner and Kistler, 2002)	29
Figure 2-12: LIF neuron model (Gerstner and Kistler, 2002)	30
Figure 2-13: Spike-Timing Dependent Plasticity (STDP) rule.....	34
Figure 2-14: Pseudocode for deSNN algorithm (Kasabov et al., 2013)	39
Figure 2-15: The deSNN learning schema.....	39

Figure 3-1: EPUSSS - Evolving Predictive Unsupervised-Supervised deep learning framework for Spike Streams.....	53
Figure 3-2: EPUSSS local error backpropagation mechanism	57
Figure 3-3: The error backpropagation schema using modified SpikeProp algorithm (Bohte, Kok, & Poutré, 2000).....	60
Figure 3-4: EPUSSS global learning mechanism using R-STDP rule	62
Figure 3-5: The EPUSSS learning algorithm pseudocode.....	69
Figure 3-6: The figure shows twin input neurons that each represent one input/output neuron.	69
Figure 3-7: EPUSSS framework from implementation point of view.....	71
Figure 3-8: Input neurons and reservoir neurons positions according to (a) wireless Emotiv Epoc EEG Neuroheadset (Blanco and Ramirez, 2019) and (b) Talairach map (Lancaster, et al., 2000).	74
Figure 3-9: The steps of mapping brain coordinates, creating the EPUSSS network with excitatory/inhibitory neurons, 4 types of neurotransmitter receivers with different synaptic delay and functionality in propagating information, the hypersynaptic connections, and encoding the input data into spike trains for processing by the EPUSSS two-level hierarchical learning model.	76
Figure 3-10: The downsampled coordinates to improve computational time complexity in the EPUSSS framework.....	77
Figure 3-11: EPUSSS SNN weights distributions: a) initial weights, b) learnt weights	78
Figure 3-12: EPUSSS local learning errors convergence, a) training error, b) test error	78
Figure 3-13: EPUSSS local error for each input in training phase for sample 1	80
Figure 3-14: EPUSSS final local error for sample 1 (all input channels) in the training phase	81

Figure 3-15: EPUSSS local error for each input in testing phase for sample 30.....	83
Figure 3-16: EPUSSS final local error for sample 30 (all input channels) in the testing phase	83
Figure 3-17: EPUSSS SNN connectivity with learnt weight, a) four types of synaptic connections, b) excitatory and inhibitory connections.	84
Figure 3-18: Neuromodulatory (reward/penalty) signal in the EPUSSS global training	84
Figure 3-19: NeuCube SNN learnt weight distribution for Wrist Movement dataset	86
Figure 3-20: NeuCube SNN spiking behaviour for Wrist Movement dataset: a) spiking behaviour of input channels connected to reservoir, b) reservoir neurons regions according to their connectivity to EEG input channels c) excitatory and inhibitory connections, d) EEG channels correlations.....	87
Figure 3-21: LSTM training accuracy and error plot for Wrist Movement dataset.....	88
Figure 4-1: The quantum mechanical model of the atom	94
Figure 4-2: The spin quantum number schematic.....	98
Figure 4-3: Bloch sphere representation of the evolution of the wave function.....	102
Figure 4-4: The structure of QIEA algorithm (Han and Kim,2002).....	111
Figure 4-5: Chirikov standard map with different k values, k=0.5, 0.8, 0.971635, 1.5, 1.8, 2 (from the left most top to the right most bottom).	115
Figure 4-6: Polar plot of the rotation gate and Chirikov map for qubit individuals	117
Figure 4-7: CQIEA algorithms pseudocode for discrete optimisation	118
Figure 4-8: CQIEA algorithms pseudocode for continuous optimisation	119
Figure 4-9: 3D representation of benchmark functions	122
Figure 4-10: The comparison between the CQIEA and QIEA for Knapsack problem with 10 items.....	123

Figure 4-11: The comparison between the CQIEA and QIEA for Knapsack problem with 100 items.....	123
Figure 4-12: The comparison between the CQIEA and QIEA for Knapsack problem with 500 items.....	124
Figure 4-13: The changes of the chaotic rotation gate's angle over time in CQIEA in the unit circle.....	124
Figure 4-14: The changes of the chaotic rotation gate's angle over time in CQIEA.....	125
Figure 4-15: New Zealand cities map for TSP problem	126
Figure 4-16: CQIEA resulted paths for the TSP problem on New Zealand cites map.	126
Figure 4-17: CQIEA result for the TSP problem on New Zealand cities map.	127
Figure 5-1: The QISNN two-level hierarchical learning framework.....	138
Figure 5-2: The role of quantum state probability amplitudes in the learning mechanisms of QISNN	139
Figure 5-3: The local learning quantum neurons and synaptic qubits and the quantum wave function for the probability amplitudes for the entangled neurons.....	140
Figure 5-4: The global learning quantum neurons and synaptic qubits and the quantum wave function for the probability amplitudes for the entangled neurons.....	142
Figure 5-5: The QISNN pseudocode	143
Figure 5-6: The QISNN implementation modular schematic.....	144
Figure 6-1: Training on full data and recall from 100% down to only 5% of the input temporal data results in 100% classification accuracy.....	152
Figure 6-2: Training on full data and recall on only 2% of the temporal input data results in less than 100% accuracy of recall, but still 88% classification accuracy is achieved.	152
Figure 6-3: Training with full data (14 channels, full temporal sequences) and validation with only 13 EEG channels, full temporal sequences): 100% accuracy.....	153

Figure 6-4: The pattern retrieval mechanism in the proposed spiking quantum associative memory	157
Figure 6-5: The mathematical representation for Grover's algorithm.....	159
Figure 6-6: The Quantum circuit for Grover's Algorithm.....	160
Figure 6-7: Grover's algorithm pseudocode	161
Figure 6-8: Schematic representation of CQIEA pattern retrieving algorithm.....	162
Figure 7-1: Data pre-processing phase diagram for the Dementia dataset	169
Figure 7-2: EEG recording scheme with 20 electrodes	170
Figure 7-3: EEG recording scheme with 17 electrodes	170
Figure 7-4: Power Spectral Density Welch algorithm pseudocode	175
Figure 7-5: Power spectrum estimator	175
Figure 7-6: Preprocessing of Mixed Dementia EEG recordings	176
Figure 7-7: Preprocessing of Vascular Dementia EEG recordings	176
Figure 7-8: Preprocessing of Alzheimer's Disease EEG recordings	177
Figure 7-9: EPUSSS SNN initialisation for Dementia	178
Figure 7-10: EPUSSS SNN learnt weights and connectivity for Dementia	179
Figure 7-11: EPUSSS local learning errors convergence, a) training error, b) test error	179
Figure 7-12: EPUSSS SNN weights distributions: a) initial weights, b) learnt weights	180
Figure 7-13: EPUSSS local error for each input in training phase for sample 1	181
Figure 7-14: EPUSSS final local error for sample 1 (all input channels) in the training phase	182
Figure 7-15: EPUSSS local error for each input in testing phase for sample 30.....	183
Figure 7-16: EPUSSS final local error for sample 30 (all input channels) in the testing phase	184
Figure 7-17: Neuromodulatory (reward/penalty) signal in the EPUSSS global training	184

Figure 7-18: Spiking energy of the EPUSSS SNN network during training for Dementia Dataset.....	185
Figure 7-19: NeuCube SNN learnt weight distribution for Dementia.....	187
Figure 7-20: NeuCube SNN spiking behaviour for Dementia: a) spiking behaviour of input channels connected to reservoir, b) reservoir neurons regions according to their connectivity to EEG input channels c) excitatory and inhibitory connections, d) EEG channels correlations.	188
Figure 7-21: LSTM training accuracy and error plot for Dementia	189
Figure 7-22: Fuzzy rough set membership	192
Figure 7-23: Fuzzy rough QUICKREDUCT algorithm	197
Figure 8-1: The classical Microtron (Lidbjork, 1994).....	210
Figure 8-2: The novel Microtron model	212
Figure 8-3: K-Nearest Neighbor classification errors for Dementia dataset	233
Figure 8-4: Decision Tree classification errors for Dementia dataset	234
Figure 8-5: Multilayer perceptron classification errors for Dementia dataset.....	234
Figure 8-6: Random Forest classification errors for Dementia dataset	235

List of Tables

Table 3-1: The EPUSSS model parameter settings	74
Table 3-2: The comparative analysis of Global Learning Classification Results.....	85
Table 3-3: The comparative analysis of Global Learning Classification Results Confusion Matrix.....	86
Table 4-1: Azimuthal quantum number subshell description	96
Table 4-2: The benchmark functions for continuous optimisation.....	121
Table 4-3: The average results of applying CQIEA and QIEA algorithm on Knapsack problem for 20 runs	125
Table 4-4: The average results of applying CQIEA and QIEA algorithm to the continuous benchmark functions for 20 runs	127
Table 7-1: The EEG channels for the two recording schemes.....	171
Table 7-2: The comparative analysis of Global Learning Classification Results for Dementia Dataset.....	186
Table 7-3: The comparative analysis of Global Learning Classification Results Confusion Matrix for Dementia Dataset	187
Table 8-1: The characteristics of the Datasets used for experiments.....	230
Table 8-2: Dementia dataset feature selection results with different Neighborhood radius using fuzzy rough set-based information entropy	231
Table 8-3: The classifiers' accuracy comparison for Dementia dataset using four feature selection algorithms	232
Table 8-4: Dementia dataset feature selection results using SMOTE data sampling method	233
Table 8-5: Feature selection results for UCI datasets	236

List of Abbreviations

AD	Alzheimer Disease
AI	Artificial Intelligence
AMPA	α -amino-3-hydroxy-5-methyl-4-isoxazolepropionic
ANN	Artificial Neural Network
BSA	Ben's Spiker Algorithm
CNN	Convolutional Neural Network
CQIEA	Chaotic Quantum-Inspired Evolutionary Algorithm
DNNs	Deep Neural Networks
deSNN	Dynamic Evolving Spiking Neural Network
EEG	Electroencephalogram
EPUSSS	Evolving Predictive Unsupervised-Supervised deep learning algorithms for Spike Streams
GABAAR	Gamma-Aminobutyric Acid A Receptors
GABABR	Gamma-Aminobutyric Acid B Receptors
LIF	Leaky Integrate and Fire
LTD	long-term depression of synapse
LTP	long-term potentiation
NMDAR	N-methyl-D-aspartate
PSD	Power Spectral Density
pSNM	Probabilistic spiking neuron model
PSP	Postsynaptic Potential
QIASM-SNN	Quantum Inspired Associative Memory for Spiking Neural Network

QISNN	Quantum-Inspired Spiking Neural Network
RO	Rank-Order
RNN	Recurrent Neural Network
R-STDP	Rewarded-Modulated Spike Timing-Dependent Plasticity
SNN	Spiking Neural Network
STDP	Spiking Time Dependent Plasticity
SSTD	Spatio and/or Spatio-temporal Data
SNNr	SNN Reservoir

Attestations of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Acknowledgement

I would like to express my deepest gratitude to my supervisors Prof. Kasabov and Prof. Velasco whose tremendous assistance and dedicated involvement in every step throughout the process of this PhD study helped to accomplish this thesis.

I am grateful for my parents whose constant love and support keep me motivated and confident. My accomplishments and success are because they believed in me and supported me. In addition, I would like to thank my siblings for their sympathetic ear and kind support who were always there for me.

Finally, I owe my deepest gratitude to the AUT administration, especially Joyce D'Mello and Karishma Bhat for their great support throughout the entire thesis process.

Abstract

Spiking neural networks (SNN) represent the third generation of the neural networks. They are inspired by the information processing principles in the human brain. The theory and applications of SNN can further benefit from:

- a) The use of new brain-inspired learning principles for more efficient learning in SNNs.
- b) The use of quantum computation principles for novel learning mechanisms, improvement in the SNNs' performance, and parameter optimisation.
- c) The use of the integration of the above.

To address point (a) outlined above, novel mathematical and computational models of spiking neural networks (SNN) are introduced in this thesis as generic SNN models, to perform both off-line and on-line prediction and classification tasks. These generic models are unified into one single framework called Evolving Predictive Unsupervised-Supervised deep learning algorithms for Spike Streams (EPUSSS) to perform both prediction and classification tasks in a hierarchical fashion.

To address point (b) outlined above, the Quantum-Inspired Evolutionary Algorithm (QIEA) is improved using Chirikov chaotic map and used as the learning rule for SNNs. A search mechanism is proposed to recall, associatively, a pattern stored in the memory. A new parameter optimisation method to improve models' performance is proposed. A novel Quantum Inspired Spiking Neural Network (QISNN) framework is introduced that combines a neuron's macro level structural functionality with its micro level physical and structural functionality to demonstrate a biological behaviour and to reinforce the computational power of SNNs.

To address point (c) outlined above, a novel Quantum Inspired Associative Memory for Spiking Neural Network (QIASM-SNN) is introduced to preserve the spiking activities

produced by the proposed models in SNNs and to recall the stored memories in the presence of partial noisy data.

Several methods for pre-processing and feature extraction of Spatio-temporal EEG data are proposed and illustrated on a real-world problem related to brain neurodegenerative disease. Also, a novel feature extraction method is introduced based on a combination of the proposed Chaotic Quantum-Inspired Evolutionary Algorithm and fuzzy rough set theory for clinical static data related to the EEG recordings. The purpose of this approach is to provide a potential for weighted learning of Spatio-temporal data based on the clinical and demographical observations of the subjects in the proposed methods.

The thesis has mainly a theoretical contribution to the area of SNN and future investigations and applications will follow. To produce precise spiking activities, the idea of a novel neuron model inspired by the nuclear physic concept of Microtron Accelerator is presented as a future study to add a self-adaptable delay mechanism to the SNNs. As a future study, several pre-processing techniques are also suggested for real-world datasets.

Four papers are under preparation to submit in order to publish the main contributions elaborated in this thesis.

Chapter 1 Introduction

Abstract

This chapter presents the motivation for the development of brain- and quantum-inspired methods of SNN to further use brain information processing principles as an inspiration to improve the functionality of SNN. Both brain- and quantum information principles are intermixed in the human brain, and this is also a motivation for this research.

1.1 Rationale and Motivation

With the technological advancements in almost every field of human life such as science, business, and healthcare, a large amount of data is available to be collected from different sources. This data-driven era requires new techniques to analyse this immense data repository for various purposes.

Machine learning techniques have been proven to be effective methods for analysing complex, nonlinear data and extracting patterns from them to classify or predict future events based on the correlation of historically learned data. Amongst these techniques, SNNs are powerful approaches to learn Spatio and/or Spectro-Temporal Data (SSTD) since they are using trains of spikes (binary temporal events) transmitted among spatially located synapses and neurons. The SNNs' inherent fast computational power, event-based learning approach, energy efficiency, and hardware friendly characteristics make them a proper choice for emerging hardware/software computational technologies under the name of neuromorphic computational platforms.

SNNs are composed of neurons that communicate by sequences (trains) of spikes (Ponulak and Kasinski, 2011; Maas1997). From 1952, when Hodgkin and Huxley introduced the first spiking neuron model (Hodgkin and Huxley, 1952), to the present, a lot of effort has been made to

develop and improve SNN and to solve a wide range of applications in the field of engineering, economics, health, and medicine.

A spiking neural model is based on the principle of emitting an output spike signal in the presence of stimuli from one or many spike inputs. The probability of emitting a spike is increased by excitatory inputs and decreased by inhibitory inputs. To simulate a spiking neuron, several models have been proposed, among them Leaky-Integrate-and-Fire (LIF), Spike Response Model (SRM), and Izhikevich neuron model are the most popular ones. Spiking neuronal models are used to build SNN. The topology of the SNN models can be categorized in three classes: feedforward SNN; recurrent SNN, and hybrid SNN, with the latter one benefitting from both feedforward and recurrent topology advantages (Gerstner and Kistler, 2002).

Many coding strategies have been proposed to convert SSTD to train of spikes such as time to first spike, rank order coding, latency code, resonant burst model, coding by synchrony, and phase coding. (Gerstner and Kistler, 2002). Some of the popular spike encoding algorithms are Ben's Spiker Algorithm (BSA), Threshold-based Algorithms, Step-Forward Spike Encoding Algorithm (SF) and Moving-Window Spike Encoding Algorithm (MW). For a review one can refer to (Kasabov et al, 2015).

So far, many supervised and unsupervised learning algorithms are proposed to train a SNN model. Spike Timing Dependent Plasticity (STDP) is an unsupervised learning algorithm. Remote Supervised Method (ReSuMe) and Spike Pattern Association Neuron (SPAN) are supervised learning algorithms. STDP which is a variant of Hebbian Learning is an unsupervised learning algorithm that is sensitive not only to the spatial but also to the temporal correlation between the spikes at pre- and postsynaptic neurons (Song, Miller, and Abbott, 2000; Buchs and Senn, 2002; Kistler, 2002; Xie and Seung, 2004; Caporale and Dan, 2008).

ReSuMe and SPAN are Hebbian based supervised learning algorithms which are based on an error measured in a time window (Mohammed et al., 2013).

Using the above encoding and learning algorithms, SNN architectures and frameworks have been created, such as the evolving Spatio-temporal data machines (eSTDM) (Kasabov et al., 2015). The eSTDM has the potential to learn incrementally from data streams, including “on the fly” new input variables, output class labels or regression outputs, continuously adapting its structure and functionality, be visualised, and interpreted for new knowledge discovery and for a better understanding of the data and the processes that generated it. The eSTDM can be used for early event prediction due to the SNN’s ability to spike early before the whole input vectors (they were trained on) are presented. The eSTDM uses a framework called NeuCube (Kasabov, 2014), a 3D evolving Neurogenetic Brain Cube of spiking neurons that is an approximate map of the structural and functional areas of interest of an animal or human’s brain.

All the above techniques represent the foundation of SNN. Based on this foundation, new methods need to be developed to improve SNN performance. In this thesis we have sought further inspiration from the brain and quantum computation to achieve this goal.

1.2. Aims of Thesis and Research Questions

This section provides the aims of this research and outlines the contributions that are made to achieve them.

1.2.1 Research Questions

In this thesis, the following research questions (RQ) are addressed:

RQ1 “How can unsupervised learning in SNN be integrated with local supervised and global supervised deep learning methods for predictive modelling and classification of spatio-temporal data?”

RQ2 “How can different types of neurotransmitter receivers be incorporated into SNN’s model synaptic connections to balance the network’s spiking activity and help in the deep learning of more precise patterns?”

RQ3 “How can a novel energy function be introduced to facilitate life-long learning in SNNs?”

RQ4 “How can a quantum-inspired evolutionary algorithm (QIEA) be further developed using a quantum chaotic system (to become CQIEA) and applied for optimised learning of noisy data, searching a pattern in stored memory, and parameter optimisation in SNN?”

RQ5 “How can the biologically probabilistic nature of spiking neurons and quantum mechanics concepts be integrated to reinforce all levels of computation in SNN, including quantum neurons, quantum synapses and chaotic quantum unsupervised-supervised learning?”

RQ6 “How can spiking patterns be stored and recalled in an SNN for recognition and prediction tasks in the presence of noisy or partially incomplete data by introducing a novel quantum associative memory (QAM) for Spatio-temporal data?”

RQ7 “How can the proposed QAM for storing and recalling spike patterns be improved in terms of time and space efficiency using the proposed CQIEA algorithm?”

RQ8 “How to pre-process spatio-temporal data and to extract useful features that can help improve the applicability of neuromorphic computation for a real-world application”

RQ9 “How can the mechanism of producing spiking patterns be improved in a self-adaptive manner according to the SNN by introducing a novel neuron model using nuclear physics concepts?”

1.2.2. Contributions of This Study

In this thesis, I have suggested a novel Evolving Predictive Unsupervised-Supervised deep learning algorithms for Spike Streams (EPUSSS) to predict and classify the events in a unified framework. In the EPUSSS framework, both excitatory and inhibitory neurons are embedded in four different types of neurotransmitter receivers with different delays and functionality. In addition, an energy function and a neuron pruning mechanism are integrated into the EPUSSS model to ensure life-long learning behaviour.

In the thesis, I have introduced an improved Quantum-Inspired Evolutionary Algorithm using Chirikov chaotic map (CQIEA) to be used as a learning rule for SNN, a search mechanism for associative memory, and a parameter optimisation technique for enhancing the proposed models' performance.

I have proposed a novel Quantum-Inspired Spiking Neural Network (QISNN) to demonstrate the biological neuron's probabilistic behaviour and to reinforce the computational power of the spiking neural network. The QISNN model follows the two-level hierarchical learning structure of EPUSSS and consists of quantum neurons, quantum synapses, and quantum learning rules to perform both prediction and classification tasks in a unified framework.

Another contribution of this thesis is introducing a Quantum-Inspired Associative Memory for Spiking Neural Network (QIASM-SNN) to store Spatio-temporal patterns and to recall them in the presence of noisy information.

Many SNN models work on raw, unprocessed Spatio-temporal data, but in some cases pre-processing and feature extraction from the data makes the SNN model more efficient, including the proposed models in this thesis. Some of the proposed methods in this thesis, such as CQIEA are applied to reduce the dimensionality of the data and the feature space. The real-world problem data is collected from existing Alzheimer's disease and dementia data.

As a future direction, I have proposed a new neuron model based on the Microtron accelerator to enhance the proposed spiking model's performance by controlling spike firing rates and delays. Also, a quantum-inspired fuzzy rough set theory is suggested as a future work for static and Spatio-temporal feature selection.

The outcome of my thesis can act as a main source of reference for Artificial Intelligence and Machine Learning scientists, as well as neuroscientists and biologists by providing them a basis for developing a diagnosis system to predict brain cognitive activities, neurodegenerative disease progression and in neurorehabilitation. Figure 1-1 illustrates the main contributions of my PhD research.

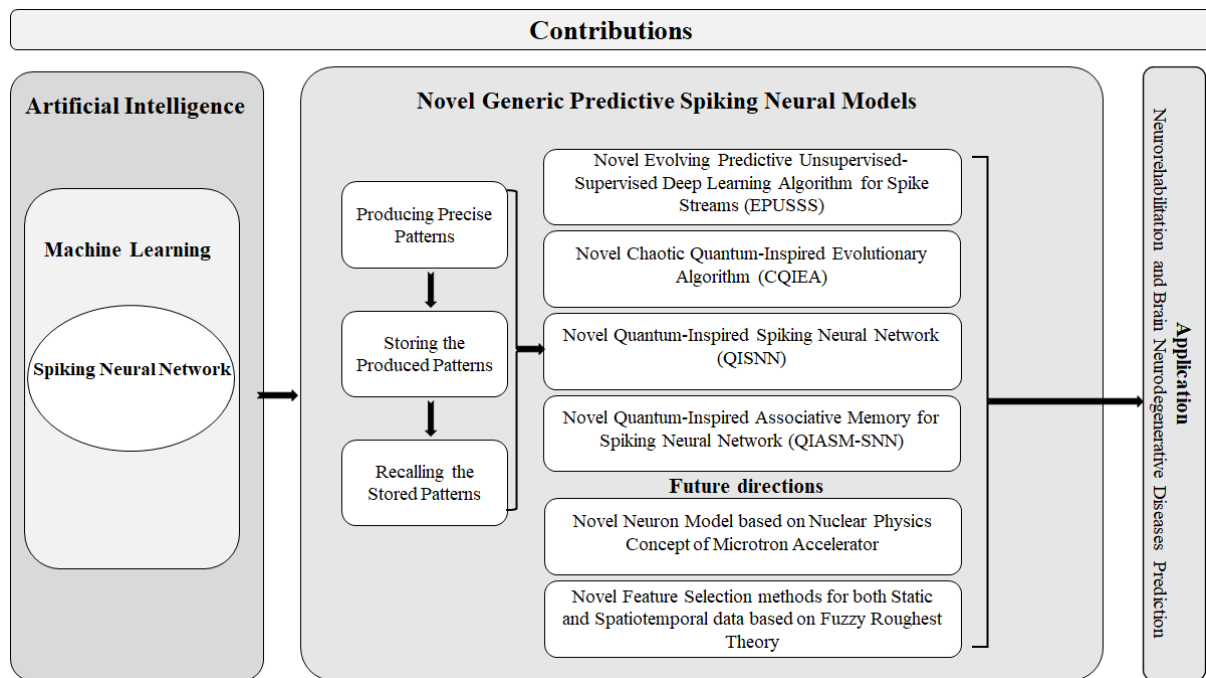


Figure 1-1: The main contributions of my PhD research

1.3 Thesis Structure

In this thesis, all the developed methods are inspired by the NeuCube brain-like framework introduced in (Kasabov, 2014). In the following section, the thesis structure is provided. The schematic representation of the thesis structure can be seen in Figure 1-2.

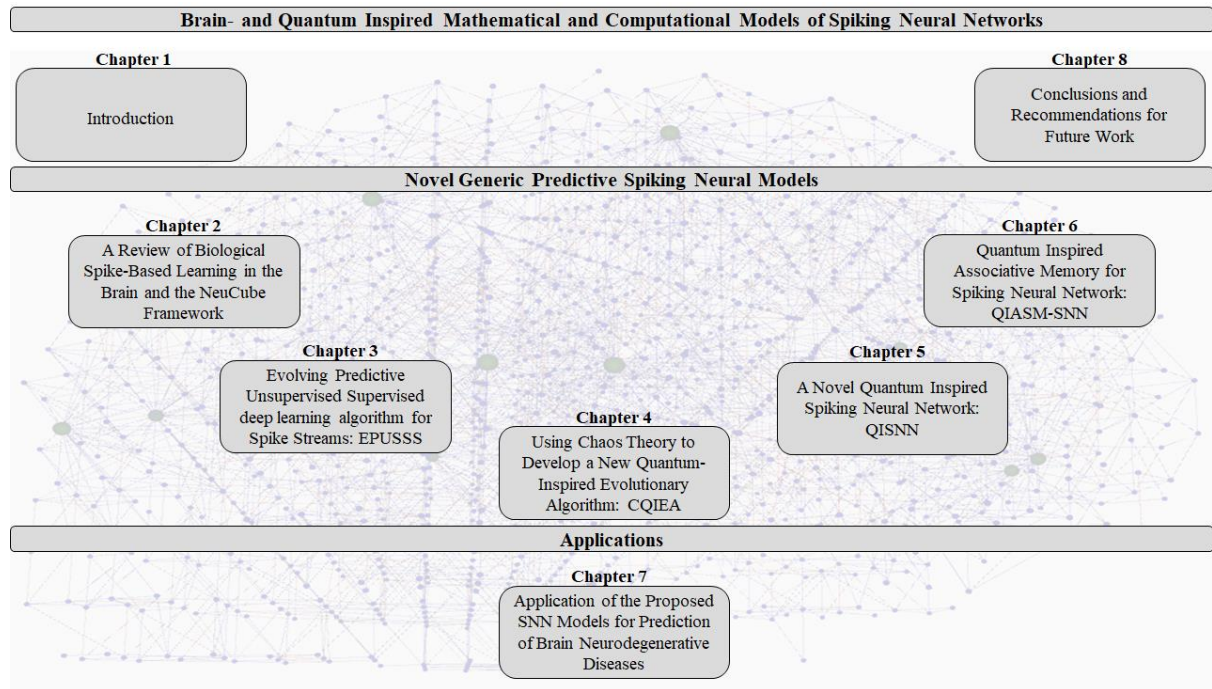


Figure 1-2: A schematic representation of the thesis structure

Chapter 1 states the PhD study's motivations, goals, and research questions, and outlines methods to address these questions.

Chapter 2 provides a brief review of biological spike-based learning in the brain and the NeuCube computational framework which are the biological inspiration and the basic network structure for the spiking neural models proposed in this thesis, respectively.

Chapter 3 introduces a novel Evolving Predictive Unsupervised-Supervised deep learning algorithms for Spike Streams to create an integrated computational model of continuous deep

learning of streaming data or alternatively, of temporal (Spatio-temporal) data of fixed time window, encoded as spike streams.

Chapter 4 introduces a novel Chaotic Quantum-Inspired Evolutionary Algorithm (QIEA) by suggesting a chaotic quantum gate that can be used as a learning algorithm, a search method for retrieving a stored pattern from a memory, and a parameter optimisation technique to improve models' performance.

Chapter 5 presents a novel Quantum Inspired Spiking Neural Network that combines neuron's macro level structural with its micro level physical and structural functionality to demonstrate the biological behaviour and reinforce the computational power of spiking neural networks. The QISNN follows the EPUSSS framework principles to perform both prediction and classification tasks in a two-level hierarchical fashion.

Chapter 6 proposes a novel Quantum-Inspired Associative Memory for Spiking Neural Networks to recall produced patterns by the SNN in the presence of partially noisy input data, achieving better computational efficiency and accuracy.

Chapter 7 investigates the proposed models' prediction, classification and pattern recalling power by applying them to the real-world brain neurodegenerative disease datasets. To increase the accuracy of learning from the data, several pre-processing methods are introduced including cleaning the dataset, oversampling techniques for balancing the datasets, and feature extraction using the Power Spectral Density (PSD) method for electroencephalogram (EEG) data.

Chapter 8 summaries the thesis achievements, key findings, and contributions. Future directions are also suggested. In the future work, a novel neuron model inspired by the nuclear physics concept of Microtron Accelerator is introduced with the aim of adding a self-adaptable delay mechanism to SNN. Also, a quantum-inspired fuzzy rough set theory is suggested for feature extraction.

Chapter 2 . A Review of Biological Spike-Based Learning in the Brain and the NeuCube Framework

Abstract

This chapter presents some principles of information processing in the human brain which are used as source of inspiration for the brain inspired SNN models developed in this thesis. It also reveals the functionality of NeuCube which is a brain inspired SNN architecture, used as a basic structure for the novel methods and models proposed in this thesis.

2.1. Introduction

The first part of this chapter provides some biological perspective on learning and memory in the brain which helped to form a strong foundation for the proposed novel brain-like models in this thesis. In the second part of this chapter, the NeuCube framework and its modules including encoding, connectivity, unsupervised, and supervised learning are explained.

2.2 Biological Learning in the Brain

The human brain is the most complex biological system in nature. Understanding the brain functionality and behaviour can be used for building machines that can perform efficient computation, recognition and learning tasks as well as having a life-long memory storage to store and retrieve memories.

The human brain is the central organ in the nerves system that controls all functions of the body and governs emotions, cognitions, memory, motor movements etc.

The brain is composed of three main parts: cerebrum, cerebellum, and brainstem (see Figure 2-1).

The **cerebrum** is the largest part of the brain which is divided into two hemispheres (left and right). The cerebrum is responsible for controlling high level functions such as vision, hearing, speech, reasoning, emotions, **learning**, and fine control of movement.

The **cerebellum** is located at the back of the brain, underlying the occipital and temporal lobes of the cerebral cortex. Despite the cerebellum small volume (approximately 10% of the brain's volume) it contains over 50% of the total number of neurons in the brain. The cerebellum is involved in the functions like maintenance of balance and posture, coordination of voluntary movements, motor learning, and certain cognitive functions.

The **brainstem** is the most inferior portion of the brain, adjoining and structurally continuous with the brain and spinal cord. The brainstem functions include regulation of heart rate, breathing, sleeping, and eating. It also conducts all information relayed from the body to the cerebrum and cerebellum and vice versa.

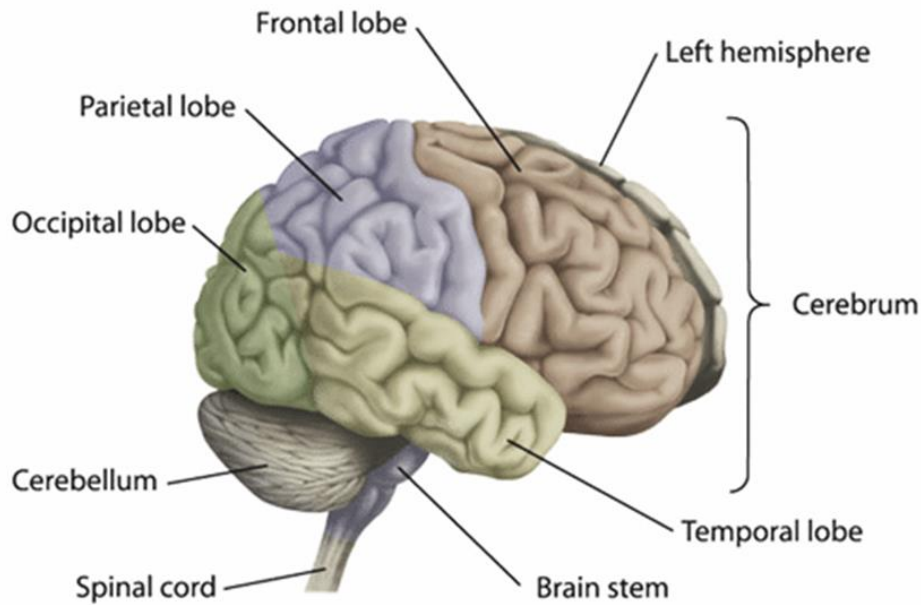


Figure 2-1: The major parts of the human brain (Bhagat, 2007).

The brain contains approximately 100 billion nerve cells (neurons). Neurons are the elementary processing units in the nerves system with different size and shapes. However, all neurons consist of a cell body, dendrites, and an axon. Neurons are connected to each other by their dendrites and axons creating a complex lattice. The connection between two neurons is called synapse. The neuron which sends information through the synaptic connections to other neurons is called presynaptic neuron and the neuron that receives information from a synaptic connection is called postsynaptic neuron. The dendrites and axons play the role of receiving and sending information respectively and cell body acts as the information processor. Once the total input to the neuron's cell body exceeds a certain threshold an electrical pulse called action potentials is generated and passes through axons to other neurons.

In the brain, the information about a stimulus (e.g., light, sound, taste, smell, and touch) is encoded in a temporal pattern of action potentials and transmitted to the neurons in the brain through the synaptic connections.

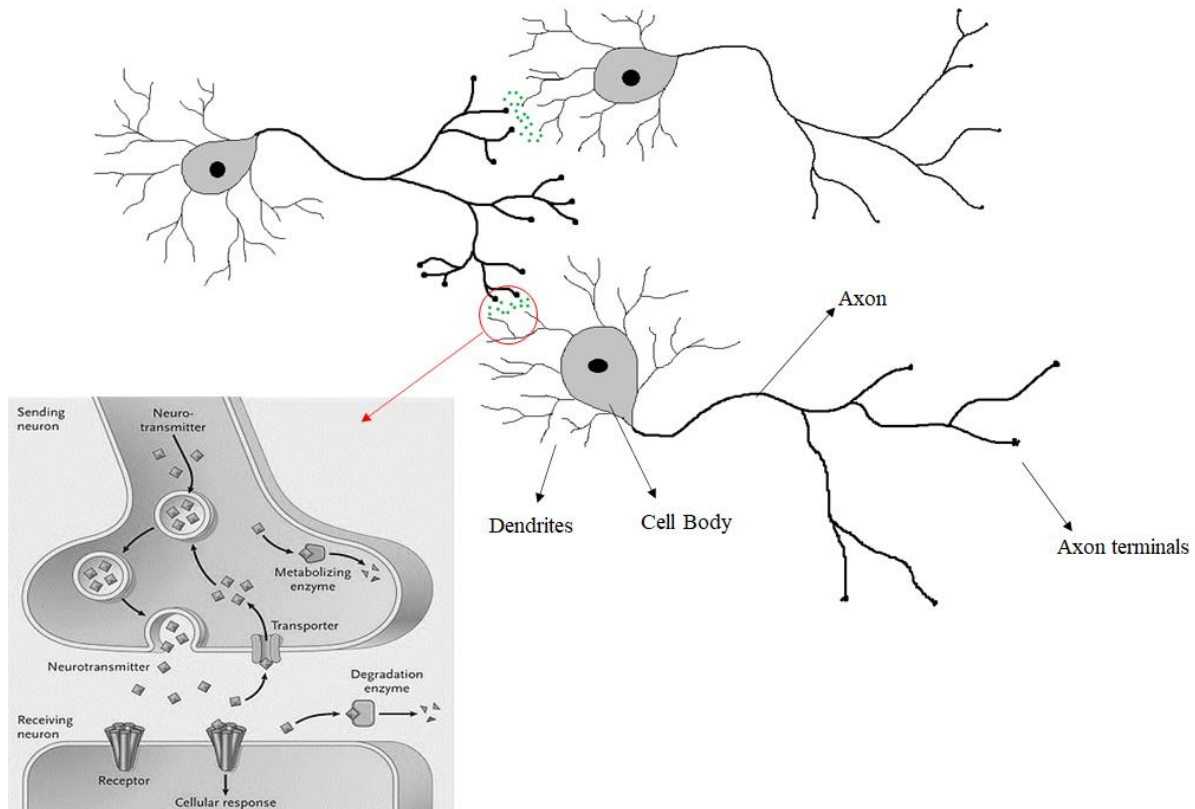


Figure 2-2: The neuronal synaptic connections (part of the above image is retrieved from <https://sbiswasphd.com/alive%3A-the-blog/f/the-broken-mind-mending-the-melancholy>)

There are two types of synapses in the brain: electrical and chemical synapses. Electrical synapses send electrical signals from one neuron to another using the direct passage of ions. While in chemical synapses, due to an action potential in the presynaptic neuron, a chemical messenger called neurotransmitter releases, diffuses through the synapse, and binds to receptors on the postsynaptic neuron. This binding between neurotransmitters and receptors generates an electrical signal in the postsynaptic neuron. Electrical synapses pass the signal very quickly, while chemical synapses transmit the signal from one neuron to another more slowly. However, chemical synapses can integrate information from multiple presynaptic neurons and determine whether the postsynaptic neuron continue to propagate the signal or not. Most of the synapses in the brain are chemical. Neurons can synthesise and release different types of neurotransmitters to the chemical synapses. There are two major types of neurotransmitters: small-molecule transmitter and large-molecule transmitter. Small-molecule

transmitters (neurotransmitters) are fast-acting and produce a short-term response to target cell. While large-molecule (neuropeptides) are slow-onset and produce long-lasting modulation of synaptic transmission to a group of neighbouring cells. Neurotransmitters have two types in terms of functionality: excitatory and inhibitory neurotransmitters. Excitatory neurotransmitters increase the likelihood of a postsynaptic action potential. In contrast, inhibitory neurotransmitters decrease the likelihood of a postsynaptic action potential. One of the most abundant excitatory neurotransmitters in the vertebrate nervous system is glutamate. There are two receptors in the postsynaptic cell to bind glutamate and activated an action potential: the α -amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor (known as AMPA receptor) and the N-methyl-D-aspartate receptor (known as the NMDA receptor). NMDA type receptors typically possess slower kinetics than AMPA type receptors.

The gamma-Aminobutyric acid, or γ -aminobutyric acid or GABA is the most common inhibitory neurotransmitter in the central nervous systems and have two types of receptors: GABAA receptors and GABAB receptors. GABAA is involved in fast synaptic inhibition while GABAB is considered in slow synaptic inhibition.

Glutamate neurotransmitters are involved in cognitive functions such as learning and memory in the brain while GABA neurotransmitters reduce the neuronal excitability throughout the nervous system (Pickel and Segal, 2014; Clark and Pazdernik, 2013; Watson and Greger, 2017; Smith, 2013).

According to the neurological evidence, the brain physically changes throughout the lifetime which is the basis of adaptation, learning, and memory. Although the large-scale changes such as the growth of new neurons or dendrites can occur very rarely, minute-to-minute changes are continuously happening between neuronal connections. These changes in neuronal connections

which are the primary mechanism for learning and memory are known as “synaptic plasticity” (Ramirez & Arbuckle, 2016).

Although most neurons and glial cell are formed by birth, the brain continues to develop by refining synapses till early adulthood. The synaptic refinement which includes dendrite development, synaptic pruning, and changes in neurotransmitter system allows brain to be modulated for learning and memory (Stephan et al. 2012).

2.3 Biological Memory in the Brain

Memory is referred to as the brain ability to recall experiences or information that is encountered or learnt previously. As developed in this thesis, the proposed methods of SNN are also memory based, here I give a brief review of the types of brain memories.

According to Squire (2009), the study of learning and memory in the human brain has been divided into three main areas: philosophy, psychology, and biology. Based on the biological study of memory three main principles has been established to guide experimental work: 1) memory is a distinct cerebral function which is separable from other cognitive abilities. 2) The medial temporal lobe is not needed for immediate memory. 3) The medial temporal lobe is not the ultimate repository of memory. It can be deduced from these three principles that memory is not a unitary part of the mind but is composed of multiple systems that have different operating principles and different neuroanatomy (Squire, 2004).

2.3.1 Types of Memory

There are three main types of memory: sensory memory, short-term or working memory and long-term memory which function in different ways (see Figure 2-3).

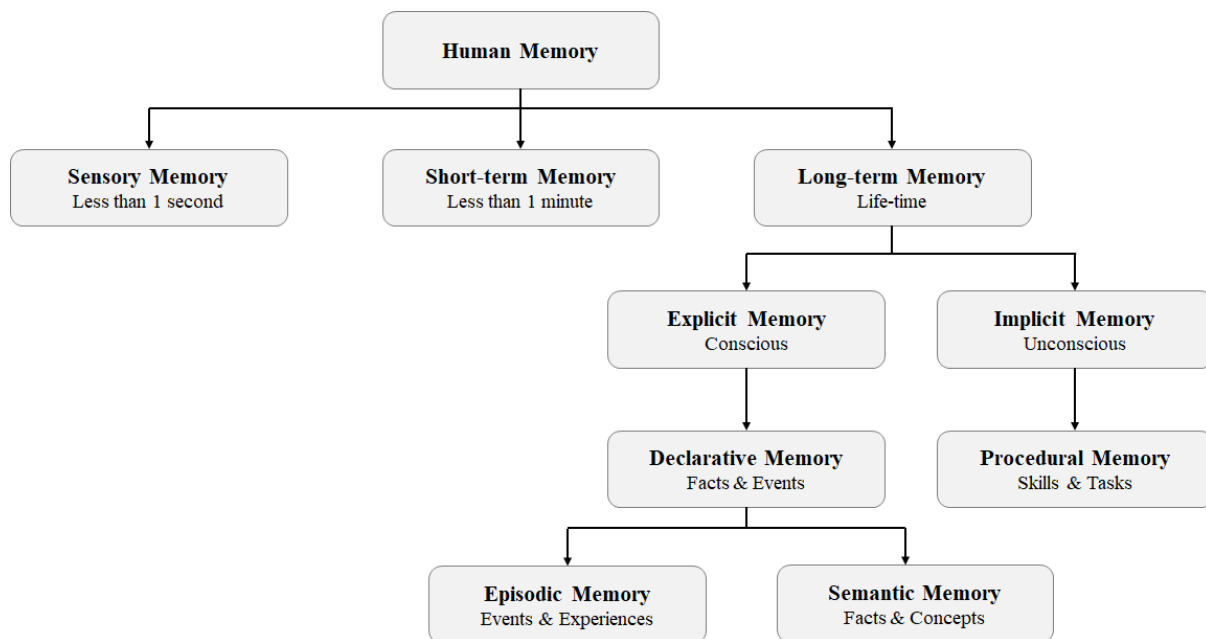


Figure 2-3: Human memory types (Mastin, 2010)

2.3.1.1 Sensory Memory

Sensory memory is the shortest form of memory (less than half a second) which acts as a buffer for stimuli received through the five senses sight, hearing, smell, taste, and touch, which are retained accurately, but very briefly.

2.3.1.2 Short-term Memory

Short-term memory, often interchanged with the term "working memory," is a temporary recall of the information which last less than a minute. It can be considered as the ability to remember and process information in a small chunk for a short period of time. In fact, short-term memory has a low capacity and once the information is processed either is quickly dismissed or transfer to the long-term memory by consciously repeat to retain the information. The short-term memory is supported by transient patterns of neuronal communication in the regions of the frontal, prefrontal, and parietal lobes of the brain.

2.3.1.3 Long-term Memory

Long-term memory is the brain mechanism for storing, managing, and recalling information over a long period of time. The process of establishing long-term memory requires physical changes in the neuron's structure which is called long-term potentiation (LTP). LTP occurs when neuron's synaptic connections strengthening persistently upon recent patterns of activity, this phenomenon is called synaptic plasticity. In other words, making meaningful association amongst information that process in the short-term memory can establish a long-term memory semantically which is widely spread throughout the interconnected brain regions. One of the brain regions that has an important role in transition of information from short-term memory to long-term memory is hippocampus. Although the hippocampus is not itself used to store information, it is an essential part of forming and indexing memories for later access.

There are different types of long-term memories which sometimes require conscious efforts to recall a piece of information which is called declarative memory and is mostly about facts and events. Other times unconsciously recollect stored information without actively thinking about it which is called procedural memory abilities such as skill learning and habit learning.

2.3.2 Brain Regions Involve in Formation of Different Memory Types

Although the human brain is highly interconnected, there are six major regions identified for different types of memory to store: the hippocampus, the neocortex and the amygdala which are responsible for long-term memories, the basal ganglia and cerebellum which are regions to store sensory memories, and prefrontal cortex that is known for storing short-term memories (ScienceDaily, 2021). These regions are shown in Figure 2-4 and their functionality is briefly described in the following.

2.3.2.1 Hippocampus

The hippocampus is located in the brain's temporal lobe and contains a group of neurons that involve forming and indexing episodic memories for later access. Episodic memories are autobiographical memories from specific events. As it was mentioned before, although the hippocampus is an essential part of the brain for the transition from short-term memory to long-term memory, it is not the place for permanent memory storage, and it is not needed for motor memories.

2.3.2.2 Neocortex

The neocortex or cerebral cortex is a sheet of neural tissue that forms the outside surface of the brain. The neocortex is 2 – 4 mm thick and contains six distinct but interconnected layers. Its groovy and folded pattern allows covering a large surface area (typically almost 0.12 m^2) to fit within the confines of the skull. Hereupon, about 90% of all the brain's neurons are located in the neocortex, mainly in its surface layer which is known as the “grey matter”. The inner surface of the neocortex or the “white matter” consists of over 170,000 km myelinated axons to support the active nerve cells. The sensory perception, generation of motor commands, spatial reasoning and language are part of the neocortex functionality. Information that forms temporally in the hippocampus can be transferred to the neocortex and form long-term memories.

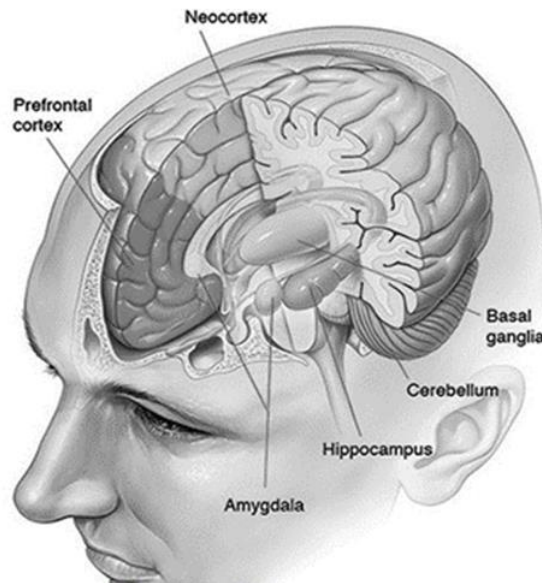


Figure 2-4: Brain regions for storing memories (retrieved from <https://qbi.uq.edu.au/brain-basics/memory/where-are-memories-stored>)

2.3.2.3 Amygdala

The amygdala is an almond-shaped structure in the brain's temporal lobe which modifies the strength and emotional content of memories. Therefore, the amygdala role is important since the strong emotional memories such as those associated with shame, joy, love, or grief are difficult to forget. This fact shows that there are interactions between the amygdala, hippocampus and neocortex which forms 'stability' of a memory. Another key function of the amygdala is forming new memories specifically related to fear. Fearful memories can be formed after only a few repetitions.

2.3.2.4 Cerebellum

The cerebellum located at the rear base of the brain which plays an important role in fine motor control is also involved in some cognitive functions such as attention, language, emotional functions by regulating fear and pleasure responses in the processing of procedural memories.

2.3.2.5 Prefrontal Cortex

The prefrontal cortex is the part of the neocortex that is placed at the very front of the brain. The prefrontal cortex is involved in many complex cognitive functions that require to hold information in the short-term memory. There is also a functional separation between the left and right sides of the prefrontal cortex. The left side is more involved in verbal working memory while the right side is more active in spatial working memory, such as remembering where an event occurred.

The biological information related to the brain structure, nerves system, and its learning mechanisms which is reviewed in the above sections has helped to build the foundation of the brain-like computational models proposed in this thesis.

In the following sections, the SNN structure and the NeuCube framework is reviewed since they provide the structural and computational basis for the proposed models in this thesis.

2.4. A Brief Overview of Spiking Neural Networks and the NeuCube Framework

Since the proposed models in this thesis follow the NeuCube framework principles, a brief overview of this framework is provided below.

NeuCube is a generic unified framework based on spiking neural network to learn, visualise, and interpret Spatio and/or Spatio-temporal Data (SSTD), introduced by Professor Nikola Kasabov in 2014 (see Figure 2-5). The NeuCube framework is an approximated map of structural and functional areas of interest of the brain with a 3D recurrent structure of spiking neurons connected to each other according to small world connectivity (Hu et al., 2014; Bassett & Bullmore, 2017). NeuCube framework consists of three main modules: input data encoding

module, 3D SNN reservoir module, and evolving output classification module which are briefly overviewed in the following sections.

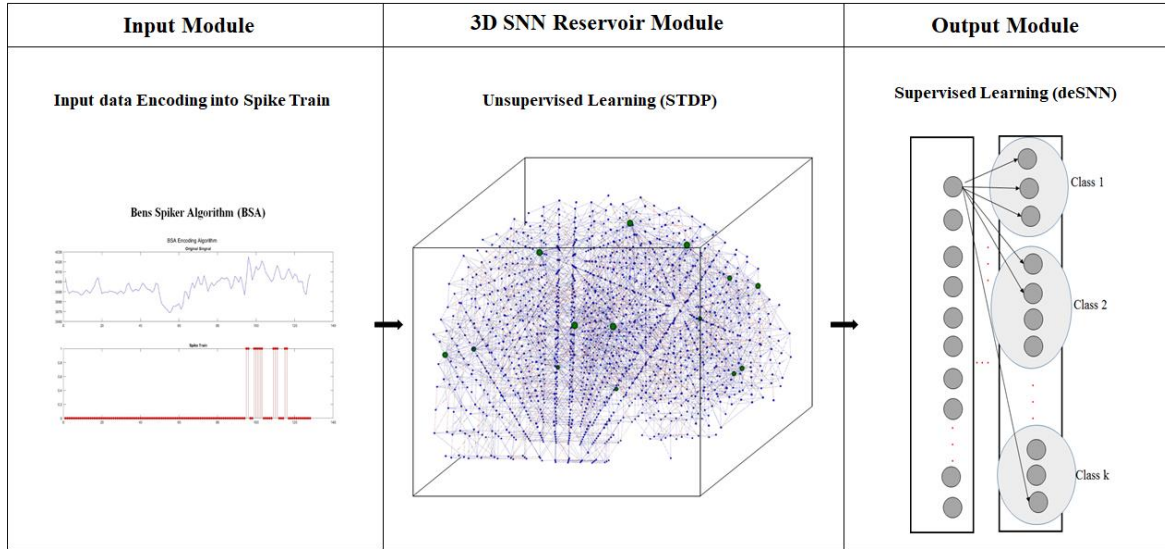


Figure 2-5: The overall NeuCube framework architecture (a simplified presentation- only one encoding algorithm is shown out of many possible and only one type of a classifier)

2.4.1 Input Data Encoding Module

As it was mentioned before in chapter 1, most of the problems in nature require spatial and/or Spatio-temporal data (SSTD) that include measuring spatial or/and spectral variables over time. SNNs have the potential to learn SSTD using trains of spikes (binary temporal events) transmitted among spatially located synapses and neurons. Since these data are intrinsically continuous valued, the encoding strategies needed to be applied to convert SSTD to train of spikes. NeuCube uses a variant of spike encoding strategies such as Ben's Spiker Algorithm (BSA), Temporal Contrast (Threshold-based representation—TBR), Step-Forward Spike Encoding Algorithm (SF), Moving-Window Spike Encoding Algorithm (MW) (Schrauwen & Campenhout, 2003; Dhoble, Nuntalid, Indiveri, & Kasabov, 2012; Petro, Kasabov, & Kiss, 2020). In this thesis, the BSA encoding algorithm (Schrauwen and Campenhout, 2003) was preferred over other methods because it is less sensitive to changes in the filter and threshold and as a result provides a more accurate spike train even for bipolar cases. BSA uses the finite

impulse response (FIR) filter for converting analog waveforms to spike trains. To convert the continuous-valued data to spike train using BSA, for every instant of the input analog signal s at in time τ two error metrics are calculated as follow:

$$e_1 = \sum_{k=0}^M abs(s(k + \tau) - h(k)) \quad (2-1)$$

and

$$e_2 = \sum_{k=0}^M abs(s(k + \tau)) \quad (2-2)$$

Where, M is the number of filter taps, s is input signal, and $h(k)$ is the impulse response of a FIR filter at tap k . Error e_1 is yielded from subtracting the filter coefficients from the subsequent signal values at each time point and error e_2 is the signal value without changing at each time point. If $e_1 \leq e_2 - threshold$ then a spike is generated, and the filter (coefficients) is subtracted from the input signal. The optimal value for the threshold is 0.9550 according to (Schrauwen and Campenhout, 2003). Figure 2-6 shows a sample signal (one channel of an EEG signal) converted to spike train using BSA. The pseudocode of the BSA algorithm can be seen in Figure 2-7.

Bens Spiker Algorithm (BSA)

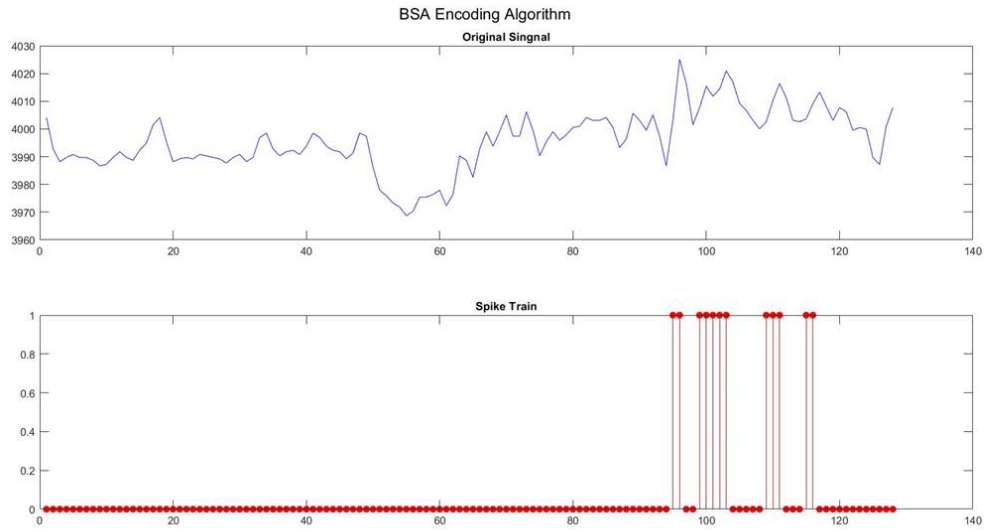


Figure 2-6: BSA encoding algorithm result for a sample input signal.

The BSA algorithm pseudocode can be seen in Figure 2-7.

Bens Spiker Algorithm (BSA)

```

1:   for  $i = 1$  to  $\text{size}(s)$ 
2:        $e_1 = 0$  ,  $e_2 = 0$ 
3:       for  $j = 1$  to  $\text{size}(h)$ 
4:           if  $i+j-1 \leq \text{size}(s)$ 
5:                $e_1 += \text{abs}(s(i+j-1) - h(j))$ 
6:                $e_2 += \text{abs}(s(i+j-1))$ ;
7:           end if
8:       end for
9:       if  $e_1 \leq (e_2 - \text{threshold})$ 
10:          output( $i$ )=1
11:          for  $j = 1$  to  $\text{size}(h)$ 
12:              if  $i+j-1 \leq \text{size}(s)$ 
13:                   $s(i+j-1) = h(j)$ 

```

```

14:         end if
15:     end for
16:     else
17:         output(i)=0
18:     end if
19: end for

```

Figure 2-7: Pseudocode for BSA algorithm (Schrauwen and Campenhout, 2003)

2.4.2 3D SNN Reservoir (SNNr) Module

The SNNr module in NeuCube has a 3D structure inspired by the structural or functional connections between different areas of the human brain. SNNr can provide an approximate map of spatially located areas of the brain according to standard brain atlas coordinate systems such as Talairach Atlas (Talairach and Tournoux, 1988). Although NeuCube is inspired by human brain structural and functional neuronal connectivity, it can approximate a map of spatial information of different types of data such as environmental (geographical) data. It has a default map for data with no specific structures. In this section, the SNNr network connectivity, its neuron models, and the unsupervised learning algorithm that governs the changes of the synaptic connection strength are explained.

2.4.2.1 Small-world Connectivity

Establishing an initial topological (structural) connectivity amongst network's neurons as well as assigning initial weights to these synaptic (neuronal) connections is important for learning the Spatio-temporal properties of the data in the model.

One of the successful approaches to creating connectivity that resembles the biological connectivity structure of the brain's neurons (Bullmore and Sporns, 2009) is small-world connectivity proposed by Watts and Strogatz in 1998.

Small-world connectivity is mathematical graph connectivity in which the probability of neurons connected to each other is proportional to their distance. In a small-world network, the typical distance L between two randomly chosen neurons increases proportionally to the logarithm of the number of neurons N in the network.

$$L \propto \log N \quad (2-3)$$



Figure 2-8: Small-world connectivity creates a local recurrent reservoir network within the NeuCube architecture. After establishing the neuronal synaptic connections, the initial weights' value can be a random uniform distribution between (1,0) or all weights can be simply initialised to a small homogeneous value (see Figure 2-8).

2.4.2.2 Spiking Neuron Models

In biological neurons, an input electrical pulse (spike) can cause the potential difference in the postsynaptic neuron's cell interior and its surroundings. This potential difference is called the membrane potential $u(t)$. If the change in the neuron's membrane potential is positive, the presynaptic connection to the neuron is called excitatory. While the negative change in the neuron's membrane potential is said to be caused by an inhibitory presynaptic connection. When there is no input spike, the neuron is at resting state with a resting membrane potential u_{rest} . When the neuron is at resting state, the membrane potential has a strong negative polarisation of about -65 mV. Once an input spike arrives at the neuron, the membrane potential changes (either positively or negatively) and decays back to constant membrane potential. Input spike from an excitatory synapse decreases the negative polarisation which is called depolarization. On the other hand, input from inhibitory synapse increases the negative polarisation of the membrane potential further and is called hyperpolarisation.

When the membrane potential reaches the firing threshold ϑ a spike (action potential), with an amplitude of about 100 mV, propagates from the neuron to all the postsynaptic neurons connected to it. Then the membrane potential decays back to a value below the resting potential through a hyperpolarisation phase which is called spike-afterpotential. The critical membrane values that can surpass the firing threshold and release a spike is about 20 to 30 mV above the resting potential. To reach this critical value about 20 to 50 presynaptic spikes must arrive within a short time window before the postsynaptic action potentials are triggered. After an action potential, a refractory period occurs in which the neuron is not able to emit a spike even though it reaches the firing threshold and generally lasts one millisecond.

In Figure 2-9, the shape of a postsynaptic action potential (PSP) is determined by the time course $(t - t_i^{(f)})$ which denoted the time that has passed since the last output spike. Presynaptic

input spikes that arrive shortly after neuron i emits a spike, have a smaller effect on the membrane potential than those which arrive much later.

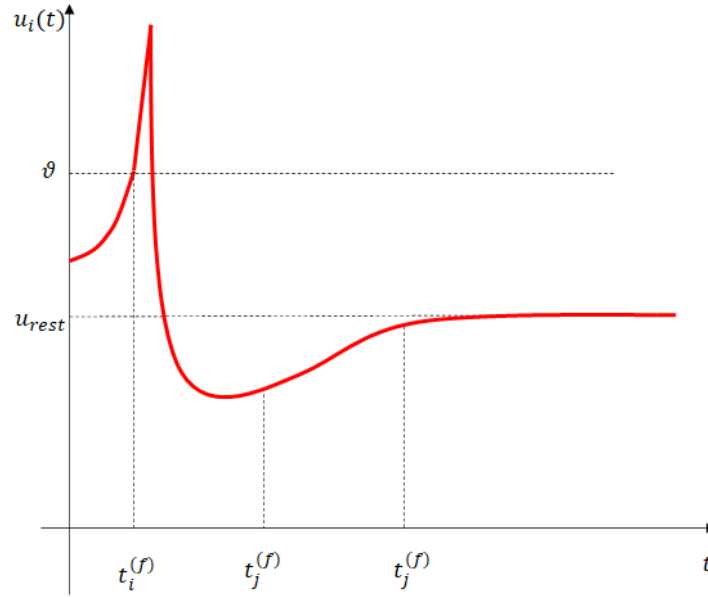


Figure 2-9: The shape of postsynaptic potential (PSP)

The membrane potential linearly responds to all input spikes received from presynaptic neurons. In other words, the total PSP of a neuron at time t is the summation of all PSPs before it surpasses the firing threshold ϑ . This behaviour can be formalised as follows:

$$u_i(t) = \sum_j \sum_f \epsilon_{ij} (t - t_j^{(f)}) + u_{rest} \quad (2-4)$$

In Eq. 2-4, $u_i(t)$ is the membrane potential of the postsynaptic neuron i at time t which has several synaptic connections to the presynaptic neurons $j = 1, 2, \dots, n$, ϵ_{ij} is the postsynaptic potential (PSP) which is evoked by a spike, $t_j^{(f)}$ indicates the time that the presynaptic neuron j propagates a spike to the postsynaptic neuron i , and $\epsilon_{ij} (t - t_j^{(f)})$ is the time course of PSP that evokes by presynaptic spikes. Figure 2-10, shows the schematic diagram of the behaviour of a postsynaptic neuron i when receives spikes from two presynaptic neurons $j = 1$ and $j = 2$.

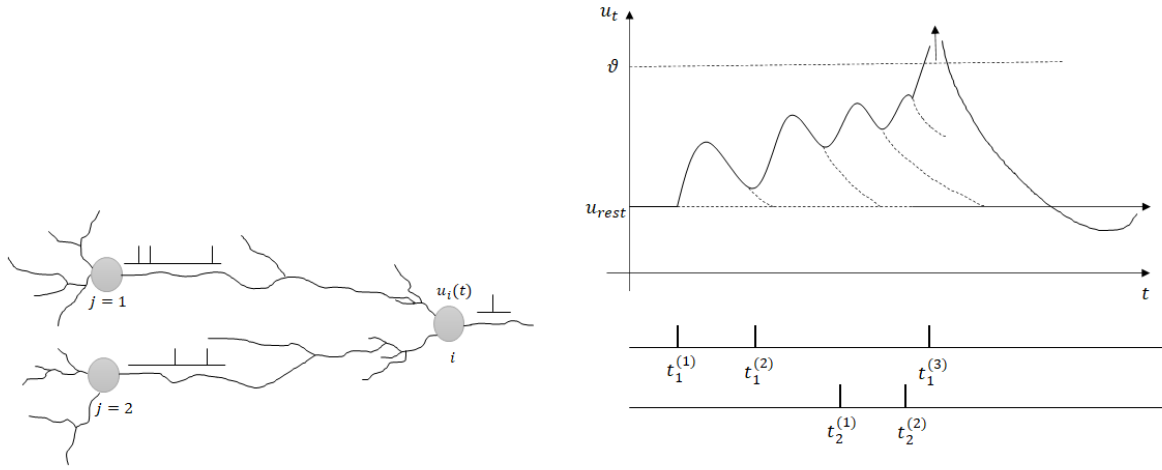


Figure 2-10: A schematic diagram of a postsynaptic neuron behaviour which receives spikes from two presynaptic neurons (Gerstner and Kistler, 2002)

The firing time $t_i^{(f)}$ ($f = 1, 2, \dots$) is the moment of passing the firing threshold ϑ . The spike train of the neuron i which is the neuron's response to a stimulating current is denoted as a sequence of firing times:

$$S_i(t) = \sum_f \delta(t - t_i^{(f)}) \quad (2-5)$$

where $\delta(x)$ is the Dirac function, $\delta(x) = 1$ means a spike occurs and $\delta(x) = 0$ means no spike:

$$\begin{cases} \delta(x) = 0 & x = 0 \\ \int_{-\infty}^{\infty} \delta(x) dx = 1 & x \neq 0 \end{cases} \quad (2-6)$$

Since the action potential always has roughly the same form, the trajectory of the membrane potential during spike can be described by a certain time course denoted by $\eta(t - t_i^{(f)})$. The term ϵ_{ij} in Eq. (2-4) describes the neuron i response to spikes of presynaptic neuron j and η describes the form of the spike and spike-afterpotential.

In a biological study of neurons, four types of neuronal dynamics are observed in the presence of a stimulating current I (Gerstner and Kistler, 2002).

- 1) **Adaptation** – Most neurons in the brain will respond to the stimulating current steps with a spike train. The time intervals between spikes increase successively until the neuron reaches a steady state of periodic firing dynamics. This adaptation behaviour is a slow process and happens after several spikes' arrival.
- 2) **Inhibitory** – In the brain, there are fast spiking neurons as well that show a faster adaptation. Many inhibitory neurons in the brain show fast neuronal dynamics in the presence of a stimulus.
- 3) **Bursting** – In this type of neuron, neuronal response to a constant stimulation is a sequence of spikes that is periodically interrupted by rather long intervals.
- 4) **Rebound** – Another observed neuronal dynamic, is a response to an inhibitory input called a post-inhibitory rebound. Even an inhibitory input can trigger an action potential, many neurons respond to the inhibitory spike with one or more rebound spikes.

Figure 2-11 shows the aforementioned four types of neuronal dynamics in the presence of a stimulating current I .

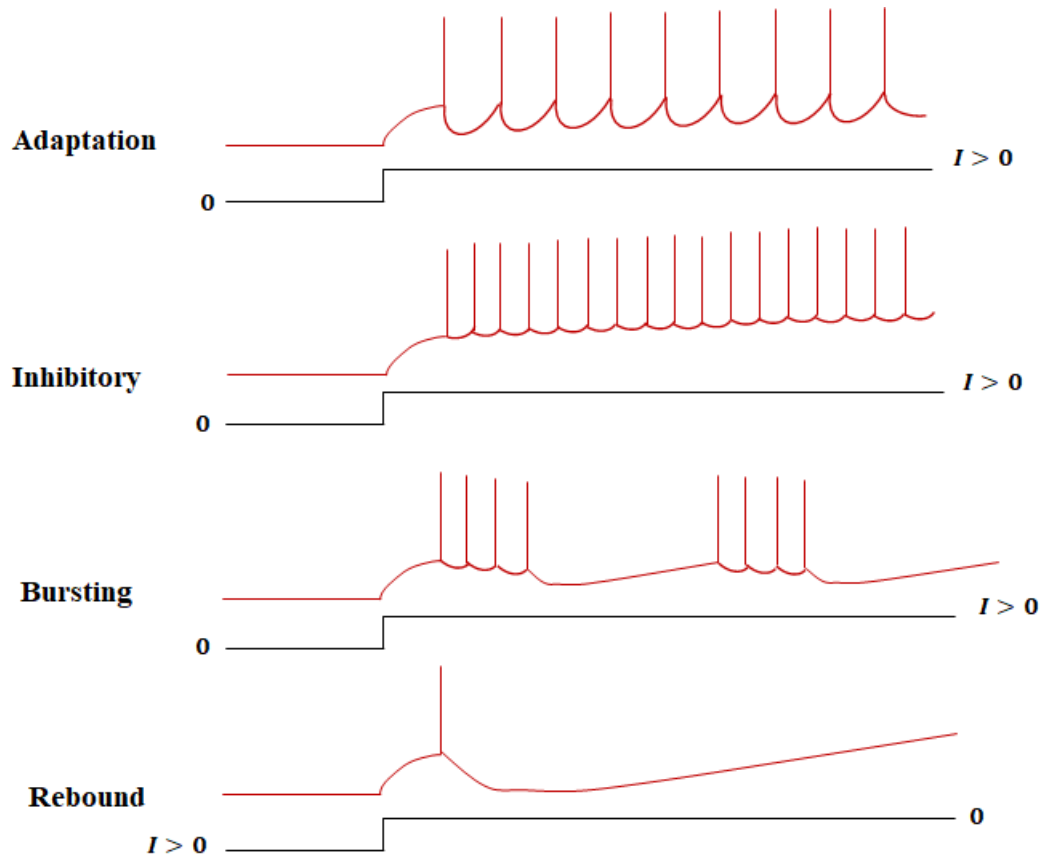


Figure 2-11: Four types of neuron response dynamics to a stimulating current (Gerstner and Kistler, 2002)

The biological neural dynamics are very complex and depends on a lot of environmental parameters. So far, several simplified models have been proposed to model the above-mentioned biological behaviour of a neuron for computational purposes. Amongst them, three formal neuron models: Leaky-Integrate-and-Fire (LIF), Izhikevich neuron model, and Probabilistic Spiking Neuron model (Knight, 1972; Izhikevich, 2003; Kasabov, 2010) are chosen to explain in this thesis because of their computational efficiency and biological plausibility. In this chapter, LIF and Izhikevich neuron models are explained as the basic neuron models for the SNNr module. Furthermore, in chapters 4 and 8, the Probabilistic Spiking Neuron model, as well as a novel neuron model based on Microtron nuclear accelerator, are introduced, respectively.

The **Leaky-Integrate-and-Fire (LIF) neuron model** is the best-known formal spiking neuron model. LIF model which simulates a threshold model of neuronal firing, is a parallel capacitor

C – resistor R circuit driven by a current $I(t)$. The stimulating current $I(t)$ charges the RC circuit and causes the membrane potential voltage $u(t)$ increases over the capacitor. Once $u(t)$ passes the firing threshold ϑ (i.e., $u(t) = \vartheta$) at time $t_i^{(f)}$, an output pulse $\delta(t - t_i^{(f)})$ is generated. After applying a low-pass filter on the presynaptic spike $\delta(t - t_j^{(f)})$ an input current pulse $\alpha(t - t_j^{(f)})$ is generated. After a spike is generated by the postsynaptic neuron i , the membrane potential leaks to $u_{rest} < \vartheta$ and the neuron enters the refractory phase. The leakage between spikes is determined by the τ_m parameter which is referred to as the membrane time constant. Eq. 2-7, formulate the LIF model.

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t) \quad (2-7)$$

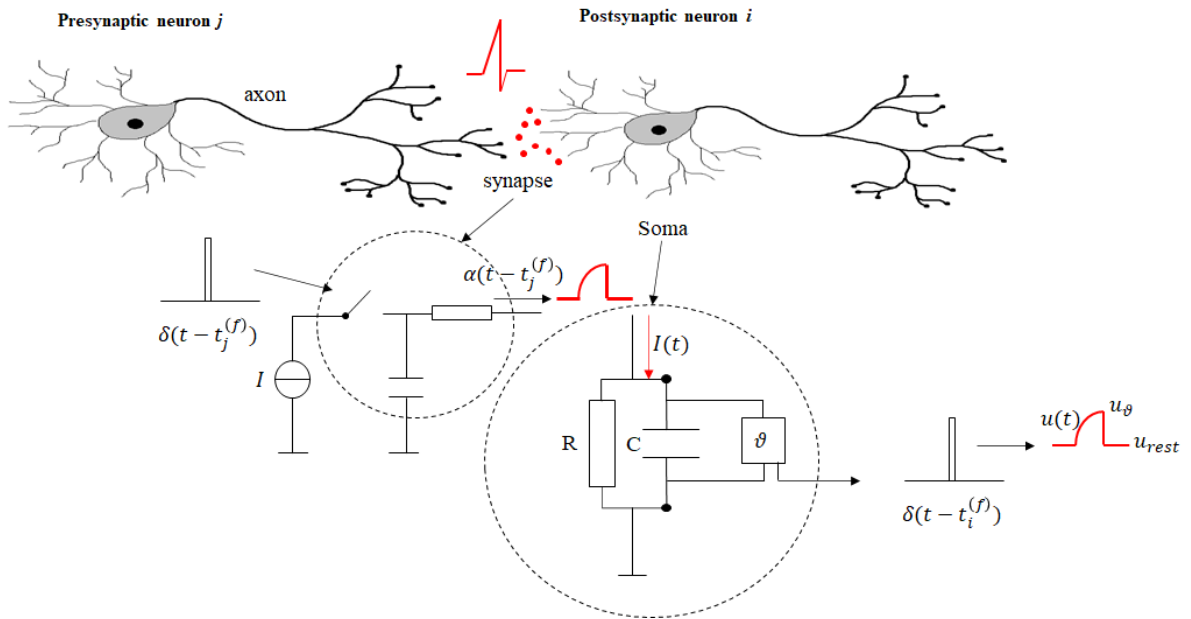


Figure 2-12: LIF neuron model (Gerstner and Kistler, 2002)

In a network of spiking neurons with the LIF model, the current $I(t)$ is generated by the activity of the presynaptic neurons. As it was mentioned before, if the presynaptic neuron j fires a spike

at $t_j^{(f)}$, the postsynaptic neuron i receives a current with time course $\alpha(t - t_i^{(f)})$. Then the total input current to neuron i can be calculated as follows:

$$I_i(t) = \sum_j w_{ij} \sum_f \alpha(t - t_i^{(f)}) \quad (2-8)$$

where w_{ij} is a measure of the efficacy of the synaptic connection between neuron j and i .

Figure 2-12 demonstrates the LIF neuron model behaviour.

The **Izhikevich neuron model** benefits from both the computational efficiency of the integrate-and-fire neurons and the biologically plausible principle of the Hodgkin–Huxley model. This model is a two-dimensional (2-D) system of ordinary differential equations of the form:

$$\frac{dv}{dt} = 0.04v^2(t) + 5v(t) + 140 - u(t) + I(t) \quad (2-9)$$

$$\frac{du}{dt} = a(bv(t) - u(t)) \quad (2-10)$$

where v represents the membrane potential of the neuron, u is a membrane recovery variable, which provides negative feedback to v , and I is the synaptic current injected to the neuron. Once the membrane potential reaches the firing threshold ϑ (+30 mV), the membrane voltage and the recovery variable are reset according to Eq. 2-11.

$$\text{if } u > \vartheta \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (2-11)$$

where a is the time scale of the recovery variable u . The smaller values of a result in slower recovery. The parameter b illustrates the sensitivity of the recovery variable u to the subthreshold fluctuations of the membrane potential v . The parameter c indicates the after-

spike reset value of the membrane potential v . The parameter d describes the after-spike reset of the recovery variable u . The recommended parameter setting according to Izhikevich (2003) are $a = 0.02$, $b = 0.2$, $c = -65$ mV and $d = 2$. Izhikevich demonstrated that with different parameter setting for a , b , c , and d the neuronal dynamics of well-known cortical excitatory neurons can be simulated. The current $I(t)$ is generated by the activity of the presynaptic neurons in a network of spiking neurons can be calculated using Eq. 2-8.

2.4.2.3 Unsupervised Learning in the NeuCube SNNr Module

After creating the network connectivity, initialising the synaptic weights, and defining the neurons' model in the reservoir module, the synaptic weights must be modified according to a learning rule to perform recognition tasks.

According to the biological observation of neuron dynamics, the amplitude of the postsynaptic response can change over time in the presence of the incoming stimulus. This phenomenon indicates a characteristic of the synaptic connections that are said to be the basis of memory and learning in the brain. The efficacy of the synaptic changes called synaptic plasticity can be described by a synaptic weight parameter. If the incoming stimulus causes a persistent increase in the synaptic transmission efficacy, then it is called long-term potentiation (LTP) of a synapse. On the other hand, if the result of stimulation causes a decrease in the synaptic efficacy, it is called long-term depression of synapse (LTD).

In the spiking neural network, a synaptic weight parameter w_{ij} is assigned to each synaptic connection to illustrate the amplitude of the postsynaptic response to an incoming action potential (spike). These synaptic weights can be adjusted or modified based on a learning rule to optimise the performance of the network for a given recognition task.

Spike-Timing Dependent Plasticity (STDP) learning (Song, Miller, & Abbott, 2000) is a type of Hebbian learning rule. It governs the synaptic efficacy based on the temporal correlation

between presynaptic spike arrival and postsynaptic firing. STDP decreases the amplitude of postsynaptic response if repeated presynaptic spikes arrive a few milliseconds before postsynaptic action potentials. This early arrival leads to Long-Term Potentiation (LTP) of the synapses. On the other hand, STDP increases the amplitude of postsynaptic response if repeated presynaptic spikes arrive a few milliseconds after postsynaptic action potentials. This late arrival leads to Long-Term Depression (LTD) of the synapses (Martinez and Derrick, 1996).

The change in the synaptic weight between a presynaptic neuron j and a postsynaptic neuron i (Δw_j) depends on the relative timing between presynaptic spike arrivals and postsynaptic firing. Considering t_j^f is a presynaptic spike arrival time at synapse j where $f = 1, 2, 3, \dots$ and t_i^f with $f = 1, 2, 3, \dots$ is the firing times of the postsynaptic neuron i . The total weight change Δw_j is calculated as:

$$\Delta w_j = \sum_{f=1}^N \sum_{n=1}^N W(t_i^f - t_j^f) \quad (2-12)$$

where $W(x)$ is the learning window calculated as follows:

$$W(x) = A_+ \exp\left(-\frac{x}{\tau_+}\right) \quad \text{for } x > 0 \quad (2-13)$$

$$W(x) = A_- \exp\left(\frac{x}{\tau_-}\right) \quad \text{for } x < 0$$

The parameters A_+ and A_- scale the strength of potentiation and depression, respectively. The time constants $\tau_+ = 10ms$ and $\tau_- = 10ms$ define the width of the positive and negative learning window. Figure 2-13 illustrate the schematic diagram of the STDP rule.

Spike-Timing Dependant Plasticity (STDP)

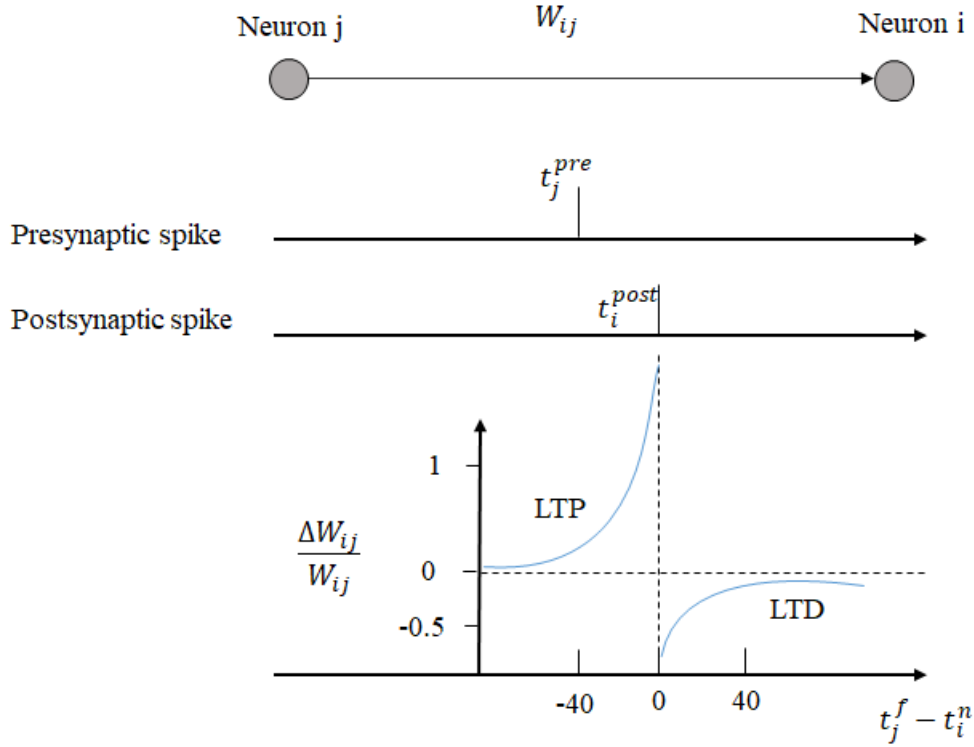


Figure 2-13: Spike-Timing Dependent Plasticity (STDP) rule

2.4.3 Evolving Output Classification Module in NeuCube

In one implementation, the output classification module in NeuCube uses the dynamic evolving spiking neural network (deSNN) (Kasabov et al., 2013) which is a fast online classifier to train the output neurons based on an association between class labels and the training samples. To this end, after the SNNr module is trained using STDP unsupervised learning rule, the input data is again propagated through the network and the output classifier learns its behaviour and dynamically assigns it to an output class neuron. In the NeuCube architecture, all the SNNr neurons are connected to each of the evolved neurons of the deSNN classifier. This network structure allows deSNN to learn the complex Spatio-temporal patterns generated in the SNNr.

The following section describes the deSNN supervised learning rule used in NeuCube in more detail.

2.4.3.1 Supervised Learning in the Output Classification Module

As it was mentioned before, NeuCube uses deSNN as its supervised learning mechanism along with the STDP unsupervised learning rule to adapt the postsynaptic weights connected to the classifier from reservoir neurons.

The deSNN algorithm uses the rank-order (RO) learning and Spike Driven Synaptic Plasticity (SDSP) spike-time learning rule in a semi-supervised way to train neurons to recognise the Spatio-temporal pattern using only one iteration of training in an online mode (Kasabov et al., 2013). In fact, deSNN uses both the information which is contained in the order of the first input spikes and the information extracted from the timing of the later incoming spikes that is learnt by the dynamic synapses. In deSNN, the information coming from the order of the incoming spikes is learnt using RO learning rules and then the initial connection weights are modified through SDSP learning over the rest of the Spatio-temporal pattern.

The deSNN method in NeuCube creates output neurons (evolved) incrementally for each training sample to capture clusters of input data and assign them to a winning class output node. There are two variants of deSNN used in NeuCube during the recall phase; a) deSNNm which uses the SNNr neurons' postsynaptic potentials (PSP) dynamics to identify the winning neuron, b) deSNNs which defines the winning neuron based on Euclidean distance between neurons' synaptic weights over time. The following deSNN learning mechanism is described in more detail.

In the deSNN training phase, for each M-dimensional training input pattern P_i , a new output neuron i is created and connected to each neuron in the "trained" SNNr module. The synaptic weights w_{ji} ($j = 1, 2, \dots, M$) between the input neuron j in the reservoir and the newly created

output neuron i are calculated based on the order of the incoming spikes on the corresponding synapses using the RO learning rule:

$$w_{j,i} = \alpha \cdot \text{mod}^{\text{order}(j,i)} \quad (2-14)$$

where α is a learning parameter (between 0 and 1), the mod is a modulation factor which indicates the importance of the order of the first spike (between 0 and 1), $w_{j,i}$ is the synaptic weight between a presynaptic neuron j and the postsynaptic neuron i , and $\text{order}(j,i)$ denotes the order of the first spike arriving through synapse ji among all spikes arriving from all other synapses to the neuron i . For the first spike to neuron i the $\text{order}(j,i)$ is 0 and increases according to the input spike order at other synapses.

After initializing the synaptic weight $w_{j,i}$ according to the first spike at the synapse j , this synaptic weight is adjusted using the SDSP (or STDP) learning rule. According to SDSP rule, if there is a spike at synapse j at time t , the synaptic weight $w_{j,i}$ increases with a small positive value called the positive drift parameter. On the other hand, if there is no spike at synapse j at time t , the synaptic weight $w_{j,i}$ decreases with a small negative value called the negative drift parameter.

$$\Delta w_{j,i}(t) = e_j(t) \cdot D \quad (2-15)$$

In Eq. 2-15, while presenting the learnt input pattern P_i to the output neuron i , if at time t there is a successive spike at synapse j , $e_j(t) = 1$ and if at time t there is no spike at synapse j , $e_j(t) = -1$. Under the influence of synaptic dynamisms denoted by $e_j(t)$ parameter, the parameter D can have positive or negative values for up and down synaptic drifts. This behaviour helps the output neurons capture the temporal relationship of spike timing across the learnt pattern P_i at time t .

While presenting the input training pattern to the output neurons, a spiking threshold Th_i is assigned to the output neuron i . The firing threshold Th_i is calculated as a fraction (C) of the total PSP_i (called PSP_{imax}) of neuron i accumulated during the presentation of the input pattern in a time-window T . During the recall phase, this firing threshold makes the output neuron release a spike in the presence of patterns similar to the learnt ones during training.

$$PSP_{imax} = \sum_{t=0}^T \sum_{j=1}^M f_j(t) \cdot w_{j,i}(t) \quad (2-16)$$

$$Th_i = C \cdot PSP_{imax} \quad (2-17)$$

In Eq. 2-16, T represents the time units in which the input pattern is presented to the output neuron, M is the number of the input synapses to the neuron i , $f_j(t)$ shows the firing dynamics of synapse j at time t (i.e., $f_j(t) = 1$ if there is a spike at synapse j for this learnt input pattern at time t and $f_j(t) = 0$ if otherwise), and $w_{ji}(t)$ is the efficacy of the synapse ji calculated at time t using of Eq. 2-15.

The parameter C that is used for calculating the Th_i threshold (Eq. 2-17) enables neuron i to emit a spike (recognise the input pattern) before the whole learnt pattern is presented.

As it was mentioned before, in the recall phase, deSNN can use two different approaches for comparing input patterns with already learnt patterns in the output neurons:

- a) deSNNm – In this approach, the output neuron's membrane potential (PSP) is used to define the best output match for the input pattern. Namely, the new data input which is encoded to a spike train is propagated to all trained output neurons and the first output neuron that responds to it with a spike (i.e., its PSP reaches the Th_i threshold) determines

the output class. Eq. 2-18, formulate the membrane potential calculation for recalling a pattern:

$$PSP_i(t) = \sum_{l=0}^T \sum_{j=1}^M f_j(l) \cdot w_{j,i}(l) \quad (2-18)$$

where t is the current time unit during pattern presentation in the recall mode, M is the number of the input synapses to the neuron i , $f_j(l) = 1$ if there is a spike at time l at synapse j for the recall pattern and $f_j(l) = 0$ if there is no spike, and $w_{ji}(t)$ is the efficacy of the synapse ji calculated at time l .

- b) deSNNs – In this approach, for each recall input pattern a new output neuron is created in the way that is created in the training phase, then the synaptic weight vector of the newly created output neuron is compared to the existing output neurons using Euclidean distance. The closest output neuron in terms of synaptic connection weights to the new neuron is the ‘winner’ output class. This method, which is inspired by transductive reasoning principles, uses nearest neighbour classification in the synaptic weight space to capture a new pattern or match it to an existing one. Figure 2-14 and Figure 2-15 illustrate the deSNN pseudocode and learning schema, respectively.

Dynamic Evolving Spiking Neural Network (deSNN)

- 1: Initialise deSNN parameters: Mod , D , C , α , and sim (similarity threshold)
- 2: for each input pattern P_i
- 3: create a new output neuron i
- 4: initialise synaptic weights $w_i(0)$ using RO learning rule denoted in Eq. 2-14
- 5: modify the synaptic weights w_i for successive spikes on the corresponding synapses using SDSP rule denoted in Eq. 2-15
- 6: calculate PSP_{imax} using Eq. 2-16
- 7: calculate the firing threshold Th_i for neuron i using Eq. 2-17

```

8:         if the new wight vector  $w_i$  is similar to an existing output neuron using Euclidean distance
           similarity measure and sim threshold
9:             merge the two neurons by averaging their threshold and synaptic weights
10:         else
11:             add new output neuron to the network
12:         end if
13:     end for

```

Figure 2-14: Pseudocode for deSNN algorithm (Kasabov et al., 2013)

The deSNN algorithm is a computationally efficient learning algorithm due to its one-pass online incremental learning mechanism and improves the classification times remarkably.

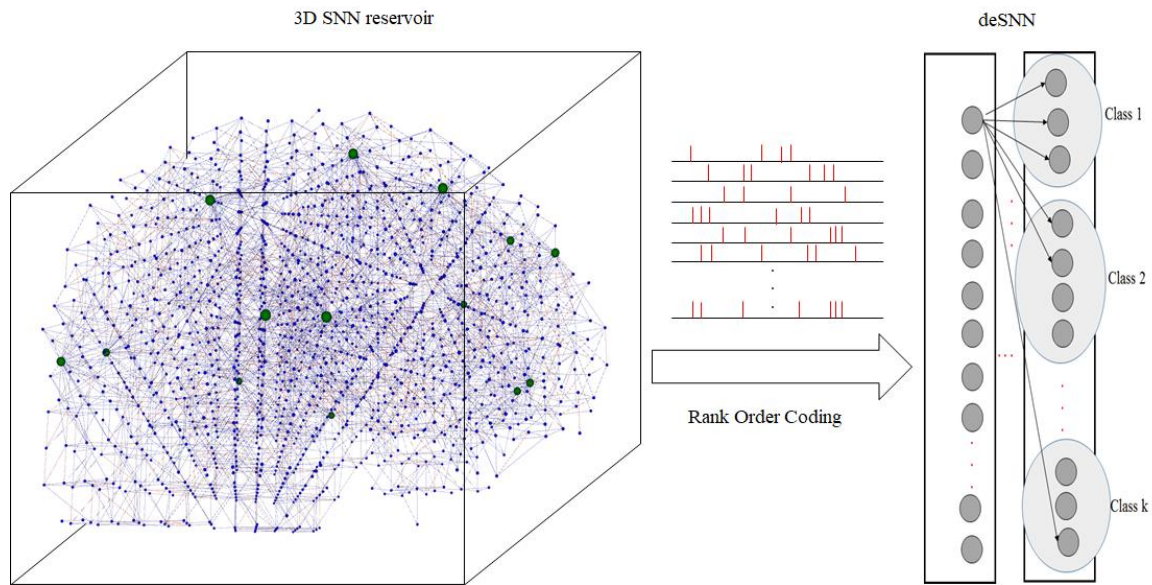


Figure 2-15: The deSNN learning schema.

2.5 Chapter Summary

This chapter provided the background information about some principles of brain information processing used further in this thesis for the development of new SNN methods. The chapter also describes the SNN architecture NeuCube (Kasabov, 2014) used as a baseline and as inspiration for the development of the new SNN methods in the rest of the chapters. Altogether,

the biological brain structure, learning, and memory along with the computational models and framework introduced in this chapter have helped to propose and develop the brain-like structural and functionality for the novel framework and learning algorithms in this thesis.

Chapter 3 . Evolving Predictive Unsupervised Supervised deep learning algorithms for Spike Streams: EPUSSS

Abstract

This chapter presents the Evolving Predictive Unsupervised Supervised deep learning algorithms for Spike Streams (EPUSSS). EPUSSS is a unified framework for SNN, including different algorithms to perform prediction and classification in a two-level hierarchical fashion. The main principles of this model were first proposed in (Kasabov, 2018, pages 221-226), and here they are fully developed and improved as a working framework, along with their illustration on exemplar data. Since EPUSSS is a deep learning SNN model, this chapter introduces and reviews deep learning neural networks as a relevant comparative background to the topic.

The chapter addresses the following research questions:

- **RQ1** “How an unsupervised learning in SNN be integrated with local supervised and global supervised deep learning methods for predictive modeling and classification in one unified framework?”
- **RQ2** “How can different types of neurotransmitter receiver be incorporated into SNN’s model synaptic connections to balance network’s spiking activity and help to create more precise patterns?”
- **RQ3** “How can a novel energy function be introduced for a life-long learning in SNNs?”

3.1 Introduction

One of the most prominent challenges scientists face is making sense of complex dynamic patterns found in multimodal streaming data, ‘hidden deep in time’. If such patterns can be interpreted, then our ability to explain phenomena in nature, understand the mechanisms of human cognition, and predict future events will be significantly improved. The current state-of-the-art of Artificial Intelligence (AI) is deep neural networks (DNNs) (Bengio, 2009; Schmidhuber, 2014; LeCun et al., 2015; Goodfellow et al., 2016). Despite their success in large-scale pattern recognition (Krizhevsky et al., 2012; Esteva et al. 2017), they have severe constraints when learning from continuous streaming data. They do not capture patterns from data that include both time and space, they are slow to learn, require the processing of unnecessarily large amounts of data, and are mainly applied to static datasets.

Inspired by the ability of the human brain to learn and predict long temporal sequences (e.g., music, texts, navigation pathways, etc.), a novel computational model for deep-in-time machine learning and predictive modelling of streaming data called Evolving Predictive Unsupervised-Supervised deep learning algorithms for Spike Streams (EPUSSS) is introduced here. In the proposed approach, data continually stream into a brain-like model at every time point (millisecond, day, etc.). As causal relationships of patterns are learnt, the model retains and evolve its structure. The knowledge of these causal relations become more deeply embedded as more temporal data enters the model.

The model is characterized by 1) learning patterns from multimodal streaming data; 2) fast, incremental adaptive, and theoretically ‘life-long’ learning; 3) the rule allows the SNN to evolve its structure from data; 4) apply unsupervised and supervised learning modes; 5) early and accurate future event prediction including events that are hidden deep-in-time.

The objective of the proposed method is to create an integrated computational model of continuous deep learning of streaming data or alternatively, of temporal (Spatio-temporal) data of fixed time window, encoded as spike streams. As only a partial case, the data can be of a fixed time window, but without any restrictions in the length. A model will learn all data entered as a sequence of spike frames and will evolve meaningful internal Spatio-temporal patterns that can be stored from time to time for retrospective analysis. If an input pattern of a specified length is entered, with a known output, a specific pattern will be activated (as a deep trajectory of spikes in the model) and this pattern can be learnt for the prediction task.

3.2 Previous Work

A Predictive model uses historical data to provide a calculated suggestion for future events by modelling the dynamics of a complex system. Predictive models have been widely used in many real-world applications like financial time series prediction, weather and natural hazards forecasting, prediction of the progress of a disease, etc. In the following, a brief review of spiking neural models for prediction tasks is provided.

3.2.1 Predictive Models Using Spiking Neural Networks

Yang and Zhongjian (2011) introduced a predicting model based on spiking neural networks (SNNs) to predict China annual grain yields. Kaplan et al. (2013) inspired by the retinotopic cortical areas architecture, presented a network model of conductance-based integrate-and-fire neurons. They implemented predictive coding through network connectivity using connection delays and tuned properties of the source and target cells to predict the motion in tracking a moving dot experiment. Reid et al. (2013) used a Polychronous Spiking Network for financial time series prediction. Reid et al. (2014), in another effort, used a Polychronous Spiking

Network for the prediction of sunspot and auroral electrojet index. Tu et al. (2014) used a Spiking Neural Network reservoir (SNNr) and dynamic evolving Spiking Neuron Network (deSNN) classifier within the NeuCube framework for the prediction tasks. Gibson et al. (2014) presented learnable temporal delays at both dendrites and axons for a spiking neural network to forecast temporal sequences in a multi-step prediction mechanism. In their model, neurons can function asynchronously to predict future events in a video sequence by learning the temporal structure of space-time events adapting to multiple scales. Gibson et al. (2014) proposed a spiking neural network to perform a long-range prediction task based on the DVS128 data. Gilra and Gerstner (2017) proposed a supervised learning scheme for the feedforward and recurrent connections in a network of heterogeneous spiking neurons to reproduce the non-linear body dynamics caused by motor commands. They backpropagated the output error, which depends on the presynaptic activity, through the random connections with a negative gain, causing the network to follow the desired dynamics. At the same time, an online and local rule changes the weights. Zhou and Wachs (2017) proposed the Cognitive Turn-taking Model (CTTM) to achieve early turn-taking prediction. This model process multimodal human communication cues both implicit and explicit and predict human turn-taking intentions in an early stage. Brusca et al. (2017) utilized the NeuCube framework for the prediction of wind farm energy production. Hamano et al. (2017) used a 3-dimensional network of locally connected neurons with the NeuCube framework to predict blood pressure. Despite the remarkable outcome of the above-mentioned research works, spiking neural networks suffer from scalability and accuracy. They also lack a proper mechanism to deal with large scale data without affecting the accuracy and computational time. Another missing concept here is deep learning techniques to improve the ability to extract hidden features from input data to improve learning outcomes. In the following, some of the works in the literature on the deep learning approaches combined with SNNs are reviewed.

3.2.2 Deep Learning in Spiking Neural Networks

Since the last decade, the concept of deep structured learning, which is commonly called deep learning or hierarchical learning, has emerged as a new area of machine learning and signal processing research. Deep learning models consist of multiple layers or stages of nonlinear information processing and includes methods for supervised or unsupervised learning of feature representation at successively higher, more abstract layers. Some of the deep learning techniques proposed so far are Stacked Autoencoders, Deep Belief Networks, Deep Boltzmann Machine, Convolutional Neural Networks, Data Stacking Networks, Compound Hierarchical-Deep Models, Convolutional Deep Belief Networks and Deep Long-Short Term Memory Neural Networks (Deng and Yu, 2013; Langkvist et al., 2014; Cai and Liu, 2016).

Although deep learning has demonstrated very successful performance in learning complex, high dimensional features, it suffers from the famous problem of vanishing or exploding gradients when the error signal is back-propagated, also known as the longtime lag problem. Initialization of the deep learning architecture including the number of layers and their neurons' activation functions and input connections is another difficulty (Langkvist et al., 2014; Schmidhuber, 2015). Currently, many researchers are trying to overcome deep learning drawbacks (Schmidhuber, 2015).

Deep learning techniques have been successfully applied to time-series data (Langkvist et al., 2014; Gashler and Ashmore, 2016) which share a lot of similar features with spatio-spectro temporal data (SSTD) which Spiking Neural Networks (SNNs) due to using trains of spikes (binary temporal events) transmitted among spatially located synapses and neurons have potential to learn them effectively (Gerstner and Kistler, 2002). Although deep learning relies

on a differentiable activation function and cannot handle discrete spikes, there are several efforts done to adapt deep learning mechanisms to SNNs:

Diehl et al. (2016) presented a train-and-constrain methodology that enables the mapping of machine-learned (Elman) RNNs on a substrate of spiking neurons while being compatible with the capabilities of current and near-future neuromorphic systems. This "train-and-constrain" method consists of first training RNNs using backpropagation through time, then discretizing the weights and finally converting them to spiking RNNs by matching the responses of artificial neurons with those of the spiking neurons. O'Connor and Welling (2016) introduced a spiking Multi-Layer Perceptron to do backpropagation on a spiking network. The authors formulate a deep spiking network whose function is equivalent to a deep network of Rectified Linear (ReLU) units. Then a spiking version of backpropagation to train this network is introduced. Compared to a traditional deep network, the Deep Spiking Network can make "early guess" about the class associated with a stream of input events, before all the data has been presented to the network, the training procedure consists only of addition, comparison, and indexing operations, which potentially makes it very amenable to efficient hardware implementation, the amount of computation that the network does is a function of the data, rather than the network size. This is especially useful given that the proposed network tends to learn sparse representations. Lee et al. (2016) proposed a novel technique, which treats the membrane potentials of spiking neurons as differentiable signals, where discontinuities at spike times are only considered as noise. This enables an error backpropagation mechanism for deep SNNs, which works directly on spike signals and membrane potentials. Thus, compared with previous methods relying on indirect training and conversion, their technique has the potential to capture the statics of spikes more precisely. Esser et al. (2016) adapted deep convolutional neural networks to perform classification tasks on neuromorphic hardware. The authors provided a description of the relevant elements of deep convolutional networks and the TrueNorth

neuromorphic chip. They described how the essence of the former can be realized on the latter. Esser et al. (2015) resolved the incompatibility between backpropagation, which uses continuous-output neurons and synaptic weights, and neuromorphic designs, which employ spiking neurons and discrete synapses. Their approach was to treat spikes and discrete synapses as continuous probabilities, which allows training the network using standard backpropagation. The trained network naturally maps to neuromorphic hardware by sampling the probabilities to create one or more networks, which are merged using ensemble averaging. Hunsberger and Eliasmith (2015) trained spiking deep networks using leaky integrate-and-fire (LIF) neurons and achieve state-of-the-art results for spiking networks on the CIFAR-10 and MNIST dataset. They showed that biologically plausible spiking LIF neurons can be integrated into deep networks and can perform as well as other spiking models. Diehl et al. (2015) presented a set of optimisation techniques to minimize performance loss in the conversion process for ConvNets and fully connected deep networks. Their techniques include using rectified linear units (ReLUs) with zero bias during training and using a new weight normalization method to help regulate firing rates. The authors used a method for converting an ANN into an SNN that enables low latency classification with high accuracies already after the first output spike. Cao et al. (2015) proposed a novel approach for converting a deep CNN into an SNN that enables mapping CNN to spike-based hardware architectures. Their approach first adapts the CNN architecture to fit the requirements of SNN, then trains the adapted CNN in the same way as one would with CNN, and finally applies the learnt network weights to an SNN architecture derived from the tailored CNN. Henderson et al. (2015) proposed a scheme for learning connectivity in spiking neural networks that learns instantaneous firing rates that are conditional on the activity in other parts of the network. Their scheme is independent of the choice of neuron dynamics or an activation function and network architecture. Their method involves two simple, online, local learning rules that are applied only in response to occurrences of

spike events. Indiveri et al. (2015) presented a full custom hardware implementation of a deep neural network, built using multiple neuromorphic VLSI devices that integrate analogue neuron and synapse circuits together with digital asynchronous logic circuits. The deep network comprises an event-based convolutional stage for feature extraction connected to a spike-based learning stage for feature classification. Huang et al. (2015) provided an overview of the Extreme Learning Machine (ELM) from the theoretical perspective, including the interpolation theory, universal approximation capability, and generalization ability. They reviewed improvements made to ELM to enhance its stability, sparsity, and accuracy under general or specific conditions and compared it with the deep learning system.

Most of the works in the above literature suggested the concept of deep learning in SNNs by combining convolutional layers with SNNs or using spiking neurons in CNN architecture. There is no method that can demonstrate deep learning of input patterns in SNN as part of its internal learning mechanism instead of embedding convolutional layers or borrowing the concept of spiking neurons in the CNN structure.

In this thesis, a novel computational SNN model for deep learning of spike streams is introduced which demonstrates deep learning as part of its learning mechanism within a recurrent network of spiking neurons.

As it was stated in chapter one, learning methods in spiking neural networks are mostly based on Spiking Time Dependent Plasticity (STDP) which is an unsupervised learning algorithm. However, there are many attempts that have been made to introduce an effective supervised learning mechanism to enhance the accuracy of the SNNs. Supervised learning in SNNs is not an easy process in comparison to Artificial Neural Networks (ANNs) because of their discontinuous nature. One of the popular supervised learning algorithms in ANNs is the error backpropagation technique because of its fast convergence that is a result of using the gradient

descent optimisation technique. However, this technique requires differentiability, which is suitable for continuous networks, while SNNs are discontinuous networks.

To overcome the limitation of using the backpropagation algorithms in SNNs, the discrete-valued functions can be approximated with linear functions during the learning phase by using a threshold function or firing rates as continuous values to solve the problem of differentiability (Bohte & Poutré, 2000; Bohte et al. 2002; Schrauwen & Campenhout 2004; Lee et al., 2016; Matsuda, 2016; Wu et al., 2018; Zhang et al., 2020; Anwani & Rajendran, 2020).

In some other approaches, STDP is combined with reinforcement learning reward and penalty mechanism inspired by the roles of neuromodulators such as Dopamine (DA) and Acetylcholine (Ach) called rewarded-modulated spike timing-dependent plasticity (R-STDP) to provide supervised learning ability for spiking neural networks (Farries & Fairhall, 2007; Legenstein, Pecevski, & Maass, 2008; Gardner, Sporea, & Gruning, 2014; Frémaux & Gerstner, 2015; Aswolinskiy & Pipa, 2015; Mozafari, et al. 2018).

The novel EPUSSS computational model introduced in this chapter has a two-level learning structure: a local learning mechanism based on the SpikeProb (Bohte, Kok, & Poutré, 2000) and a global learning mechanism according to the R-STDP rule to demonstrate more biologically plausible learning behaviour like the brain.

3.3 The EPUSSS Learning Principles

Neuroscientists proposed theories that in which learning in the brain is not only because of synaptic plasticity but also is due to some sort of error backpropagation mechanism (neuronal feedback) that leads to self-regulation of brain activities (Strehl, 2014). However, the backpropagation algorithms in artificial neural networks are biologically implausible due to the lack of local error representation, symmetry of forwards and backwards weights, and

unrealistic models of neurons (Whittington and Bogacz, 2019). The EPUSSS model which is inspired by biologically learning theories and adopted the NeuCube framework as its backbone structure provides a more biologically plausible learning model with brain-like characteristics by addressing the above-mentioned issues. EPUSSS, which is a type of recurrent SNN model has two-level learning mechanisms that resemble the biological learning in the brain:

- a) a local learning mechanism in which the synaptic plasticity is combined by neuronal feedback from input data dynamics for self-regulation of synaptic activities to learn and predict input data behaviour without considering the target class label.
- b) a global learning approach in which learning is reinforced by a neuromodulatory signal that is generated by the classification results to adjust the neuron's synaptic weights in the proportion of their contribution to learning a specific time course.

The EPUSSS architecture, which is influenced by the NeuCube framework, has three major modules: the encoder module which uses the BSA algorithm that encodes input data into spike trains; a 3D SNN reservoir module in which the local learning mechanism of EPUSSS governs the synaptic efficacy in a semi-supervised way, and the output module which deploys the global learning mechanism of EPUSSS, combined with a modified deSNN model to provide neuromodulatory feedback to modify synaptic efficacy in a supervised way. EPUSSS uses spiking neuron models such as LIF and Izhikevich.

In the EPUSSS 3D SNN reservoir, an input neuron is connected to reservoir neurons by synaptic connections and to other input neurons - by “hypersynaptic” connections. The input neurons are mapped in a 3D (2D) SNN structure according to their 3D (2D) spatial coordinates or the similarity between the input data streams. The model always learns from a spike sequence time series data to predict the next spikes in a certain time-window. Predictive modelling is achieved through a semi-supervised local learning mechanism using the error of

prediction in a certain time-window and a backpropagation rule that adjust the efficacy of synaptic weights of the reservoir neurons. The output neurons inspired by the deSNN model's concept are created incrementally for each training sample to associate the input data with a class label. For this purpose, all the input neurons and reservoir neurons (or just input neurons) are connected to output neurons using the RO method, and their synaptic weights are adjusted based on the EPUSSS global learning rule using a neuromodulatory signal that is generated by the results of the classification in the training phase. The principle behind the algorithm is to perform both prediction and classification in a two-level hierarchical learning structure. In the proposed model, the input neurons contribute enormously to both local and global learning. In local learning, the system learns to predict the input data dynamics through modifying the oscillatory rhythmic patterns of neuronal firing activities. A sliding time-window allows EPUSSS to capture the firing behaviour of the input neurons for a certain time course in which the input data is exposed to the network. The input neurons have recurrent connections from the last layer of reservoir neurons that helps them to learn the activity of the reservoir neurons and influence other input neurons to have optimal learning like the cooperative learning in the brain. Then by cross-matching input neurons firing behaviour with the input data dynamics in one (several) time unite(s) ahead the error will be calculated and backpropagated to modify the synaptic and hypersynaptic connections. This process is similar to synaptic plasticity based on local signals in the brain.

At the end of each episode of learning corresponding to the presentation of the entire input pattern with T lasting duration to the network, a neuromodulatory signal from the output neurons is generated based on the target labels. Using a Rewarded-Modulated Spike-Time-Dependent Plasticity (R-STDP), rule the generated neuromodulatory signal helps reward or penalise the synaptic and hypersynaptic connections efficacy. This process is the global learning mechanism of EPUSSS which resembles the feedback loop learning from external

stimuli in the brain. The EPUSSS architecture and its learning principles can be seen in Figure 3-1.

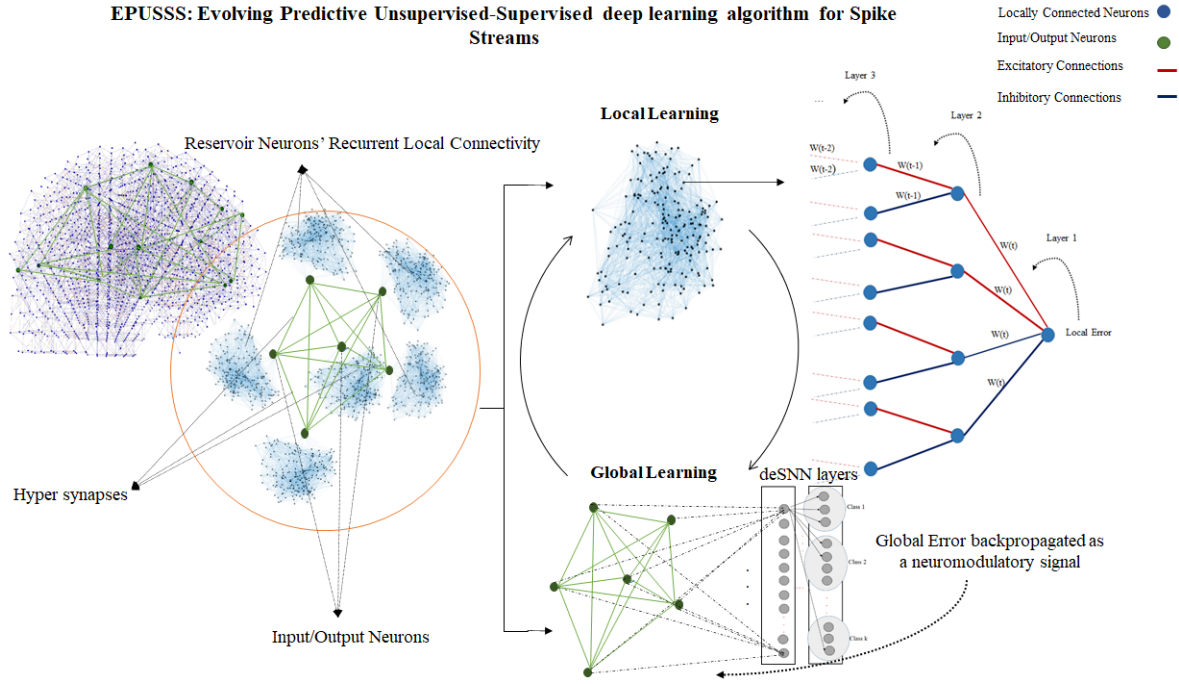


Figure 3-1: EPUSSS - Evolving Predictive Unsupervised-Supervised deep learning framework for Spike Streams

3.4 EPUSSS Novel Characteristics

The EPUSSS model has several novel characteristics that make it a powerful computational model which is more biologically plausible. In this section, biological resemblant characteristics, and computational characteristics of EPUSSS is explained.

3.4.1 Biological Resemblance Characteristics

The EPUSSS local error representation mechanism uses local signals from neuronal pre-and postsynaptic activities to modify synaptic connections' efficacy like what is known as synaptic plasticity in the brain. The aim of this mechanism is to capture the Spatio-temporal correlation in the input data space by adapting to the statistical properties of the neuronal firing behaviours. The global learning mechanism of EPUSSS uses the concept of neuromodulatory signals in the

brain, which influence the synaptic activities by reward and punishment to help learn and create memories. Neuromodulators change the excitability of neurons to form memories. Some neuromodulatory signals which have been linked to reward are released both as a response to the moment of rewards and to stimuli that are predictive of rewards. Some other neuromodulatory signals are related to sensory map remodelling and inhibitory avoidance in training. Using neuromodulatory concepts, neurons in the proposed model get feedback of reward or punishment to modify their spiking activities to form memories and recognize the learnt pattern in the presence of the external stimuli (i.e., target class labels). Another biological aspect of EPUSSS is theoretically being capable of life-long learning with the ability of transfer learning. These characteristics that allow EPUSSS to continuously learn from the input data without being saturated are as follow:

- In one implementation, EPUSSS has 80 percent excitatory and 20 percent inhibitory neurons in its spiking neural network in such a way that the inhibitory neurons spike faster than excitatory neurons to balance the network spiking behaviour.
- The synaptic connections in EPUSSS have four types of neurotransmitter receptors:
 - Excitatory neurons have two types of neurotransmitter receptors:
 - AMPAR: fast excitatory synaptic transmission receptors
 - NMDAR: slow excitatory synaptic transmission receptors
 - Inhibitory neurons have two types of neurotransmitter receptors:
 - GABAAR: fast inhibitory synaptic transmission
 - GABABR: slow inhibitory synaptic transmission

By introducing synaptic delays, the synaptic transmission time can be controlled to mimic fast and slow signal transmission like biological neurotransmitters behaviour.

- Synaptic pruning which is part of brain development is another feature of EPUSSS in which neurons that do not show activity for a certain time-window will lose their synaptic connections gradually by decreasing their postsynaptic weights. This mechanism helps to eliminate noisy neurons from the learning process and improve the model prediction and classification performance.

3.4.2 Computational Characteristics

Besides the biological characteristic, EPUSSS has adopted several computational methods to perform its learning mechanisms in a more efficient way.

In the local learning, the error backpropagation method, which is based on the SpikeProp error backpropagation rule, uses the temporal derivative of the postsynaptic input neuron's activity to adjust synaptic connections. The global learning mechanism uses the R-STDP rule to feedback the reward and penalty according to the target class labels from the deSNN output neuron to adjust both hypersynaptic and synaptic connections. In this mechanism, neurons' synaptic connections are modified according to the proportion of their contribution to the learning of input patterns. Finally, the modified Lyapunov energy function (Kobuchi, 1991; Yerramalla et al., 2003; Lee et al., 2010; Yan et al., 2013; Frady and Sommer, 2019; Stern and Shea-Brown, 2020) is used to keep the network stable while accelerating the learning process by a closed-loop control mechanism. This characteristic of EPUSSS also plays an important role in "life-long" learning ability. Namely, by controlling the synaptic activities corresponding to the outcome of learning (i.e., the energy of the network), the Lyapunov energy function prevents the network's synaptic weights saturation while EPUSSS performs classification tasks.

3.5 The Proposed EPUSSS Learning Mechanisms

EPUSSS learns from both local and global feedbacks from neuronal activities and external stimuli, respectively. In the EPUSSS architecture, multiple hierarchical areas are connected to each other through input neurons. This feature allows information from different areas (input neurons) to contribute to the learning process by affecting each other and their locally connected networks more efficiently. Furthermore, the input neurons can represent different types of data, different topological connections, and areas of interest which makes EPUSSS a powerful learning tool with explainability and interpretability that can analyse the impact of each type, areas of interest and topological connectivity on the outcome of learning.

The idea behind the EPUSSS two-phase synaptic weight updating mechanism (local and global learning) is to provide a self-regulatory ability in both unsupervised and supervised ways. In other words, the memory of unsupervised local learning will transfer from the locally connected reservoir neurons to the input neurons at the higher level. Then, these biologically plausible computational units adjust their learning experience by updating their hypersynapses according to the classification outcome. The hypersynaptic connections provide an ability to engage a different spatial region of the EPUSSS model which is a 3d representation of the brain. This approach helps to mimic the real brain learning mechanism to some extent as well as offering an efficient predictive model that can predict the next step(s) in the input patterns and classify the predicted patterns simultaneously.

3.5.1 EPUSSS Local Learning Mechanism

In the EPUSSS local learning process, the input neurons propagate the input pattern to the locally connected reservoir neurons. Then in a semi-supervised manner, the input neurons behaviour is observed in a certain time-window and compared to the input pattern dynamics at the same time-window or several steps ahead. Thereupon, the error from the comparison

between the input pattern dynamics and the input neuron spiking activity is used to adapt the synaptic efficacy of locally connected reservoir neurons to force the input neuron to capture the spikes (events) rhythm hidden in the input pattern. The process of backpropagating the temporal derivative of input neuron activity in proportion to the input pattern dynamics in a sliding time-window repeats until the end of simulation time which is the length of the input pattern.

The reason that the spiking activity of the input neuron in a specific time-window is used for the error calculation is that unlike observing the input neuron spiking activity at each time point, this value is continuous and differentiable. Moreover, using the average firing rate in a certain time-window provides a more robust approach that is less affected by the noise in the input pattern and gives the post-synaptic neuron enough time to capture the temporality hidden in the input pattern dynamics.

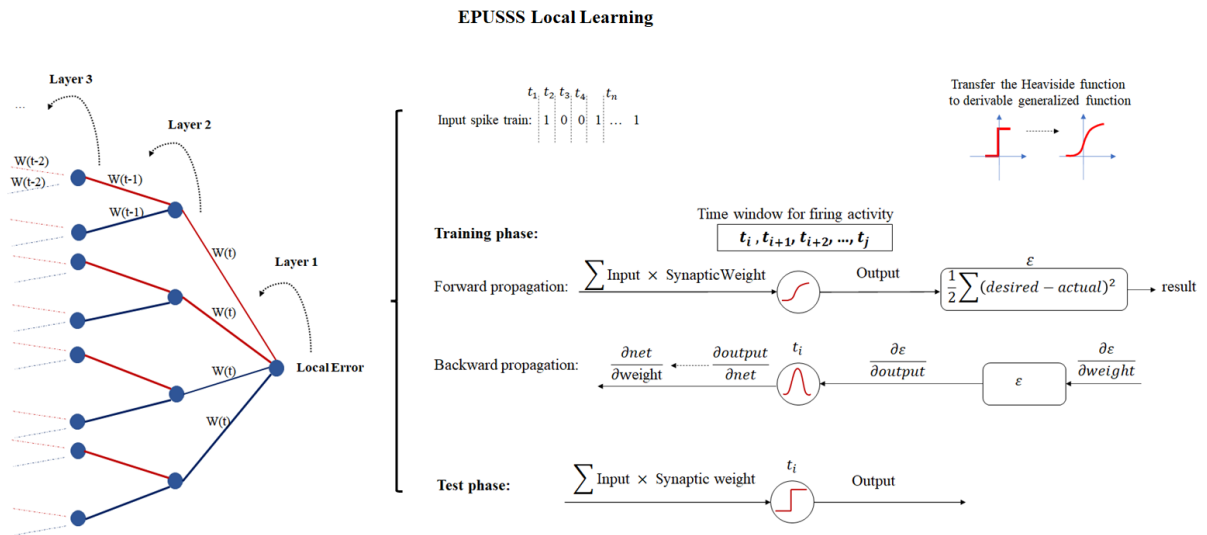


Figure 3-2: EPUSSS local error backpropagation mechanism

As it was mentioned before, in the EPUSSS model there are 80 percent excitatory and 20 percent inhibitory neurons with two different types of neurotransmitter receptors for each: fast and slow receptors.

To simulate the behaviour of neurotransmitter receptors in the network a delay factor is introduced for the synaptic connections. This delay mechanism helps the input neuron to adapt to the temporal rhythm of input pattern dynamics and create the memory for prediction tasks. The synaptic delay value d , firing threshold ϑ , and synaptic efficacy w_{ij} are modified during the learning process using an error backpropagation method to enhance model performance.

The backpropagation method used in the EPUSSS local learning is a modified version of the SpikeProb (Bohte, Kok, & Poutr , 2000). As it was mentioned before, EPUSSS uses LIF (Izhikevich) to model neuronal activity. Considering the postsynaptic neuron i receives a current with time course $\alpha(t - t_i^{(f)})$ from several presynaptic neurons j which their synaptic connections according to their type of receptors (fast or slow) has a delay factor, then the total input current to the neuron i can be calculated as follow:

$$I_i(t) = \sum_j w_{ij} \sum_f \alpha(t - t_j^f - d_{ij}^f) \quad (3-1)$$

where w_{ij} is a measure of the efficacy of the synaptic connection between neuron j and i and t_j^f is the time at which the presynaptic neuron spikes.

The target of the local learning is to predict the next values in the input pattern by capturing the input pattern dynamics in a specific time-window. Therefore, the input neuron firing activity (firing rate) in a specific time window is compared to the desired firing behaviour to calculate the local error. The error function is the least mean squares error function that is calculated as follow:

$$E_{local} = \frac{1}{2} \sum_{j \in R} (fr_j^a - fr_j^d)^2 \quad (3-2)$$

where, fr_j^a is the input firing rate in the time-window and fr_j^d is the desired firing rate that needed to be predicted from input pattern dynamics.

Using the differentiable error term in Eq. (3-2), the synaptic weight, the delay parameter, and the neuron firing threshold can be adjusted according to the desired internal input pattern behaviour.

$$\Delta w_{ij} = -\eta_w \frac{\partial E}{\partial w_{ij}} \quad (3-3)$$

$$\Delta d_{ij} = -\eta_d \frac{\partial E}{\partial d_{ij}} \quad (3-4)$$

$$\Delta \vartheta_i = -\eta_v \frac{\partial E}{\partial \vartheta_i} \quad (3-5)$$

where Δw_{ij} is the weight changes proportional to the error in a certain time window observation of neuron firing activity, Δd_{ij} is the changes for synaptic delay adaptation, and $\Delta \vartheta_i$ is the changes for neuron firing threshold adjustment. Parameters η_w , η_d , and η_v are learning rate for synaptic weights, synaptic terminal delays, and firing thresholds, respectively. The error is minimised by changes in the weights, delays, and firing thresholds according to the negative local gradient.

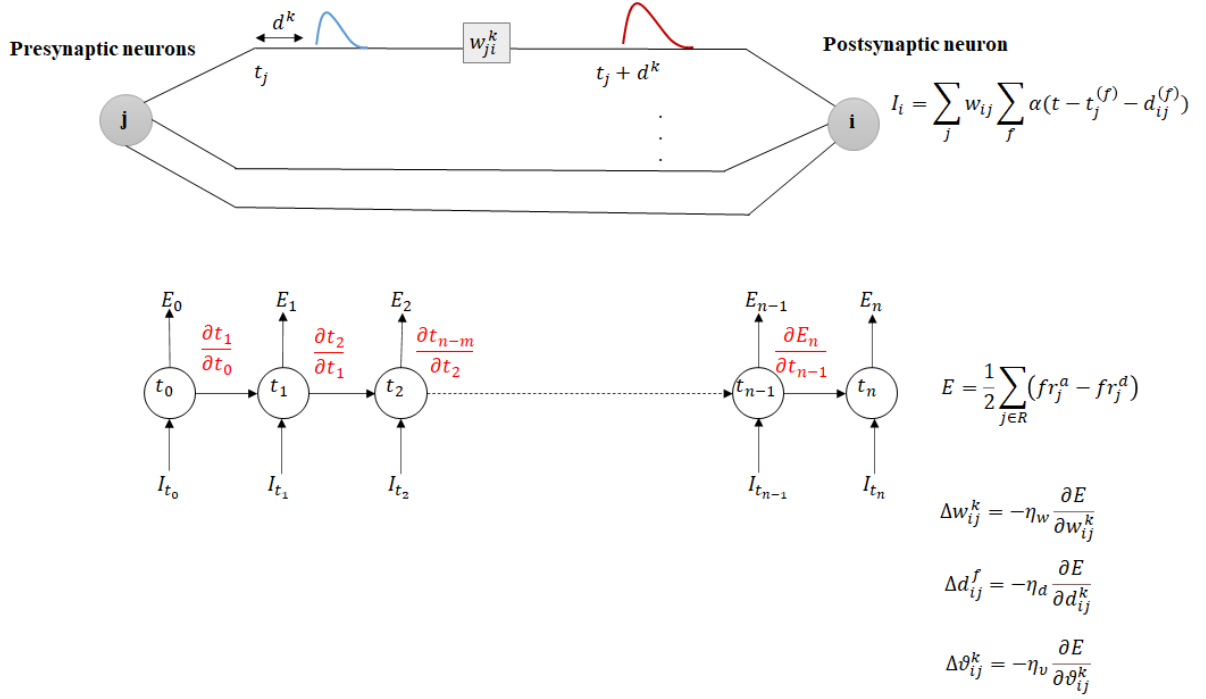


Figure 3-3: The error backpropagation schema using modified SpikeProp algorithm (Bohte, Kok, & Poutré, 2000)

Figure 3-3, illustrates the schematic description of the modified SpikeProp backpropagation algorithm used in the EPUSSS local learning mechanism. During the training phase using the proposed local error backpropagation, the STDP rule changes the synaptic weights of reservoir neurons until the observation time to enable the input neuron to capture the temporality behaviour in the input pattern. Afterwards, the backpropagation rule changes the synaptic efficacy, delay values, and firing threshold according to the error in the input neuron firing activity. This process continues until the sliding time-window reaches the end of the input pattern length. Another important fact in the EPUSSS model is that the inhibitory neurons are implemented by considering a negative sign for the synaptic connections between an inhibitory presynaptic neuron and a postsynaptic neuron. This important feature helps EPUSSS use the same STDP rule and error backpropagation mechanism for both inhibitory and excitatory connections. In other words, using (-1) to label inhibitory and $(+1)$ to label excitatory

connections allows to decrease and increase synaptic efficacy with the same rule which is another novel characteristic in the EPUSSS model.

3.5.2 EPUSSS Global Learning Mechanism

In the global learning mechanism, learning takes place on an episodic basis, meaning that at the end of each episode corresponding to the presentation of an input pattern to the network, a feedback signal is generated according to the success or failure of the target class detection. Thereupon, this reward signal which is analogous to the neuromodulatory signal in the brain contributes to synaptic weight adjustment at the end of each episode of learning (presenting the entire input pattern to the network).

In the EPUSSS global learning mechanism, a rewarded-modulated STDP approach similar to the model described by Legenstein et al. (2008) is used to link the higher-level supervised learning with the lower-level semi-supervised learning using a reinforcement signal. In this approach, at the end of each learning episode, a neuromodulatory signal according to the firing behaviour of the input neuron is produced by comparing the deSNN's output and the desired classification outcome. Then, this neuromodulatory signal is feedboxed into the STDP's weight change formula to adjust the synaptic weights proportional to the neuron contribution in the classification results.

In the EPUSSS model, the deSNN output layer follows the same procedure described in chapter 2 to create the output neurons. However, the learning rule for adjusting the synaptic efficacy is the R-STDP rule instead of the SDSP rule which is described in the following.

EPUSSS Global Learning

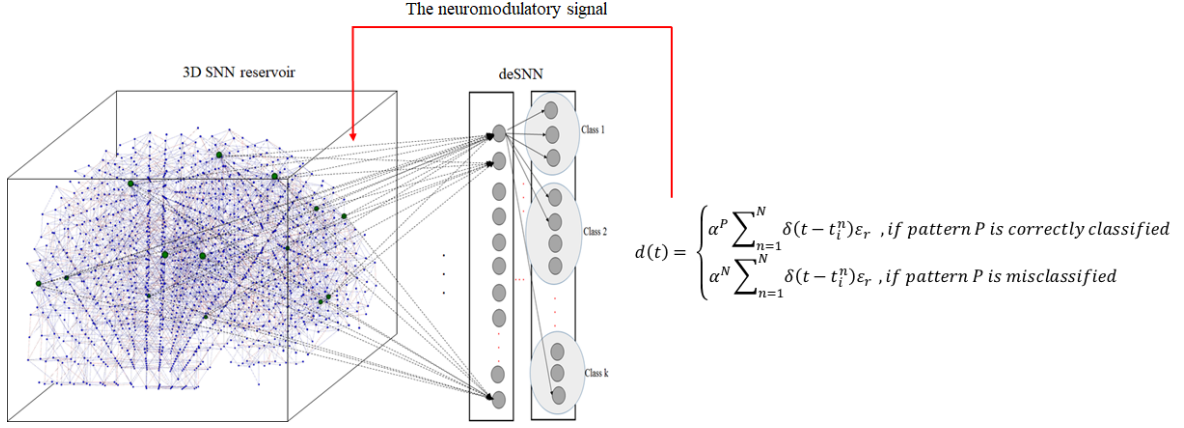


Figure 3-4: EPUSSS global learning mechanism using R-STDP rule

In the EPUSSS model, the deSNN output neuron can either have synaptic connections to only input neurons or both input neurons and reservoir neurons. The former approach is more desirable because the locally connected reservoir neurons can transfer the memory captured during the local learning to the input neurons thus by connecting only the input neurons to the deSNN output neurons, the performance of detection will be increased while the risk of carrying noise to the high-level information processing at classifier will be reduced (see Figure 3-4). All the connections from input neurons (reservoir neurons) to the deSNN output neurons are excitatory connections to mimic dopaminergic neurons' behaviour which are related to the novelty and reward prediction (Legenstein et al. 2008).

In order to learn through R-STDP to classify Spatio-temporal presynaptic firing activities to match with a specific input pattern, a reward signal $d(t)$ is generated at the time of classification outcome observation. This signal reinforces the synaptic efficacy of connections between excitatory input neurons and deSNN output neurons through rewards. To be more specific, neurons are rewarded for increasing and decreasing their synaptic efficacy based on their contribution to the classification outcome. This provides an asynchronous irregular firing behaviour in the network that resembles neuronal activity in the cortex.

The positive and negative weight changes in the pre- and postsynaptic weights between input neurons and deSNN output neurons are collected in an eligibility trace $c_{ij}(t)$. The eligibility trace is the moment that memory is created at the synapse site after a temporal delay between the sensory input and the moment of reward delivery. Then, the weight changes according to the R-STDP rule explained here for EPUSSS global learning can be formulated as follow:

$$\Delta w_{ji}(t) = c_{ji}(t)d(t) \quad (3-6)$$

Where w_{ji} is the weight of the synaptic connection between presynaptic input neuron i and postsynaptic output neuron j , $c_{ji}(t)$ is the eligibility trace of this synapse which collects weight changes proposed by STDP, and $d(t) = h(t) - \bar{h}$ is the reward resulted from a neuromodulatory signal $h(t)$ with mean value \bar{h} .

In the R-STDP rule, an eligibility kernel function $f_c(t)$, scales the contribution of pre- and postsynaptic neurons with the second spike at time 0 and to the eligibility trace at time t .

$$f_c(t) = \begin{cases} \frac{t}{\tau_e} e^{-\frac{t}{\tau_e}} & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3-7)$$

where the τ_e is a time constant set to 0.4. Then, the eligibility trace $c_{ji}(t)$ can be calculated as follow:

$$c_{ij}(t) = f_c(t) \sum_{f=1}^N \sum_{n=1}^N w_{ij}(t_i^n - t_j^f) \quad (3-8)$$

In Eq. (2-8), $f_c(t)$ is the eligibility kernel function, t_j^f is the time of presynaptic firing, t_i^n is the firing time of the postsynaptic neuron, and w_{ij} is the weight of the synaptic connection between presynaptic neuron j and postsynaptic neuron i that follows STDP rule formulated in Eq. (2-13).

The reward signal $d(t)$ generated at the end pattern presentation to output neuron is calculated using the following formula:

$$d(t) = \begin{cases} \alpha^P \sum_{n=1}^N \delta(t - t_i^n) \varepsilon_r & , \text{if pattern } P \text{ is correctly classified} \\ \alpha^N \sum_{n=1}^N \delta(t - t_i^n) \varepsilon_r & , \text{if pattern } P \text{ is misclassified} \end{cases} \quad (3-9)$$

Where, α^N and α^P are reward constant ($\alpha^N < 0 < \alpha^P$) and ε_r is the reward kernel calculated as follow:

$$\varepsilon_r(t) = A_r^+ \frac{t}{\tau_r^+} e^{1-\frac{t}{\tau_r^+}} - A_r^- \frac{t}{\tau_r^-} e^{1-\frac{t}{\tau_r^-}} \quad (3-10)$$

As it was stated before $d(t) = h(t) - \bar{h}(t)$ which is a signal with zero mean. To satisfy this constraint, the reward kernel ε should have a zero mean, i.e., $\bar{\varepsilon}_r = \int_0^\infty dr \varepsilon_r(r) = 0$. The reward kernel ε_r is the difference between the positive α function and the negative α function to satisfy the abovementioned condition. The parameters in the reward kernel according to Legenstein et al. (2008) are set as follow: $A_r^+ = 1.379$, $A_r^- = 0.27$, $\tau_r^+ = 0.2$, and $\tau_r^- = 1$.

As can be deduced from Eq. (3-9) and Eq. (3-10) the reward signal depends both on the input neuron spiking activity and the output neuron behaviour in pattern recognition. The reward signal both adjust the hypersynapses and synaptic connections between input neurons and deSNN output neurons. This mechanism allows the network to feedback the outcome of learning to the lower-level reservoir neuron to reinforce the local learning as well as global learning.

To avoid unbounded weights growth, the changes in the synaptic weights are limited between w_{min} and w_{max} . In the next section, a Lyapunov energy function is introduced as an extra

control mechanism over synaptic weight changes to guarantee life-long learning in the EPUSSS model without the risk of network saturation.

3.5.3 Lyapunov Energy Function

The artificial neural networks that use the Lyapunov function to govern their learning rule for either memory retrieval or performing classification tasks are based on Hopfield's symmetrically connectivity structure assumption, which facilitates analytical calculations of network behaviour (Kobuchi, 1991; Yerramalla et al., 2003; Lee et al., Stern and Shea-Brown, 2020).

However, spiking neural networks which are proven to be more realistic in comparison to the traditional artificial neural networks, have asymmetrically connectivity structures. Due to this characteristic, SNNs can perform synaptic plasticity to capture input dynamics which cannot be applied in a symmetrical structure or bidirectional connectivity.

In symmetric neural networks, the network energy landscape is constructed in a way that decreases with time by starting from an initial state and following a gradient path that falls into its minimum value (the equilibrium state). In the spiking neural networks which are asymmetric models, the original Hopfield energy function is not applicable since the synaptic plasticity makes following the trajectory evolution impossible (Yan et al., 2013).

The Lyapunov energy function introduced in EPUSSS is constructed from neuronal potentiation in relation to the outcome of learning for controlling synaptic plasticity which ensures EPUSSS life-long learning mechanisms. Therefore, the energy of the output neuron's excitations is used to control the network's activity instead of the synaptic plasticity of network connections which is like free energy. Free energy is the maximum amount of work that a thermodynamic system can perform in a process. An increase in the output neurons excitation

means that the network is performing classification by assigning input learnt patterns to an existing class or a new one continuously.

EPUSSS model is an open system that constantly exchanges energy and information with its environment. In the EPUSSS model, while the local and global learning attracts the system down to the state of cognition using internal neuronal activity and external environmental stimulus, the STDP rule is responsible for the asymmetric synaptic plasticity of the neuronal connectivity.

The overall dynamics of the EPUSSS model can be described using the free energy concept. The thermodynamical definition of system free energy can be used to characterise the EPUSSS model neuronal activity in terms of output neuron excitation level.

$$E = \frac{1}{2} \sum_k \sum_j w_{kj} \sum_f a(t - t_k^f) \quad (3-11)$$

In Eq. (3-11), E is the Lyapunov energy function for calculating the deSNN output neurons activity, k is the output neuron connected to input neurons through synaptic connections with w_{kj} efficacy, and $a(t - t_k^f)$ is a time course that within a postsynaptic output neuron receives presynaptic spikes from input neurons.

$$\Delta E = E(t) - E(t - 1) \quad (3-12)$$

At the end of each episode of learning, the change in the model's energy is calculated using Eq. (3-12) and by introducing a parameter γ the synaptic weights of the network are adjusted to ensure the life-long learning concept in EPUSSS.

$$\gamma = \begin{cases} \zeta & \text{if } \Delta E > 0 \\ 0 & \text{if } \Delta E \leq 0 \end{cases} \quad (3-13)$$

If the changes in the model energy are positive, it means that the excitation level of the output neurons is increasing, then all the synaptic weights in the EPUSSS model are decreased by the amount of ζ , otherwise, the synaptic weights will remain unchanged.

This strategy helps EPUSSS to learn continuously from input patterns without going to a saturation phase.

3.6 The Proposed EPUSSS Algorithm

EPUSSS is a predictive model that can perform prediction and classification by learning Spatio-temporal information hidden in the input pattern. In the EPUSSS model, learning happens both at the microscopic level using synaptic plasticity and at the macroscopic level using neuromodulatory signals. As it was explained in detail in the previous section, this novel powerful computational brain-like model benefits from a broad range of biological learning characteristics and computational features. In this section, the pseudocode of the EPUSSS algorithm is provided.

Evolving Predictive Unsupervised Supervised deep learning model for Spike Streams (EPUSSS)

- 1: Generate 3D-SNN network including input neurons I , Reservoir neurons R using neurons coordinates (e.g., Talairach map for EEG data).
Create EPUSSS network connectivity by establishing hypersynapses and synaptic connections
- 2: amongst input neurons and between input neurons and reservoir neurons, respectively using small world connectivity algorithm (see chapter 2).
- 3: Set 80% of synaptic connections excitatory (+1) and 20% of synaptic connections inhibitory (-1).
Set neuron firing thresholds ϑ for excitatory and inhibitory neurons (if at least one of the
- 4: postsynaptic connections is inhibitory the neuron is considered as inhibitory neuron).
- 5: Set 50% of excitatory connection to AMPAR with delay value 0 and 50% excitatory connection to NMDAR with random delay value.

6: Set 50% of inhibitory connection to GABAAR with delay value 0 and 50% inhibitory connection to GABABR with random delay value.

7: Initialise random weights for synaptic and hypersynaptic connections.

8: Set *time – window* value (e.g., 10 time units) and *max _iteration* value (e.g., 50)

9: Encode all input pattern to spike train using BSA algorithm (see chapter 2).

10: while *it* ≤ *max _iteration*

11: for each input pattern P_s

12: for each input neuron I_i

13: while $1 \leq \text{length}(\text{time} - \text{window})$

14: Propagate input spikes to reservoir neurons using LIF neuron model (Eq. 3-1)

15: Adapt synaptic and hypersynaptic weights according to STDP rule (Eq. 2-12)

16: Calculate E_{local_i} (Eq. 3-2).

17: Update weights, delays, and firing thresholds using Eq. 3-3, Eq. 3-4, and Eq. 3-5.

18: Set the synaptic weights for inactive neurons to zero.

19: $\text{time} - \text{window} += \text{time} - \text{window}$

20: Initialise deSNN parameters: *Mod*, *D*, *C*, α , and *sim* (similarity threshold)

21: for each input pattern P_i

22: Create a new output neuron *i*

23: Initialise synaptic weights $w_i(0)$ using RO learning rule denoted in Eq. 2-14

24: Calculate PSP_{imax} using Eq. 2-16

25: Calculate the firing threshold Th_i for neuron *i* using Eq. 2-17

26: if the new wight vector w_i is similar to an existing output neuron using Euclidean distance similarity measure and *sim* threshold

27: if the desired class labels of the similar weight vectors are different

28: Generate the corresponding $d(t)$ reward signal according to Eq. 3-9.

29: Update the synaptic weights w_i for successive spikes on the corresponding synapses using R-STDP rule denoted in Eq. 3-6.

30: Add new output neuron to the network

```

31:         else
32:             Generate the corresponding  $d(t)$  reward signal according to Eq. 3-9.
33:             Update the synaptic weights  $w_i$  for successive spikes on the corresponding synapses
34:             using R-STDP rule denoted in Eq. 3-6.
35:             Merge the two neurons by averaging their threshold and synaptic weights
36:             Calculate energy function  $E$  (Eq. 3-11)
37:             Adapt all synaptic weights according to Eq. 3-13.

```

Figure 3-5: The EPUSSS learning algorithm pseudocode

In the implementation phase of the EPUSSS model, to avoid conflict between the input spike train and the processed spike in the next time point, a twin neuron for each input neuron is created. These twin neurons have the same synaptic connections as their former counterparts (input neurons) but with the reverse direction from the reservoir to twin neurons.

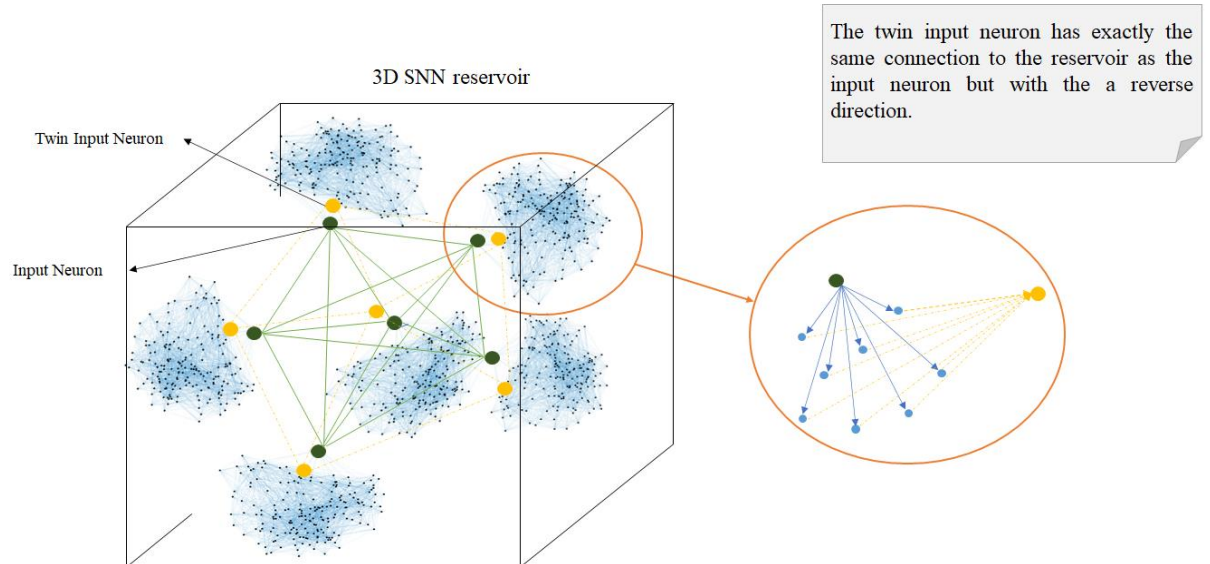


Figure 3-6: The figure shows twin input neurons that each represent one input/output neuron.

3.6.1 EPUSSS Computational Complexity

Assume that there are N neurons, including input neurons (I_n), reservoir neurons (R_n), and output neurons (O_n) with M synaptic connections in the EPUSSS model. P_T input patterns with the length of T are presented to the network during local and global learning. According to Maass (1994), if the computational complexity for each synaptic weight w_{ij} is $O(1)$ and each input pattern $\log P_T$, then the by considering K epoch of training in the local error backpropagation process, the computational complexity of EPUSSS model is $O(\log P_T((K \times I_n \times R_n) + (I_n \times O_n)))$.

3.7 Implementation and Experimental Analysis

In this section, the EPUSSS implementation details are explained, and the performance of its learning mechanisms are explored by using Wrist Movement EEG data (Taylor et al., 2014) as a testbed. All the simulations and experiments were conducted in MATLAB R2020a and in an Intel Core-i7 - 2.80GHz processor.

3.7.1 The EPUSSS Framework Implementation

The EPUSSS framework implementation consists of several modules, including network, connectivity, encoding, neuron models, unsupervised learning, supervised learning, cross-validation, and visualisation. The EPUSSS framework structure can be seen in detail in Figure 3-7.

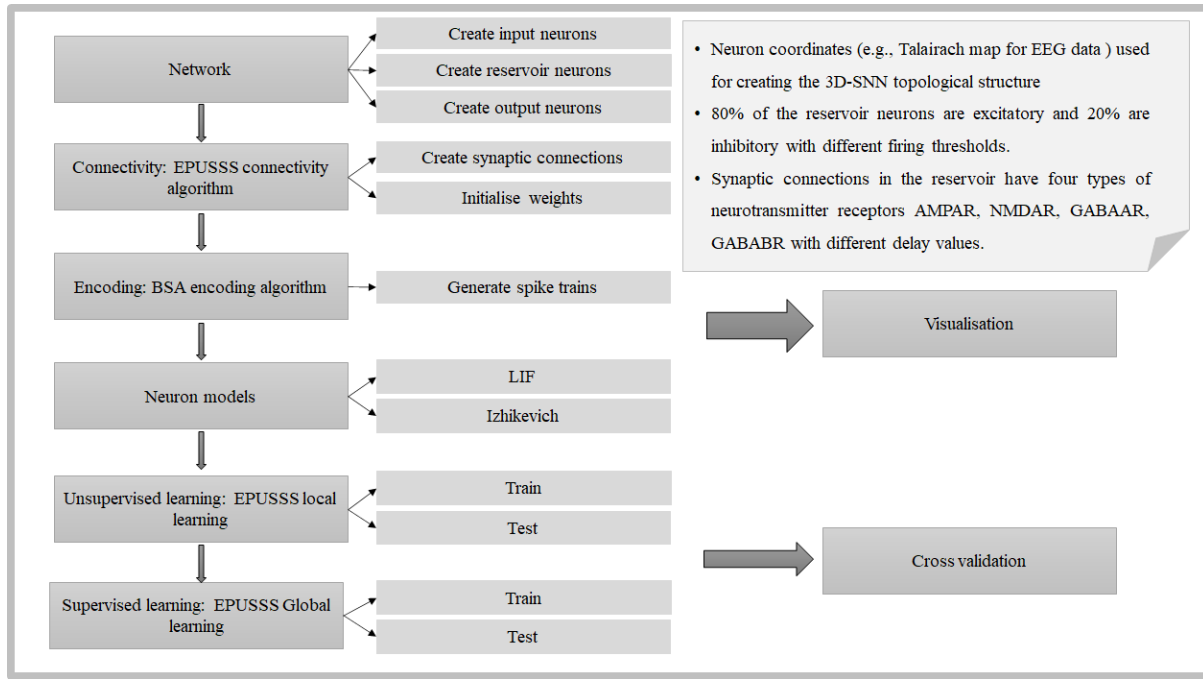


Figure 3-7: EPUSSS framework from implementation point of view

The EPUSSS network module creates the 3D-SNN topological structure using Talairach map neuron coordinates (or any arbitrary coordinates) for the reservoir neurons, EEG input channel coordinates (or any arbitrary coordinates) for the input neurons, and arbitrary coordinates for the output neurons to create a brain-like 3D structure for the SNN. In the connectivity module, the EPUSSS connectivity algorithm establishes synaptic connections between input neurons and reservoir neurons and hypersynaptic connections amongst input neurons. Furthermore, in this module, the synaptic types, neurons firing thresholds, four types of neurotransmitter receptors AMPAR, NMDAR, GABAAR, GABABR with their corresponding delay values, and the initial synaptic and hypersynaptic weights are defined and assigned to the EPUSSS model. 80 per cent of the synaptic connections are labelled as excitatory and 20 per cent inhibitory. If at least one of the postsynaptic connections of a reservoir neuron is inhibitory, then that neuron is considered as an inhibitory neuron with an inhibitory firing threshold. Excitatory neurons have an excitatory firing threshold that should always be greater than

inhibitory neurons' firing threshold which allows them to fire at a higher rate. Finally, the initial synaptic and hypersynaptic weights are assigned to the neuronal connections.

The Encoding module converts input signals to spike trains using the BSA encoding algorithm. In the neuron model module, LIF and Izhikevich neuron models are implemented and adapted to the EPUSSS local and global learning algorithms. Unsupervised and supervised modules are the local and global learning implementation of the EPUSSS model, respectively. The cross-validation module consists of leave-one-out, leave-many-out, and k-fold cross-validation methods, and a confusion matrix for evaluating the model performance. Eventually, the visualisation module presents a visual graph demonstration for the EPUSSS model for a better understanding of the proposed model behaviour.

3.7.2 Experimental Analysis

To explore the performance of the EPUSSS model, the Wrist Movement dataset (Taylor et al., 2014) is used as a testbed. The hypothesis behind using this dataset was to explore the NeuCube capabilities for developing a brain-computer interface (BCI) system to assist in the rehabilitation of complex upper limb movements (Taylor et al., 2014). The dataset is described in more detail in the following section.

Here I use the Wrist Movement dataset. It contains EEG recordings of three healthy volunteers from KEDRI¹ with no history of neurological disorders and all right-handed. The participants were requested to perform the specified Wrist Movements or imagine the movements or remain at rest. The Wrist Movement task includes three stages, resting, flexing the wrist, and extending the wrist all with closed eyes to reduce visual and blink related artefacts. Participants were asked to perform the movement intention task starting from a mid-pronation with the forearm

¹ Knowledge Engineering and Discovery Research Institute – AUT, <https://www.kedri.aut.ac.nz>

resting on their lap and either imaging or performing the demanded movements each in 2 seconds and repeating it 10 times. The Emotiv Epoc EEG Neuroheadset with 14 channels based on International 10-20 locations (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) was used to record EEG data. Also, two additional electrodes (P3, P4) were used as reference. The recorded data were digitized at a 128 Hz sampling rate with no filtering applied to the data, neither online nor offline.

The dataset includes 60 recordings with 14 input channels that each of these recordings were labelled as one of the three joint classes flexion, extension, and rest of real and imagined movements of the hand. The dataset is perfectly balanced since Each class label has equally 20 samples. The input electrode position and the Talairach map coordinates were used to create the input and reservoir neurons position in the 3D-SNN topological structure in EPUSSS (see Figure 3-8). Table 3-1 shows the EPUSSS model parameter settings. As it was mentioned before using the Talairach map coordinates 1485 reservoir neurons were created in the EPUSSS model and according to the Emotiv Epoc EEG Neuroheadset electrodes' position, input neuron coordinates were mapped to the 3D-SNN structure. The output neurons were evolved through the learning process and arbitrary coordinates were assigned to them. The EEG recordings through 14 input neurons (the same as EEG channels) were fed to the model and the results of simulations were plotted in the figures below.

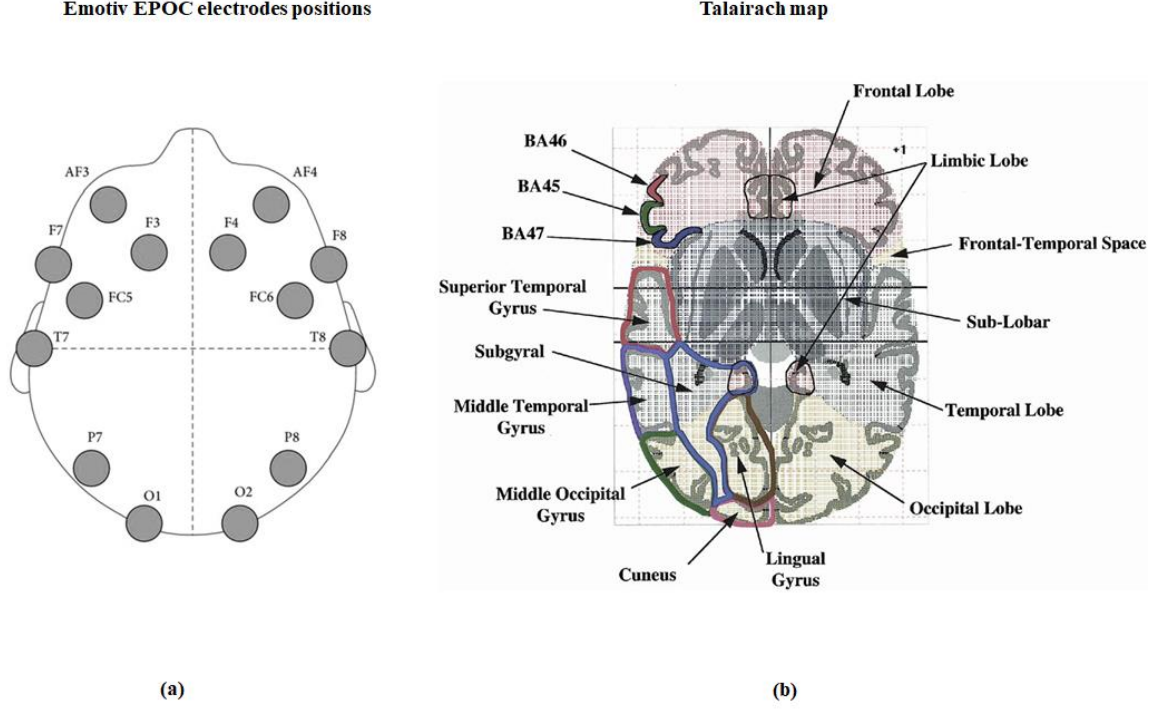
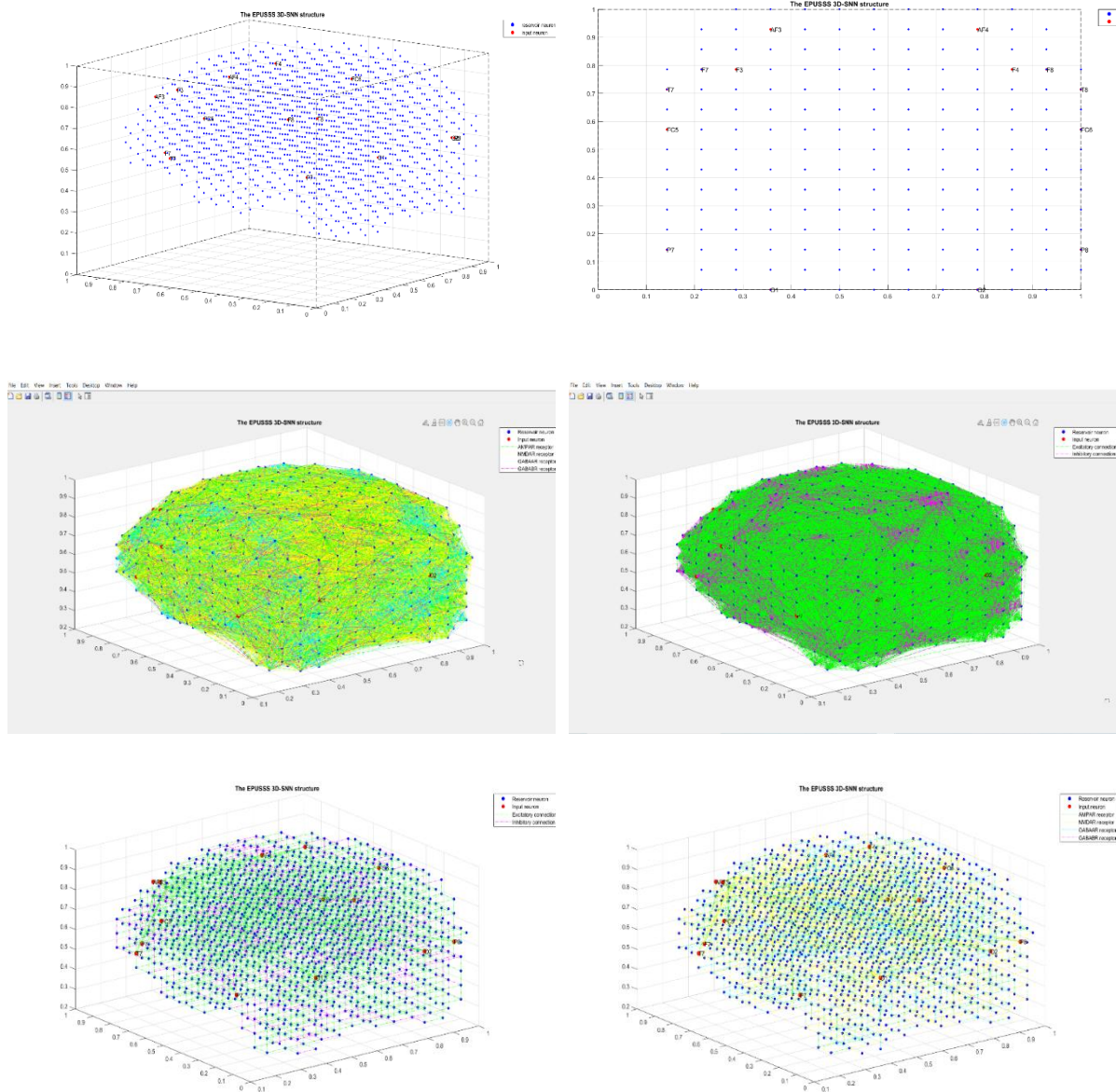


Figure 3-8: Input neurons and reservoir neurons positions according to (a) wireless Emotiv EPOC EEG Neuroheadset (Blanco and Ramirez, 2019) and (b) Talairach map (Lancaster, et al., 2000).

Table 3-1: The EPUSSS model parameter settings

Algorithm	Parameters Settings
BSA	$threshold = 0.9550$
LIF	$\vartheta_I = 0.5, \vartheta_E = 0.2$
STDP	$A_+ = 0.001, A_- = -0.001, \tau_+ = 10, \tau_- = 10$
Local Learning	$d_{AMPA} = 0, d_{NMDAR} \sim N(0,1), d_{GABAAR} = 0, d_{GABABR} \sim N(0,1),$
	$\eta_w = 0.0075, \eta_d = 0.0075, \eta_\theta = 0.0075, time - window = 10,$
	$max_iteration = 50$
Global Learning	$\alpha = 0.5, mod = 0.8, C = 0.55, \tau_e = 0.4, \alpha^N = -1.4, \alpha^P = 1.4,$
	$A_r^+ = 1.379, A_r^- = 0.27, \tau_r^+ = 0.2, \tau_r^- = 1, \zeta = 0.05, sim = 0.25$

Figure 3-9 represents some steps taken in one experiment: encoding the data; mapping; downsampling the mapped coordinates to reduce computational time complexity.



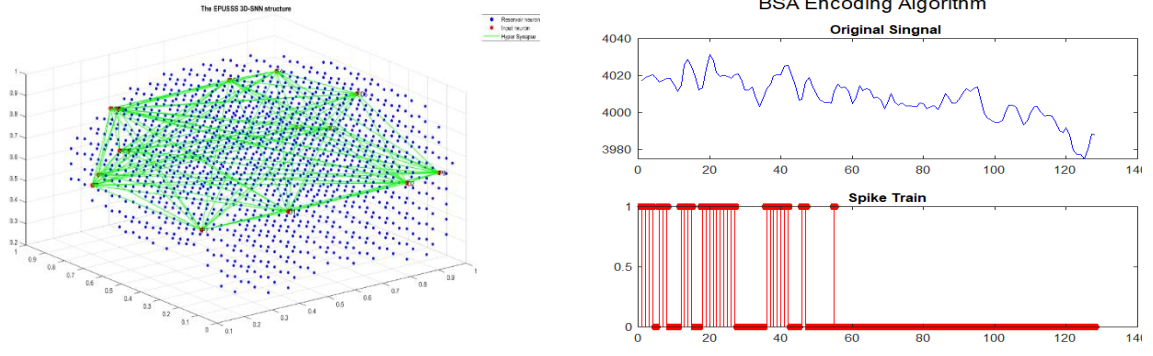


Figure 3-9: The steps of mapping brain coordinates, creating the EPUSSS network with excitatory/inhibitory neurons, 4 types of neurotransmitter receivers with different synaptic delay and functionality in propagating information, the hypersynaptic connections, and encoding the input data into spike trains for processing by the EPUSSS two-level hierarchical learning model.

Due to the high computational complexity of the EPUSSS model, in the experiments, the brain coordinates are downsampled by 8 times to 186 reservoir neurons to improve the processing speed.

The above experiment reflects projecting the input space into a 4-dimensional space (spatiotemporal projection) by EPUSSS to extract hidden correlation amongst input data and then use that extracted information for prediction and classification.

The twin input neurons are connected to the deSNN output layer in the EPUSSS model to interact with the upper-level learning in the hierarchical learning structure of the EPUSSS framework. In this way, the learnt neuronal behaviour in a semi-supervised manner will transfer to the higher level to empower the classification results. Then the neuromodulatory signal from the outcome of the classification at the end of each episode of the training is used to correct the connections between the local and global learning hierarchy (Figure 3-10).

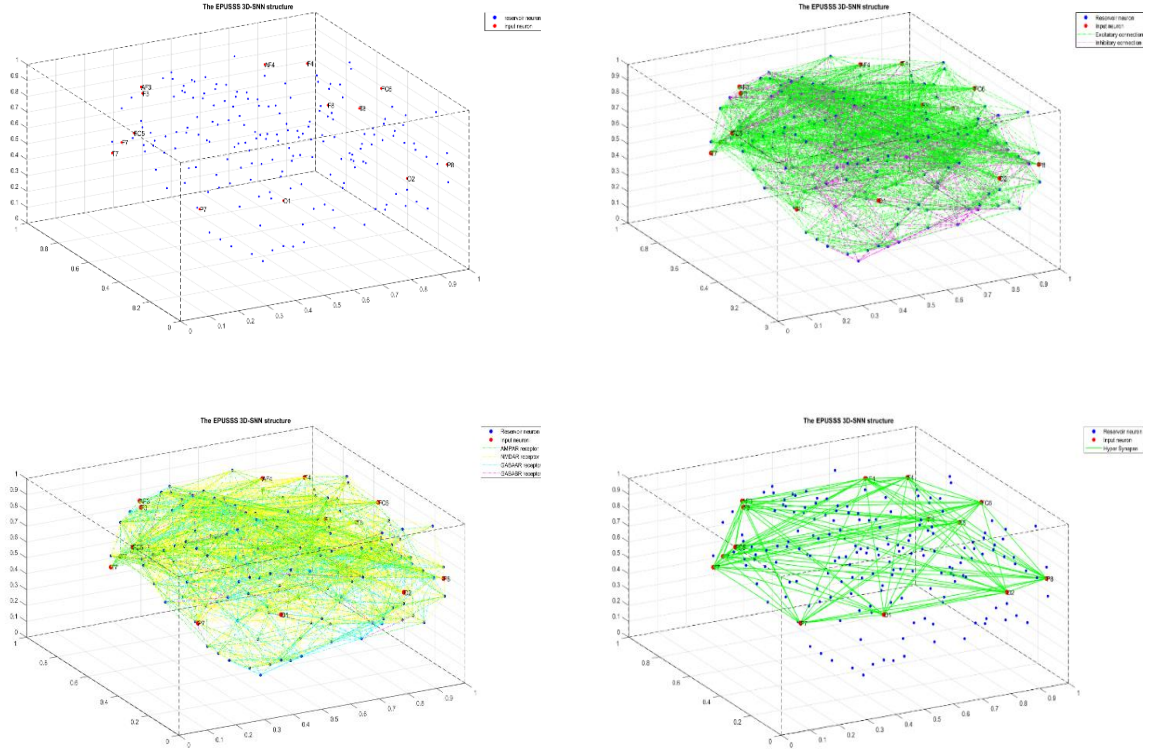
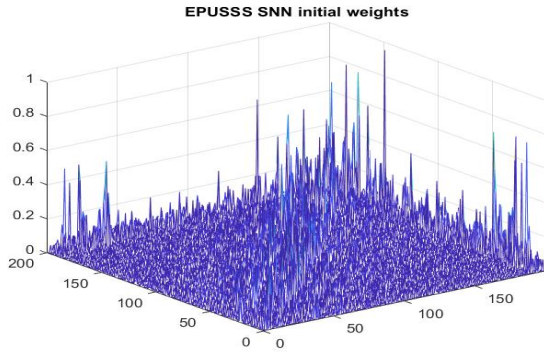


Figure 3-10: The downsampled coordinates to improve computational time complexity in the EPUSSS framework.

The EPUSSS framework has many neurons and a complicated connectivity structure that can increase the computation time polynomially. This limitation of the EPUSSS model will be addressed in chapter 5 by introducing a quantum-inspired SNN model to reduce the computation time.

a)



b)

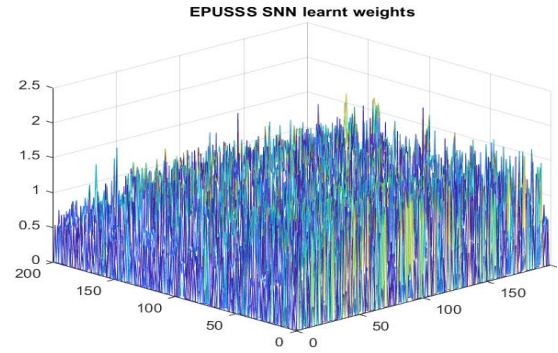
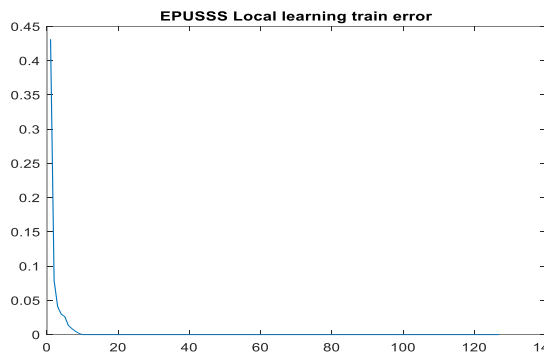


Figure 3-11: EPUSSS SNN weights distributions: a) initial weights, b) learnt weights

The Gaussian distribution in the learnt weights confirms learning fulfilled in EPUSSS local learning.

a)



b)

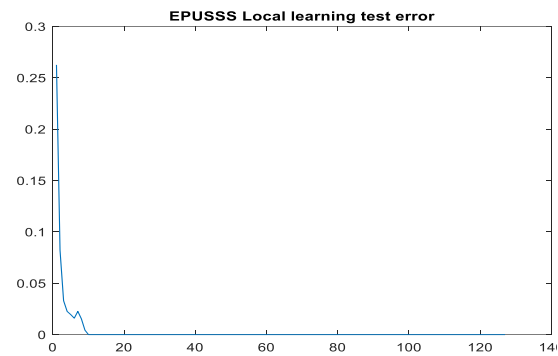
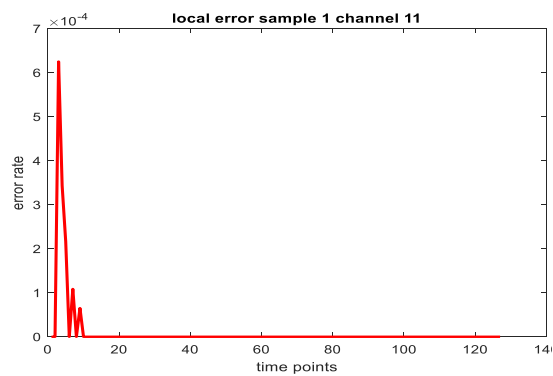
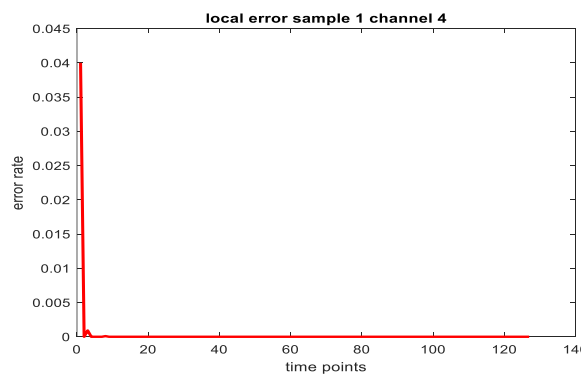
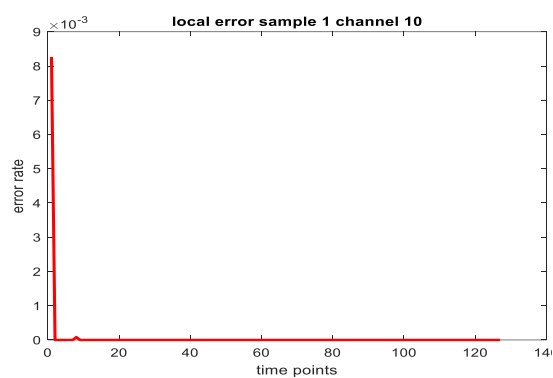
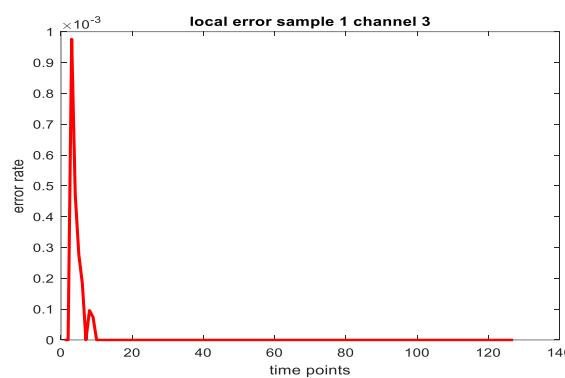
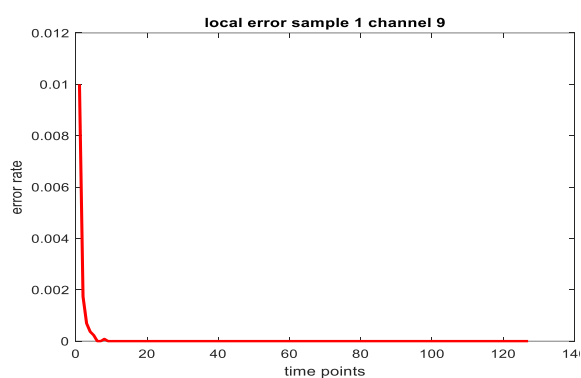
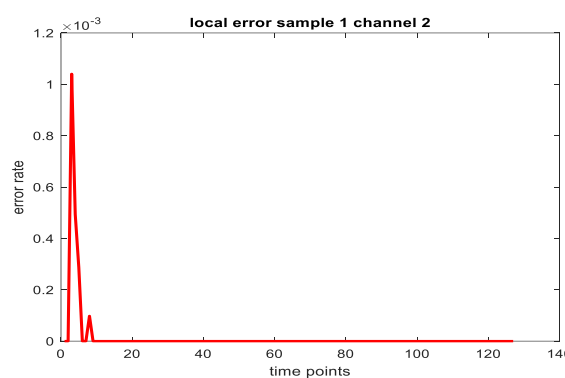
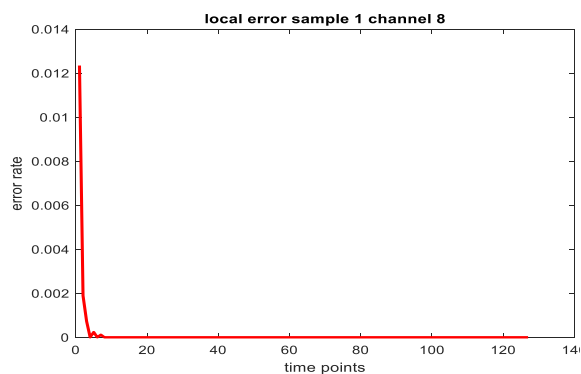
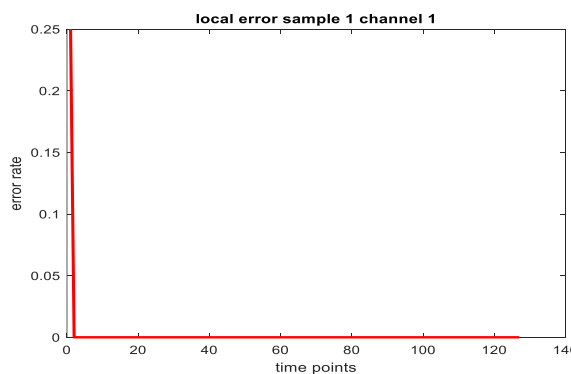


Figure 3-12: EPUSSS local learning errors convergence, a) training error, b) test error

In Figure 3-12, the average local learning of the EPUSSS model in the training and testing phase can be seen. Each plot shows that the error backpropagation mechanism in the EPUSSS local learning mechanism decreases the error rate between the desired and actual firing rates.



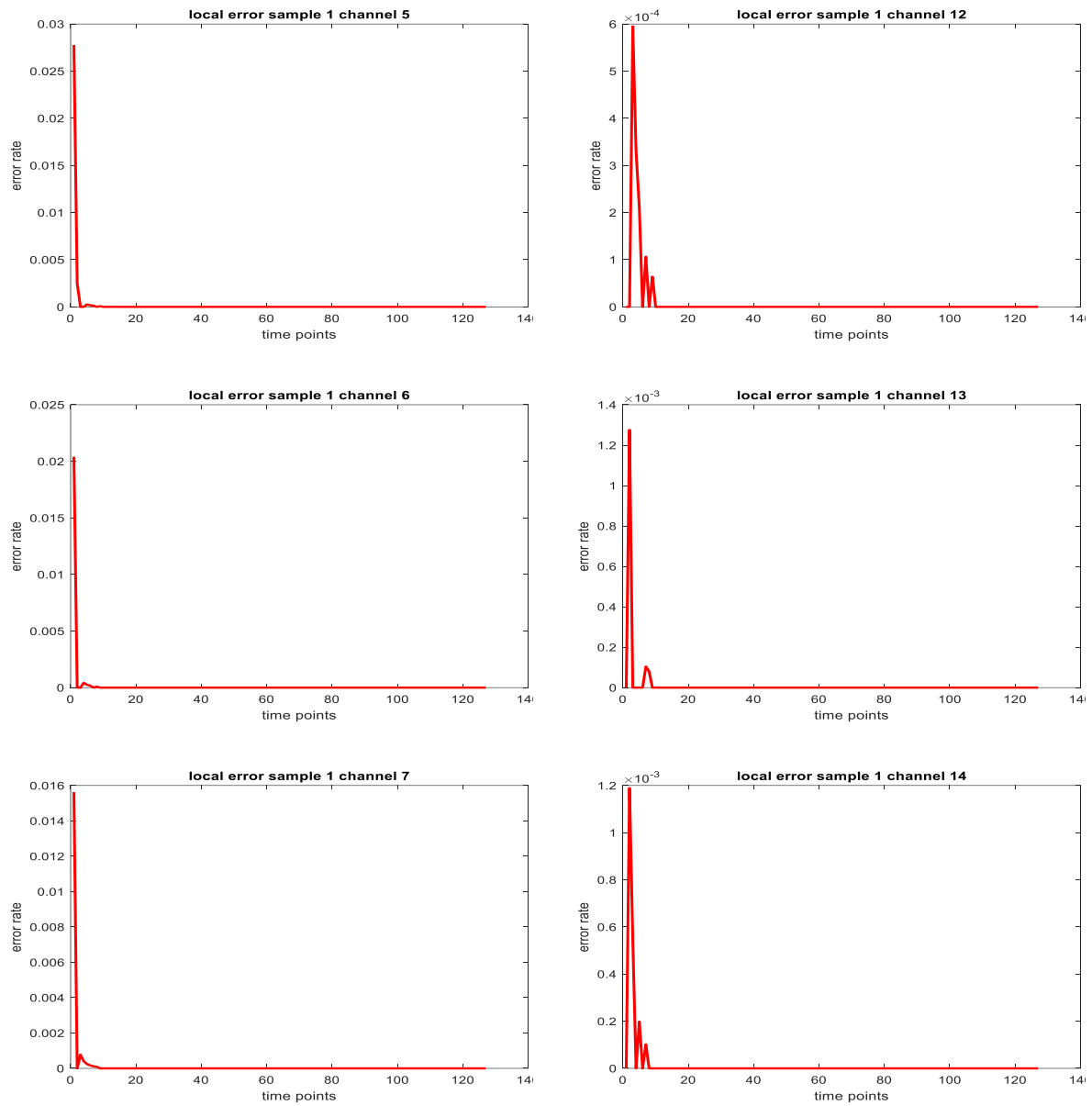


Figure 3-13: EPUSSS local error for each input in training phase for sample 1

Figure 3-13 illustrates the EPUSSS local error of all 14 input channels in the training phase for sample 1. The average local error for all channels for this sample can be seen in Figure 3-14.

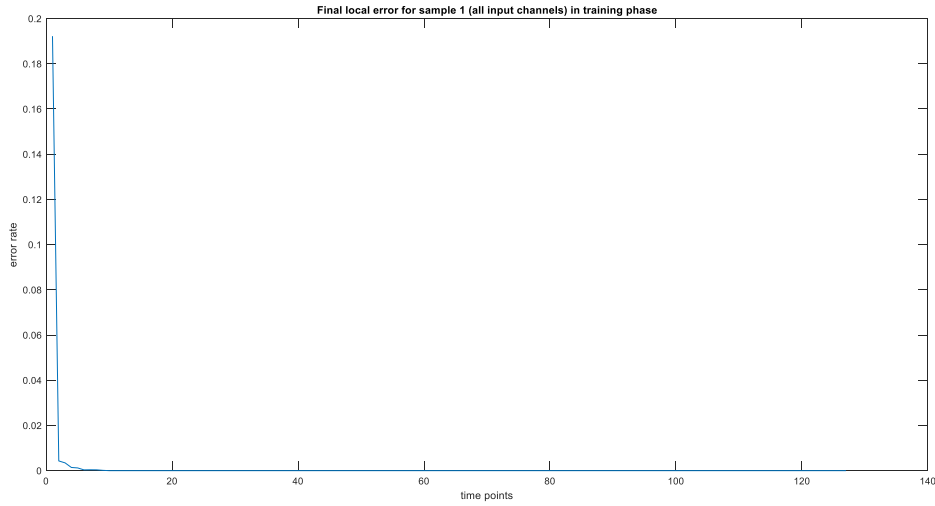
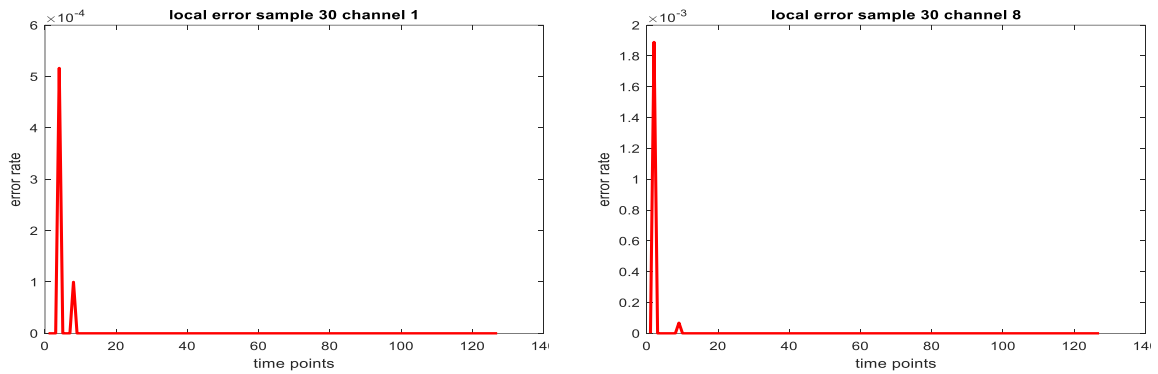
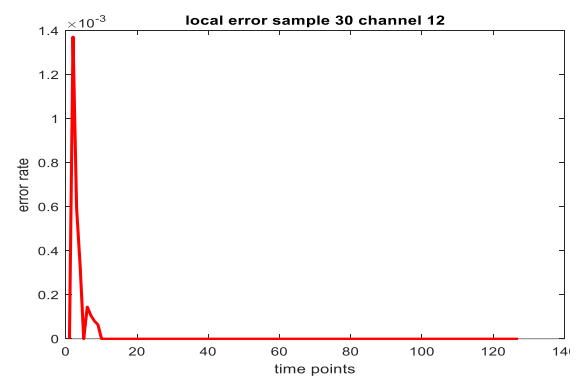
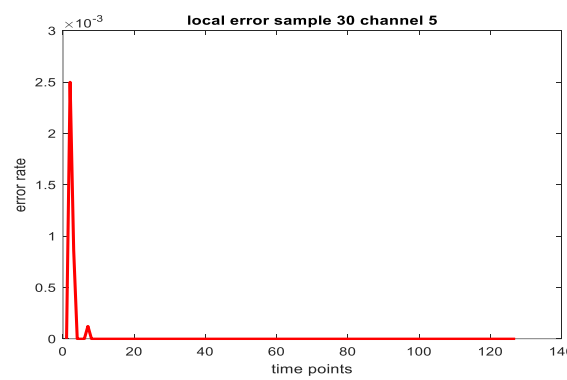
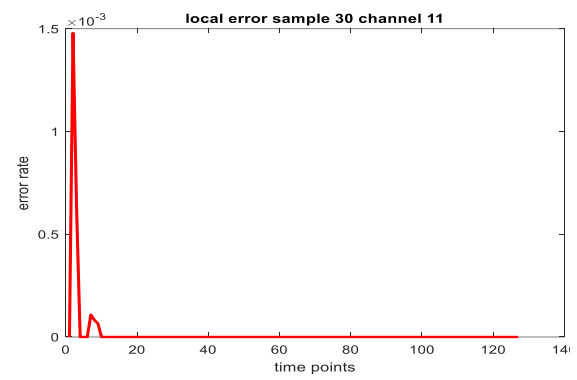
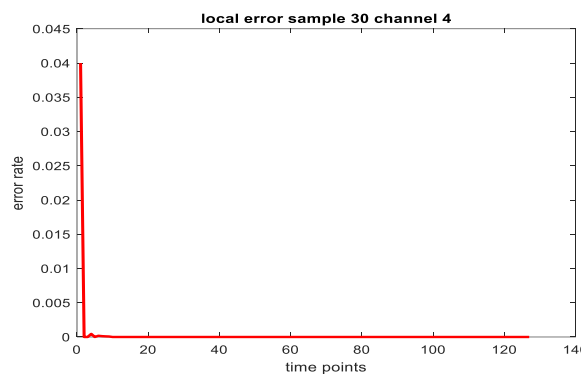
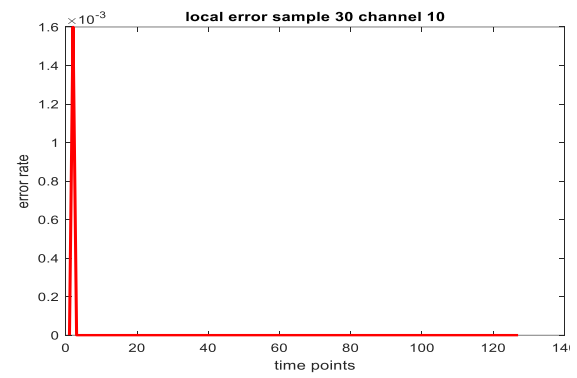
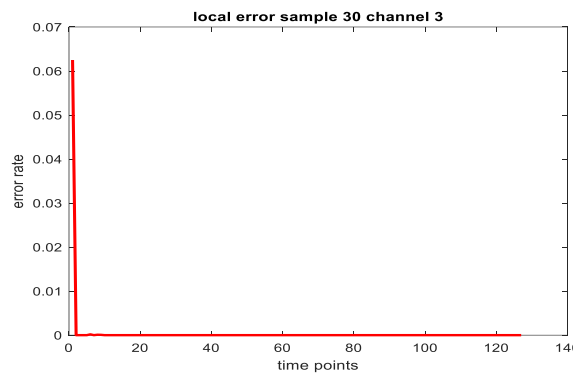
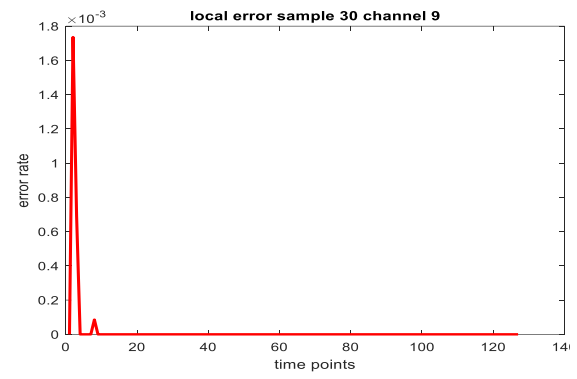
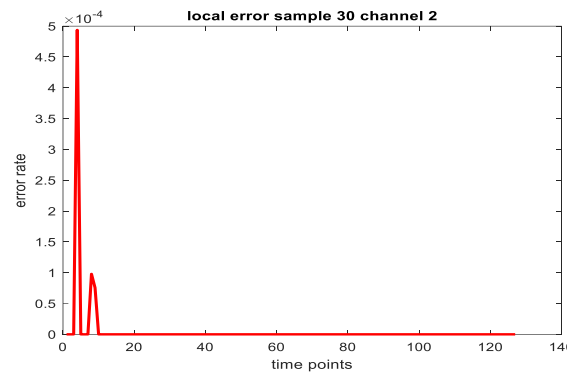


Figure 3-14: EPUSSS final local error for sample 1 (all input channels) in the training phase

In Figure 3-15, the EPUSSS local error of all 14 input channels in the testing phase for sample 30 is shown. The average local error for all channels for this sample can be seen in Figure 3-16.





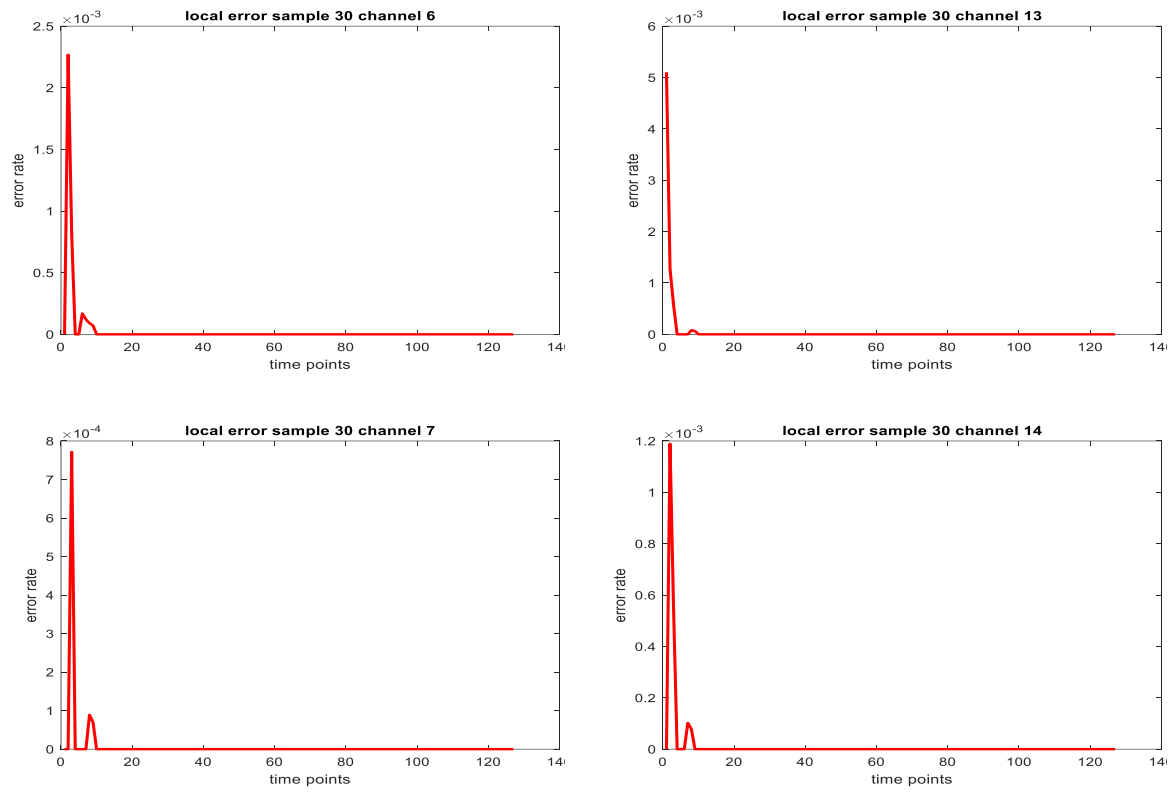


Figure 3-15: EPUSSS local error for each input in testing phase for sample 30

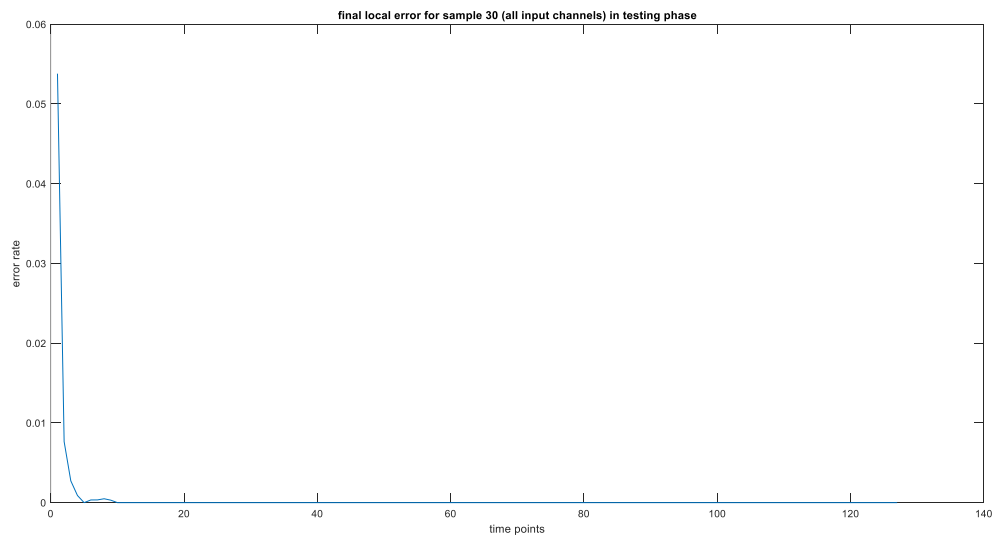
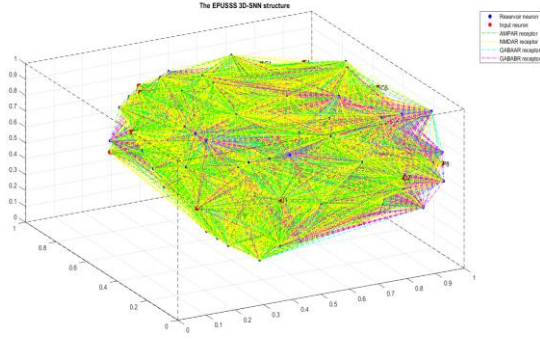


Figure 3-16: EPUSSS final local error for sample 30 (all input channels) in the testing phase

a)



b)

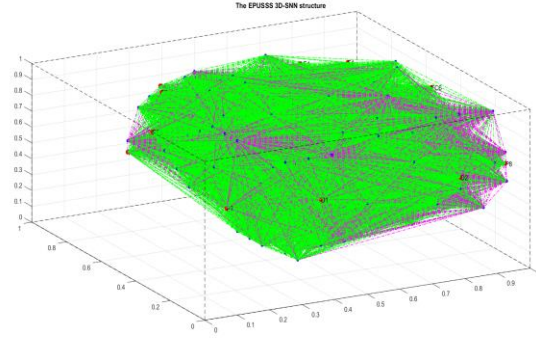


Figure 3-17: EPUSSS SNN connectivity with learnt weight, a) four types of synaptic connections, b) excitatory and inhibitory connections.

Figure 3-18 shows the neuromodulatory signals generated for global learning training phase. The results of classification of the Wrist Movement dataset using EPUSSS global learning can be seen in Table 3-2 and Table 3-3.

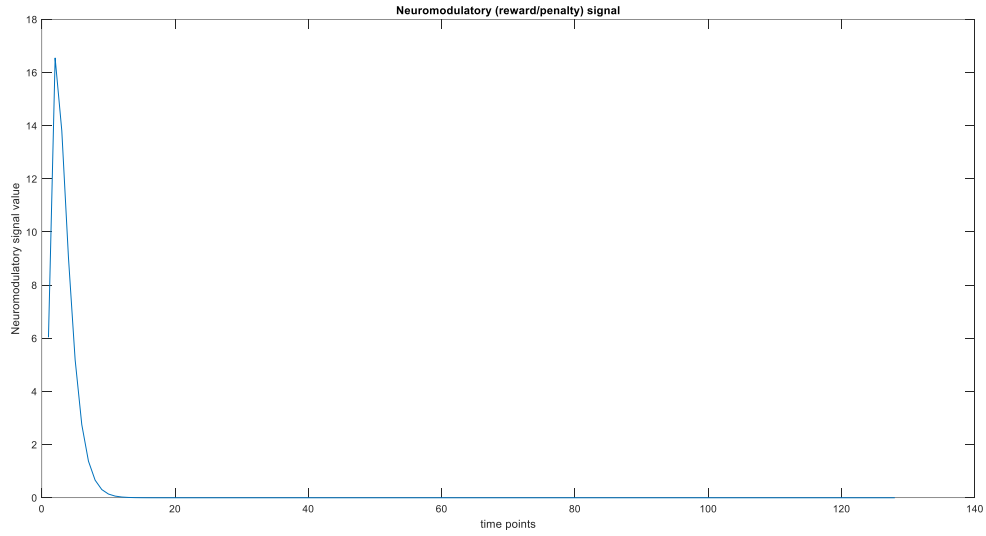


Figure 3-18: Neuromodulatory (reward/penalty) signal in the EPUSSS global training

In this thesis, NeuCube and LSTM models have been used for the comparative analysis. The same parameter setting as EPUSSS was applied for BSA, STDP, and deSNN models in NeuCube to have a fair comparison. Moreover, the LSTM model was designed with 5 layers including a sequence input with 14 dimensions, a BiLSTM layer with 100 hidden units, and

three fully connected layers. The SoftMax model was used as the activation function and the cross-entropy loss was used for measuring the classification error.

The comparative results of applying NeuCube and LSTM on the Wrist Movement dataset can also be found in Table 3-2 and Table 3-3.

Table 3-2: The comparative analysis of Global Learning Classification Results

	Classification	Accuracy	Precision	Recall	F-score
EPUSSS	Class 1	81%	0.80	0.72	0.76
	Class 2	73%	0.65	0.65	0.65
	Class 3	78%	0.55	0.73	0.62
NeuCube	Class 1	66%	0.55	0.50	0.52
	Class 2	61%	0.66	0.44	0.53
	Class 3	71%	0.55	0.57	0.56
LSTM	Class 1	48%	0.30	0.24	0.21
	Class 2	38%	0.10	0.09	0.9
	Class 3	56%	0.20	0.28	0.22

Table 3-3: The comparative analysis of Global Learning Classification Results Confusion Matrix

	Confusion Matrix	Class 1	Class 2	Class 3
EPUSSS	Class 1	16	3	1
	Class 2	4	13	3
	Class 3	3	6	11
NeuCube	Class 1	11	6	3
	Class 2	7	8	5
	Class 3	4	5	11
LSTM	Class 1	6	10	4
	Class 2	12	2	6
	Class 3	7	11	4

NeuCube SNN network's learning behaviour and LSTM training plot on the hand Wrist Movement can be seen in Figure 3-19 to Figure 3-21.

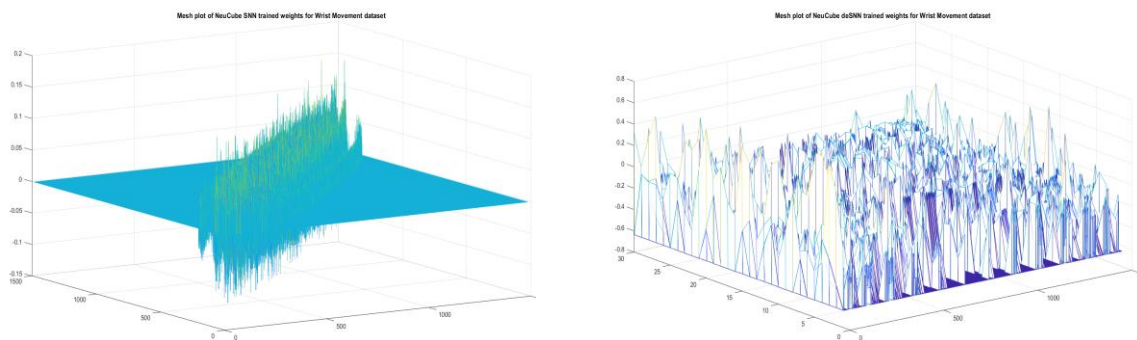
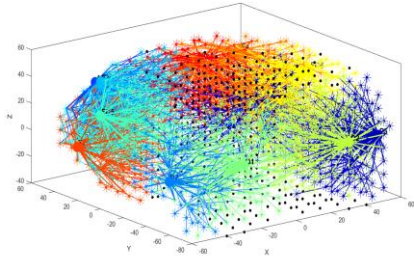
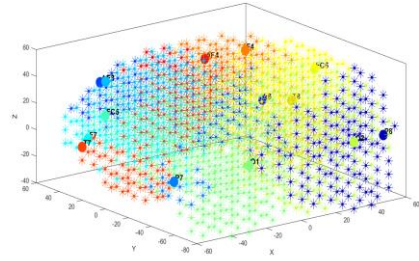


Figure 3-19: NeuCube SNN learnt weight distribution for Wrist Movement dataset

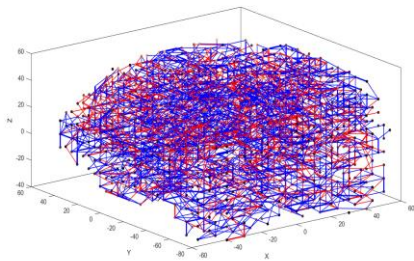
a)



b)



c)



d)

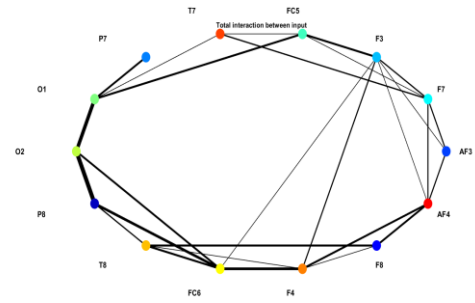


Figure 3-20: NeuCube SNN spiking behaviour for Wrist Movement dataset: a) spiking behaviour of input channels connected to reservoir, b) reservoir neurons regions according to their connectivity to EEG input channels c) excitatory and inhibitory connections, d) EEG channels correlations.

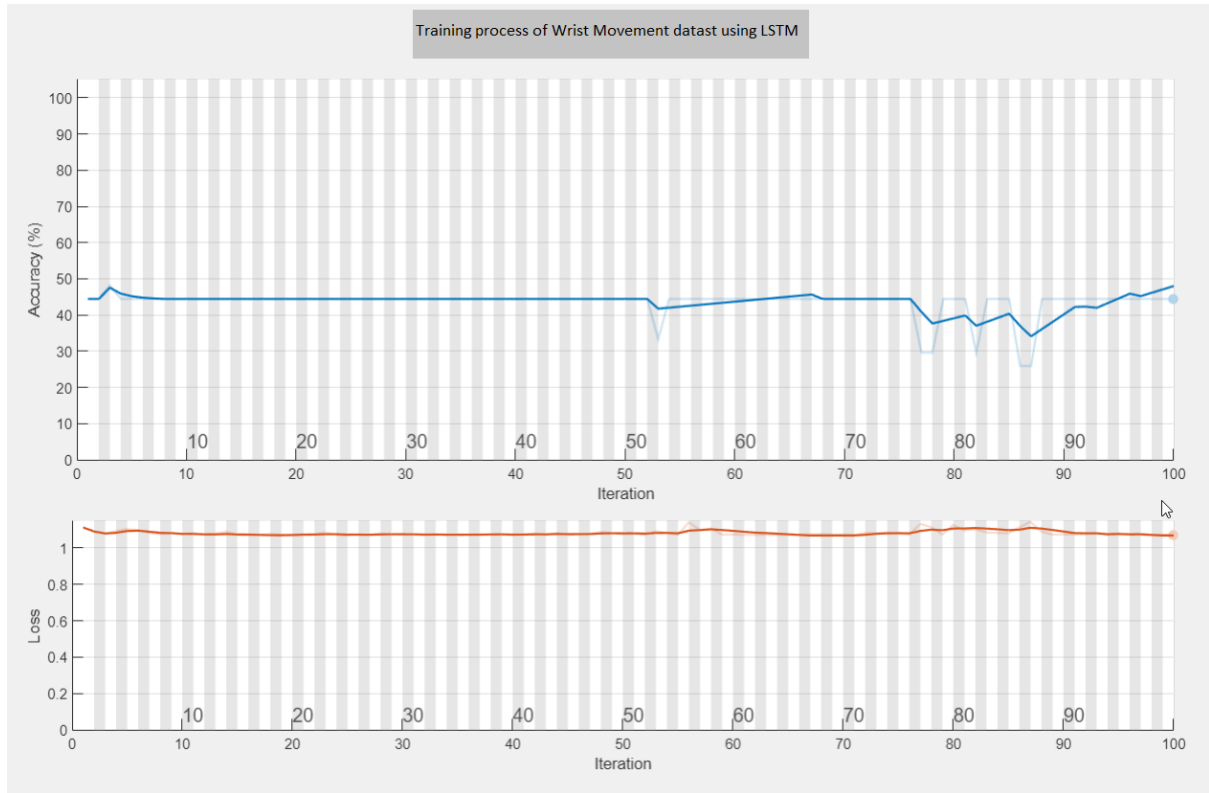


Figure 3-21: LSTM training accuracy and error plot for Wrist Movement dataset

In the above experiments, the EPUSSS model outperformed LSTM and NeuCube in the classification of the Wrist Movement dataset. Also, EPUSSS good prediction capability was demonstrated by the error convergence plots provided above. As it can be seen both in the numerical results and training plot, the LSTM model showed a poor performance in the classification of the Wrist Movement dataset. In contrast, EPUSSS and NeuCube demonstrate an acceptable learning outcome. This happened because EPUSSS and NeuCube can learn from events, and both have structure to capture spatiotemporally in the input data.

The above results prove that the spatiotemporally of EEG data cannot be captured by conventional deep learning models like LSTM but spiking neural networks with the capability of capturing both temporality and spatiality can outperform their traditional learning models counterparts.

Despite EPUSSS acceptable performance in prediction and classification of the Wrist Movement EEG data, due to a large number of neurons, complicated connectivity structure, and the iterative process of local and global learning, the computation time of EPUSSS increases polynomially by the size of the problem. This limitation of the EPUSSS model will be addressed in chapter 5 by introducing a quantum inspired SNN model to reduce the computation time. Another aspect of the EPUSSS framework that needs to be explored more is the encoding algorithm. Currently, the BSA algorithm has been used to encode the input data to spike train. Although BSA has been proved to be less sensitive to changes in the filter and threshold and as a result provides a more accurate spike train even for bipolar cases, the nature of the problem should be considered to choose a proper encoding algorithm.

3.8 Chapter Summary

This chapter presents an Evolving Predictive Unsupervised Supervised deep learning algorithms for Spike Streams (EPUSSS). The EPUSSS framework consists of several novel algorithms and computational components to mimic the brain learning process and cognitive behaviour in a two-level hierarchical manner. The proposed strategies in the EPUSSS model allows having a unified framework to perform both prediction and classification tasks in a hierarchical fashion.

Overall, the following are the advantages of the proposed EPUSS model, some of them demonstrated in the experimental analysis above:

- Prediction of the next time step(s) by learning input pattern's dynamics.
- Learning from the semi-supervised local learning outcome for classification purposes to improve the classification outcome in a supervised manner.

- The ability of life-long learning by controlling the energy of the system and avoiding network saturation.
- Keep the balance between network excitation and inhibition to empower the life-long learning behaviour and avoid vanishing and exploding in the learnt weights.
- The pruning mechanism and energy function integrated into the EPUSSS model helps to boost the idea of long-life learning.

The EPUSSS method is a subject of a paper under construction.

Chapter 4 Using Chaos Theory to Develop a New Quantum-Inspired Evolutionary Algorithm: CQIEA

Abstract

SNN are known for having difficulties in optimising their numerous parameters and its performance. The EPUSSS model from chapter 3 also suffers from polynomial computational complexity. Evolutionary algorithms can provide promising solutions to overcome these drawbacks. Quantum-Inspire Evolutionary Algorithm (QIEA) is one of the fastest evolutionary algorithms that use quantum computation concepts to perform optimisation tasks. In this chapter an improved version of QIEA using chaos theory is introduced to address the research question 4:

RQ4 “How can the QIEA be further developed using a quantum chaotic system and applied for optimised learning, searching a pattern in stored memory, and parameter optimisation in SNN?”

The Chirikov chaotic map is used to improve the convergence rate and the quality of the final solution of the Quantum-Inspired evolutionary algorithm. The new method, which is called Chaotic Quantum-Inspire Evolutionary Algorithm (CQIEA) extends the methods proposed by Defoin-Platel, Schliebs and Kasabov (2009) and by Dias, Vellasco and Abs da Cruz (2020) and its illustration here demonstrates a faster convergence to an optimal solution.

The method is also subject to a paper under construction.

4.1 Introduction

While the first part of this thesis was more focused on introducing a novel predictive model consisting of methods more biologically plausible to mimic the brain behaviour in learning and cognitive tasks, in this chapter and the next two chapters 5 and 6, computational models inspired by the quantum computing theory are proposed to empower the computational capacity of spiking neural networks. This chapter provides a new chaotic quantum-inspired evolutionary algorithm based on chaos theory that serves three purposes through this thesis, 1) used as an optimisation mechanism to optimise the proposed models' parameters to improve their performance, 2) applied as a learning rule in the novel quantum-inspired spiking neural network proposed in chapter 5, and 3) is part of the novel quantum-inspired associative memory for spiking neural network introduced in chapter 6 to retrieve the learnt patterns. Although these three chapters including the novel quantum-based computational models might look completely different from the first part of this thesis, there is a strong connection between the biological behaviour of neurons in the macrolevel (molecular level) and the quantum-based behaviour at the microlevel (atomic level) suggested by Hameroff and Penrose (2014).

In the following sections, quantum mechanics concepts related to the proposed quantum-inspired models are briefly reviewed. Afterwards, a short introduction to chaos theory is presented. Finally, the novel chaotic quantum-inspired evolutionary algorithm (CQIEA) is introduced for both discrete and continuous optimisations.

4.2 Quantum Mechanics Concepts

The idea of quantum theory first was sparked in 1900, when Max Planck shook the foundation of two classical physics concepts of particle and electromagnetic wave by publishing a theory about black-body radiation. In classical physics, a particle is a discrete entity with definite position and momentum which obeys Newton's laws of motion, and an electromagnetic wave

is an extended entity that can exist at any point of space provided by electric and magnetic fields which change according to Maxwell's laws of electromagnetism (Phillips, 2003). Planck's observation about black-body radiation that changed the previous assumption of two distinct, unrelated concepts of particle and wave in classical physics, revealed a strange phenomenon in which a physical entity can have both wave-like and particle-like properties. Later, in 1923, Louis de Broglie formally proposed the idea that particles of matters such as electrons can be both particle-like and wave-like. Thomson in 1972, proved that electrons' particles with no doubt can behave like waves. Since 1972, so many experiments done by researchers providing evidence about the wave-like behaviour of electrons, protons, neutrons, atoms, and molecules. These findings were in contrast with the classical definition of particles with well-defined trajectories. Therefore, the ambiguous word "particle" was not well suited for describing these microscopic objects. Instead, the term "quantum particle" shows that the object under discussion can have both particle-like and wave-like properties. A quantum particle does not have a clear position until the time of observation meaning that the probability of distribution of positions defines the quantum particle.

The quantum mechanical model of the atom

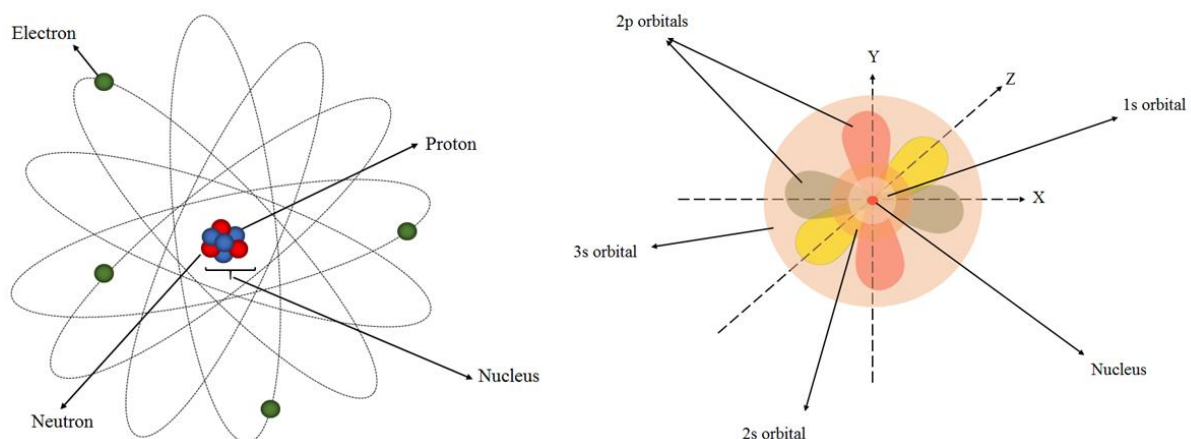


Figure 4-1: The quantum mechanical model of the atom

According to the Bohr atom model, the electrons in free atoms can exist in certain quantised energy states. The Bohr model provided the foundation for understanding the quantum theory of the atom and consecutively developing the quantum mechanical model of the atom (see Figure 4-1). These quantised energy states are associated with the orbits or shells of electrons in an atom.

According to the Pauli Exclusion Principle, the quantised state of any electron in the atom can be described by four unique quantum numbers: (1) principal quantum number, (2) azimuthal quantum number, (3) magnetic quantum number, and (4) spin quantum number (Beiser, 1987; Sherwin, 1959; Phillips, 2003; Peleg et al., 2010; Jankovic et al., 2014; Dommelen, 2018;).

4.2.1 Principal Quantum Number

The principal quantum number designates the energy level of the electron (electron shell) and is represented by n ranges from 1,2,3, ..., n ($n \in \mathbb{N}$). In other words, the principal quantum number shows the shell level (energy level) in which the electron is located. Higher principal energy levels consist of orbitals that are larger in size than the orbitals from lower energy levels.

4.2.2 Azimuthal Quantum Number


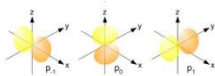
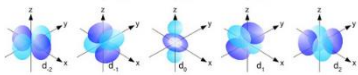
The azimuthal quantum number, also known as the angular quantum number or orbital quantum number, designates the subshell in which the electron is located. In other words, the azimuthal quantum number represents the magnitude of the orbital angular momentum using the following equation:

$$L^2 = \hbar^2 l(l + 1) \quad (4-1)$$

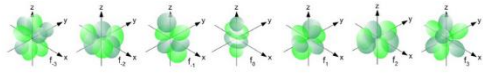
Where L is the angular momentum which is related to its azimuthal quantum number l ($l = 0, \dots, n - 1$) and \hbar is the Planck's constant which is $\hbar = 6.626 \times 10^{-34} Js$.

The azimuthal quantum number l refers to the shape of orbitals in any subshell. These shapes represent the most probable location that the electron can be found and are denoted by a letter according to a spectroscopy convention. Table 4-1, illustrates a brief description of the azimuthal quantum number and the corresponding subshell shapes² (Roesler and Mobley, 2015).

Table 4-1: Azimuthal quantum number subshell description

Azimuthal quantum number (l)	Correspon ding letter	Orbital shape	Maximum electron in a subshell	Visual representation
0	sharp	Spherical	2	
1	principal	Three dumbbell- shape on x, y and z axes	6	
2	diffuse	Complex dumbbells	10	

² In quantum physics, it is known that electrons move in orbitals around an atom's nucleus which is called the electron shell, or a principal energy level. The closest shell to the nucleus has the lowest energy and the shells further away from the nucleus have a higher energy level. Each shell contains a fixed number of electrons. Electrons occupy outer shells only if the more inner shells have already been completely filled by the maximum number of electrons that the shell can hold (the n th shell can hold up to $2n^2$ electrons). Each shell is also composed of one or more subshells. The first shell has one subshell (1s), the second shell has two subshells (2s and 2p), the third shell has three subshells (3s, 3p, and 3d), the fourth shell has four subshells (4s, 4p, 4d and 4f), and the fifth shell has five subshells (5s, 5p, 5d, and 5f) and can theoretically hold more in the 5g subshell that is not occupied in the ground-state electron configuration of any known element.

3	fundamenta l	Complex	14	
		multi lobes dumbbells		
4	g	Complex	18	The 5 th shell or higher theoretically defined for unknown atoms
		multi lobes dumbbells		

4.2.3 Magnetic Quantum Number

The magnetic quantum number describes the orientation of orbitals in any subshell in the three-dimensional space and is denoted by (m_l). The magnetic quantum number determines the exact orbital in which the electron is located. The value of m_l can range from $-l$ to $+l$ ($-l \leq m_l \leq l$). Each subshell (s, p, d, or f) contains a certain number of orbitals. In the s subshell ($l = 0$), the magnetic quantum number is zero, therefore there is one possible orientation of the orbital. In the p subshell ($l = 1$), the magnetic quantum number has three values ($-1, 0, 1$), therefore, there are three possible orientations of the orbitals. In the d subshell ($l = 2$), the magnetic quantum number has five values ($-2, -1, 0, 1, 2$), therefore, there are five possible orientations of the orbitals. In the f subshell ($l = 3$), the magnetic quantum number has seven values ($-3, -2, -1, 0, 1, 2, 3$), therefore, there are five possible orientations of the orbitals. All three axes for orbital orientations are perpendicular.

4.2.4 Electron Spin Quantum Number

The spin quantum number designates the type of spin that electron has in orbital (clockwise or counterclockwise) and is denoted by (m_s). In one orbital, the orientation of the spin of two electrons is always the opposite of each other, hence, the electrons can have m_s of $\frac{1}{2}$ or $-\frac{1}{2}$. In other words, one electron spins up and the other electron spins down. For a spin up the $m_s =$

$\frac{1}{2}$ and for a spin down the $m_s = -\frac{1}{2}$. The electron spin creates a small magnetic field that determines the magnetic property of an atom. If all the electrons in an atom are paired in the orbitals, then the atom is diamagnetic. Meaning that the spins with opposite signs ($+\frac{1}{2}$ and $-\frac{1}{2}$) repel each other magnetic fields and the total spin value is zero. If an atom has unpaired electrons in the orbitals, then the atom is paramagnetic. Meaning that electrons are attracted to a magnetic field and the whole atom will have a net spin.

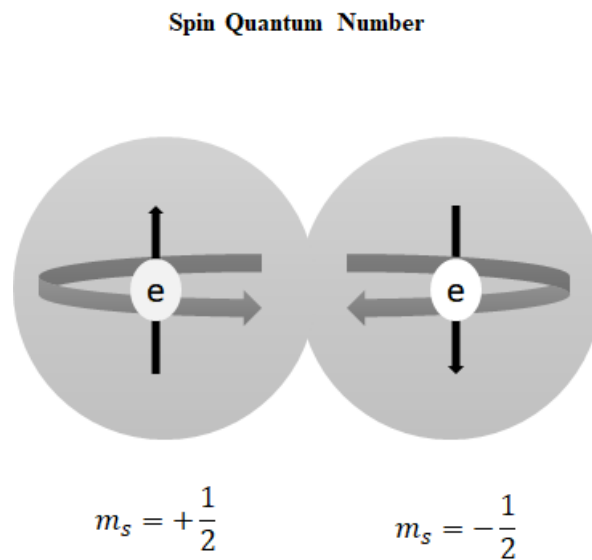


Figure 4-2: The spin quantum number schematic

Walter Heitler and Fritz London in 1927, proposed a model for hydrogen atoms that explains atomic magnetism (Magnasco, 2007; Misra, 2012; Jascur, 2013). According to their model hydrogen molecules are formed from hydrogen atomic orbitals u_A and u_B with nuclei centres A and B . The hydrogen atoms orbital $\Theta(r_1, r_2)$ in their model which is called two-body molecular orbitals are calculated using the following formula:

$$\Theta(r_1, r_2) = \frac{1}{\sqrt{2}} (u_A(r_1)u_B(r_2) + u_B(r_1)u_A(r_2)) \quad (4-2)$$

In Eq. (4-2), r_1 and r_2 are the first and the second electrons in an atomic hydrogen-orbital with A and B nuclei as their centres. The later product term in the above formula shows electrons exchange between nuclei A and B in which the first electron r_1 is in the atomic orbital centered at the second nucleus while the second electron orbits around the first nucleus. This electron exchange depicts the quantum concept in which particles with identical properties cannot be distinguished. Furthermore, the electron exchange leads to chemical bond formation as well as atomic magnetism. As it was mentioned before, magnetism is generated because of atomic spins. Therefore, according to Pauli's principle, the spin function $x(s_1, s_2)$ can be represented as follows using Heitler and London model:

$$x(s_1, s_2) = \frac{1}{\sqrt{2}} (\alpha(s_1)\beta(s_2) - \beta(s_1)\alpha(s_2)) \quad (4-3)$$

In Eq. (4-3), the symmetric spin function s_1 (with the + sign as spin-up \uparrow) multiplies by an antisymmetric spin function s_2 (with the – sign as spin-down \downarrow), and vice versa. The $s_i = \pm \frac{\hbar}{2}$ which in particles constitute ordinary matters is $s = \pm \frac{1}{2}$ is a discrete value with α and β spin orientation (0 and 1 respectively).

The abovementioned formula helps to explain the diamagnetic and paramagnetic behaviour in materials according to quantum mechanics theory (Eisberg, 1974; Tang, Toennies, and Yiu, 1998; Esposito and Naddeo, 2013).

4.2.5 Quantum Wave Function

In classical physics, the wave function for a freely moving particle is the real function of space and time. However, in quantum mechanics, the wave functions which describes the hidden wave-like behaviour of a quantum particle are complex functions. In fact, the wave function which is denoted by ψ is a complex-valued probability that defines all possible results of measurements made on a quantum system (Beiser, 1987). The wave function is described by the Schrodinger equation in quantum mechanics which is analogous to Newton's Second Law for describing a classical particle motion (Eq. (4-4)) (Phillips, 2003).

$$\Psi(x, t) = \int_{-\infty}^{+\infty} A(k)e^{i(kx-\omega t)} dk \quad (4-4)$$

With

$$\hbar\omega = \frac{\hbar^2 k^2}{2m} \quad (4-5)$$

Where $k = \frac{2\pi}{\lambda}$ is the wave number with λ wavelength, ω is velocity, m particle mass, and $A(k)$ is an arbitrary complex function of k and the integral indicates the sum over all possible values of k . In the above equation, $\hbar\omega$ represents the energy of the particle and $\hbar k$ is the particle momentum. In other words, Eq. (4-4) describes a wave moving in the x direction with wave number k and angular velocity ω with a momentum $p = \hbar k$ and energy $\hbar\omega$. The Schrodinger wave function in agreement with the Heisenberg uncertainty principle represents uncertainty in the position and momentum of a quantum particle.

4.3 Quantum Computation

Quantum computing is an interesting research area in which researchers adopt quantum mechanics concepts to develop more powerful computational models.

In quantum computation theory, a quantum system is described by a wave function ψ that is formed based on quantum states in the Hilbert space:

$$|\psi\rangle = \sum_i C_i |\phi_i\rangle \quad (4-6)$$

Where, C_i is the probability amplitudes and $|\phi_i\rangle$ is the quantum states. These quantum states are called quantum bits or qubits. The qubit can be in a superposed state. It means that, unlike the classical bit, in addition to the usual state “0” and “1” the qubit can take both values simultaneously. Hence, according to the superposed state definition of the qubits Eq. (4-6) can be rewritten as follows:

$$|\psi\rangle = \sum_i \alpha_i |0\rangle + \beta_i |1\rangle \quad (4-7)$$

$$|\alpha|^2 + |\beta|^2 = 1 \quad \alpha, \beta \in \mathbb{C} \quad (4-8)$$

$|\alpha|^2$ and $|\beta|^2$ are the probabilities of the qubit to be found in the states $|0\rangle$ and $|1\rangle$ respectively after a measurement and $i = 1, \dots, n$.

In the quantum system $|\psi\rangle$ qubits have a quantum correlation which means that any interaction with one of the qubits affects instantaneously the other. This phenomenon is called quantum entanglement. The superposition and the entanglement properties of a quantum system make it capable of performing multiple computations simultaneously in a parallel way.

While a quantum system is in a linear superposition of its basis states it is coherent. Once the system intersects in any way with its environment, the superposition is destroyed. This loss of coherence is called decoherence and is governed by the wave function $|\psi\rangle$. In other words, at the time that the system is observed, its qubits can only be in one state. Quantum gate operators on a Hilbert space describe how one wave function is changed into another. These quantum gates are unitary and reversible which includes, Feynman gate, Walsh-Hadamard gate, Not gate and rotation gate.

The changes in the state of the wave function can be represented by the Bloch sphere which is a generalised representation of Hilbert space.

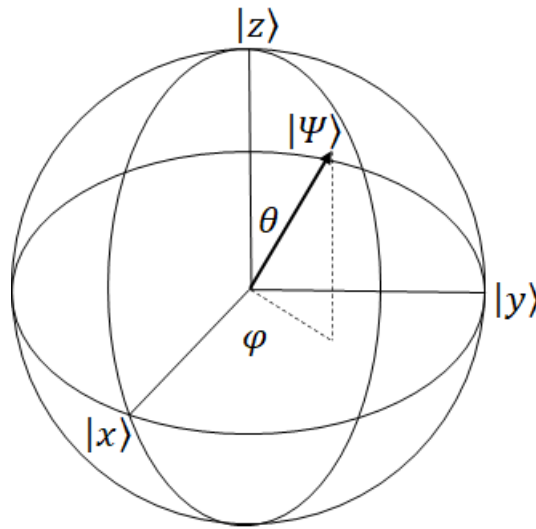


Figure 4-3: Bloch sphere representation of the evolution of the wave function

The equivalent Bloch sphere representation of the wave function described in Eq. (4-7) for two-level quantum system $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is formulated as follows:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \quad (4-9)$$

With:

$$0 \leq \theta \leq \pi \quad \text{and} \quad 0 \leq \varphi \leq 2\pi \quad (4-10)$$

where θ and φ are angles of rotation in the z and x direction.

In the Bloch sphere, the poles represent classical bits $|0\rangle$ and $|1\rangle$, and the changes in the quantum bits that can cover the whole sphere represents the possibility of the positions that qubits can hold in the wave function. Thus, the quantum bits can represent much more information in comparison to the classical representation of bits.

In the next section chaos theory and linear systems are briefly reviewed to provide an insight into the proposed CQIEA algorithm.

4.4 Chaos Theory

Chaos is not a strange concept to human beings. In fact, we can see chaotic systems all around us. Some of the chaotic systems examples in nature can be found in meteorology, solar system, heart, and brain of living organisms (Boccaletti et al., 2000). This natural phenomenon has been attracting scientists' attention in almost every discipline such as physics, engineering, economics, biology, and philosophy since the 1880s.

Chaos theory studies the behaviour of a dynamic system that is deterministic intrinsically but reacts random and unpredictable whenever small changes happen in its initial conditions (Ogorzalek, 1993; Gaspard and Wang, 1993; Rovatti et al., 1998; Hirsch et al., Stojanovski and Kocarev, 2001; 2004; Werndl, 2010; Strogatz, 2014). In other words, a dynamic system is called chaotic if it is sensitive to initial conditions and small changes in those conditions can lead to quite different outcomes (Ogorzalek, 1993; Rovatti et al., 1998; Stojanovski and Kocarev, 2001; Werndl, 2010).

The dynamic of a chaotic system is relatively simple and can be expressed in the form of a mathematical expression with few variables (Strogatz, 2014). Since chaotic systems are

evolving dynamic systems, in their trajectory movement they tend to stabilise to a set of numerical values which are called strange attractors. A chaotic attractor is a collection of an infinite number of periodic orbits, each one being unstable. Due to the ergodic nature of a chaotic attractor, during its temporal evolution, the system ergodically visits a small neighbourhood of every point in each one of the unstable periodic orbits embedded within the chaotic attractor (Gaspard and Wang, 1993; Yang and Chua, 1997; Hirsch et al., 2004). These important properties of chaotic systems can inspire evolutionary meta-heuristic search algorithms so that by mimicking chaotic behaviour, these algorithms will be more powerful and precise in terms of solving complex non-linear NP-hard problems.

The trajectory of a chaotic system through the space of possible states is a complicated tangle of looping paths of unstable periodic orbits. The geometry of these unstable periodic orbits is one of the distinctive characteristics of chaos. Due to the critical dependence of chaotic systems on the initial conditions, these systems are naturally unpredictable (Strogatz, 2014). However, using mathematical notions such as the Lyapunov exponent, the speed at which the trajectories of a chaotic system starting out with similar initial conditions will diverge, can be quantified (Gaspard and Wang, 1993; Hirsch et al., 2004). As it was mentioned before and based on the ergodic nature of chaotic attractor, chaotic dynamics of a system can be seen as periodic behaviour at a given time, and erratically jumping from one to another periodic orbit. Therefore, controlling chaos when a trajectory approaches ergodically a desired periodic orbit embedded in the attractor, by applying small perturbations to stabilise such an orbit (Yang and Chua, 1997), can motivate designing more sophisticated search mechanisms for evolutionary meta-heuristics algorithms.

4.5 Previous Work

In this section, some of the worthwhile research work related to quantum-inspired evolutionary algorithms and using chaos theory in optimisation methods are briefly reviewed.

4.5.1 Quantum-Inspired Evolutionary Algorithms

Quantum-Inspired Evolutionary Algorithms (QIEAs) are inspired by the concepts of quantum computing like quantum bits and quantum gate, was first introduced by Feinman (Feinman, 1980) A brief review of the work done so far on QIEAs is provided in the following:

A simple QIEA was used by Narayanan and Moore in (1996) to solve the travelling salesman problem. Han and Kim (2000) proposed the genetic quantum algorithm (GQA) which was based on the concept and principles of quantum computing such as qubits and superposition of states. They used qubit chromosome for representation of the individual and quantum gates as genetic operators for searching the solution space. Han and Kim (2002) proposed the quantum-inspired evolutionary algorithm (QEA) based on the concept and principles of quantum computing, such as a quantum bit and superposition of states. They introduced qubit for representation of the individual and a q-gate as a variation operator to drive the individuals toward better solutions. Han and Kim (2004) proposed a new termination criterion, a q-gate, and a two-phase scheme quantum-inspired evolutionary algorithm to improve the performance of QEA. Zhang et al. (2006), combined the quantum computing concept with genetic algorithm and called it novel quantum genetic algorithm (NQGA) and introduced a rotation quantum gate to enhance the search capability and avoid premature convergence. NQGA is used for selecting features in radar emitter signal recognition. Zhang and Rong (2007), proposed a real-observation quantum-inspired evolutionary algorithm (RQEA) to solve a class of globally numerical optimisation problems with continuous variables. Platel et al. (2009), proposed the

versatile quantum-inspired evolutionary algorithm (vQEA) which uses the attractor concept to move the population through the search space and can adapt the search toward the last promising solution found. Zhang et al. (2008), introduced an evolutionary algorithm that combined the concepts and principles of the quantum-inspired evolutionary approach with the hierarchical arrangement of the compartments of a P system for solving the knapsack problem. Vlachogiannis and Lee (2008), used a quantum computation evolutionary algorithm for the combinatorial optimisation problem of optimal real and reactive power (P-Q) dispatch considering the bid-offered cost. Platel et al. (2009), stated that QEA belongs to the class of estimation of distribution algorithms (EDAs) and provided a comparison between versatile QEA and three classical EDAs in terms of loss of diversity, scalability, solution quality, and robustness to fitness noise. Zhang (2010) provided a unified framework and comprehensive survey of work done in the field of Quantum-Inspired Evolutionary Algorithm. The author discussed the main concepts and key ideas behind quantum-inspired evolutionary algorithms and the advantages and limitations of those methods. Zhang et al. (2010), combined the quantum-inspired evolutionary algorithm and P systems and introduced a modified variant quantum-inspired evolutionary algorithm for analysing radar emitter signals, called MQEPS. Their model used the hierarchical framework of cell-like P systems consisting of quantum-inspired bits and classical bits which the system rules are quantum-inspired gate evolutionary rules, evolution rules in P systems, and a tabu search in the skin membrane. Nowotniak and Kucharski (2014) introduced a higher-order quantum-inspired genetic algorithm in which genes relations are modelled using quantum phenomena for optimisation tasks. Silveira et al. (2016) proposed a Quantum Inspired Evolutionary Algorithm for Ordering Problems. González et al. (2019) proposed resource optimization for elective surgical procedures using quantum-inspired genetic algorithms. Dias et al. (2020) proposed a quantum-inspired method

for optimisation applications. Pereira et al. (2020) proposed a Quantum-Inspired Genetic Programming Algorithm for the Crude Oil Scheduling of a Real-World Refinery.

QIEAs have a weak local search. In other words, QIEAs have poor performance in searching the neighbourhood of the best solution. The other limitation of these techniques is that the rotation quantum gate's angle and the direction of its movement require prior knowledge to be initialised.

Celecia et al. (2020) proposed perhaps the first Chaotic Quantum-inspired Evolutionary Algorithm and applied it to enhance feature selection in Brain-computer Interface (BCI). This thesis continues the above direction with the introduction of a new CQIEA.

4.5.2 Chaotic Optimisation Algorithms

The non-repetition nature of chaotic systems can speed up search mechanisms in meta-heuristic algorithms. To overcome the premature convergence drawback of these algorithms and to perform an efficient fast exploration and exploitation in the problem space, many researchers deployed different kinds of chaotic maps in combination with evolutionary algorithms such as Genetic Algorithm, Particle Swarm Optimisation, Differential Evolution algorithm etc. to improve these meta-heuristics performances. Logistic map, Tent map, Sinusoidal map, Gauss map, Circle map, Sinus map, Henon map, Ikeda map, Zaslavskii map, Sprott-Linz chaotic attractor, Lozi map, Lorenz attractor are some of these chaotic maps. In the following, some of the remarkable recent research which has been combining chaotic mechanism with meta-heuristic algorithms are mentioned.

Differential Evolution (DE) algorithm convergence rate and population diversity were improved by applying different chaotic maps (chaotic logistic map, Lozi's map and Tent Map) into its control parameters such as the population initialization, the mutation, and the crossover

operators (Santos Coelho, 2009; Santos Coelho, et al., 2009; Lu et al., 2011). A wide range of Particle Swarm Optimisation (PSO) variations are enhanced by various chaotic maps (such as Henon map, Zaslavskii map, dissipative standard map, Lozi map, Arnold's Cat map and logistic map) to speed up their convergence rate and avoid premature convergence to local optima (Santos Coelho and Mariani, 2009; Santos Coelho and Rodrigues Coelho, 2009; Alatas and Akin, 2009; Tang et al., 2010; Chuang et al., 2011; Pluhacek, et al., 2013; Gandomi et al., 2013; Chen et al., 2014; He et al., 2014; Kaveh et al., 2014). In these works, researchers combined chaos theory in particle position, velocity formula, and particle acceleration control parameter to improve PSO exploration and exploitation capabilities. In references (Enayatifar et al., 2014; Yu and Xu, 2014; Chen et al., 2014), Genetic Algorithm search ability is improved by using chaotic sequence generators (logistic map and Cat map) into crossover and mutation operators. In another effort, the chaotic logistic map was applied to fireflies' movements in Firefly Algorithm to improve the quality of solutions and prevent the algorithm from trapping in local optima (Kazem et al., 2013).

To overcome QIEA limitations, Chirikov standard map is used to substitute the rotation gate since it satisfies all the conditions that a quantum gate requires.

4.6. The Proposed Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA)

The proposed here CQIEA is inspired by the QIEA (Defoin-Plate, Schliebs and Kasabov, 2009) and by the CQIEA (Celecia, Alimed & Vellasco (2020).

The novel Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA) proposed in this thesis is an improved version of the QIEA algorithm as it is modified by the Chirikov standard map as the quantum gate to change the qubits states in the search space and with the help of the

strange attractor concept the search mechanism can be guided in a more sophisticated way towards the global optimum solution.

The QIEA algorithms have a strong searching capability, rapid convergence, short computing time, and small population size in comparison to other population-based evolutionary algorithms. The QIEA is based on quantum computation concepts that were introduced in the previous section of this thesis. This algorithm like every other population-based evolutionary algorithm has a population of possible solutions called qubits and a mechanism to guide this population towards the optimum places in the search space using a quantum gate (q-gate) operator.

In QIEA, qubits are the smallest unit of information that provide the probabilistic representation of the individual solutions and are defined with a pair of α and β as $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$. The α represents the probability in which the qubit is in the ‘0’ state while β expresses the probability in which the qubit is in the ‘1’ state. According to quantum mechanics, the qubit can be in the ‘0’ state, in the ‘1’ state, or in a linear superposition of both states.

In QIEA, a qubit individual solution consists of m qubit is defined as follow:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \quad (4-11)$$

where $|\alpha|^2 + |\beta|^2 = 1$ and where $i = 1, 2, \dots, m$.

The linear superposition of states probability in the qubit representation allows creating a more diverse population in comparison to other representations. For instance, one qubit individual consisting of three qubits $\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix}$ can represent eight entangled states $\alpha_1\alpha_2\alpha_3|000\rangle +$

$\alpha_1\alpha_2\beta_3|001\rangle + \alpha_1\beta_2\alpha_3|010\rangle + \alpha_1\beta_2\beta_3|011\rangle + \beta_1\alpha_2\alpha_3|100\rangle + \beta_1\alpha_2\beta_3|101\rangle +$
 $\beta_1\beta_2\alpha_3|110\rangle + \beta_1\beta_2\beta_3|111\rangle$ while in the binary representation at least eight individuals are needed to code these states.

The state of qubits can be changed by the quantum gate (q-gate) operator. As it was mentioned before, a q-gate is a reversible unitary operator. A unitary operator U which is defined on a Hilbert space H is a bounded linear operator that maps a vector space to another. To be a unitary operator, U must satisfy the following condition.

$$UU^\dagger = U^\dagger U = \mathbb{I} \quad (4-12)$$

where U^\dagger is the Hermitian adjoint of U . QIEA uses the quantum rotation gate as a q-gate:

$$G(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad (4-13)$$

where θ_i , is a rotation angle of each qubit that can change the qubit's state to either 0 or 1 depending on its sign. The θ_i angle should be defined according to the problem.

The QIEA algorithm consists of a population of qubit individuals to search the solution space for the optimum solution. The population of qubits at generation t is $Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$, where n is the size of the population. The binary solutions $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ can be obtained by observing the state of $Q(t)$ at generation t . The binary solution $x_j^t, j = 1, \dots, n$, is a binary string of length m in which each bit in the string can be selected to be 0 or 1 using $|\alpha_i^t|$ and $|\beta_i^t|$, $i = 1, 2, \dots, m$, probabilities determined by the q-gate. At observation time, at each generation, the best local solutions $B(t) = \{b_1^t, b_2^t, \dots, b_n^t\}$ are selected among the binary solutions $P(t)$ by evaluating the qubits' fitness value (i.e., by observing the states of $Q(t - 1)$). Then, the best local solution(s) at each generation will be compared with the best global

solutions found so far and if the migration condition is satisfied the current the best local solution(s) will be migrated to the global $B(t)$. This process will be repeated until the termination criteria are satisfied and then the global best solution(s) will be provided at the final generation.

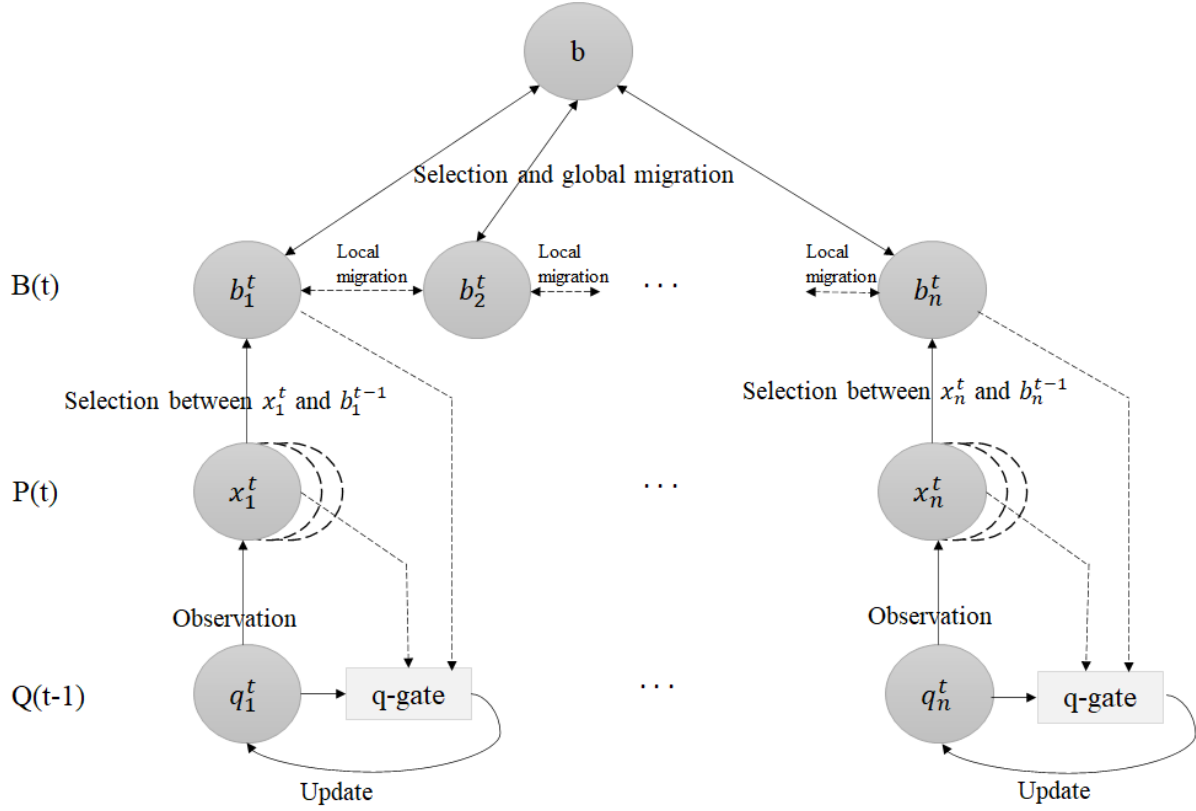


Figure 4-4: The structure of QIEA algorithm (Han and Kim,2002).

The steps in the QIEA algorithm are described as follow:

Step 1 initializing $Q(t)$: a population $Q(t)$ with the qubit individual is generated, $Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$, each arbitrary individual in $Q(t)$ is represented as

$$q_i^t = \begin{bmatrix} \alpha_{i,1}^t | \alpha_{i,2}^t | \dots | \alpha_{i,l}^t | \\ \beta_{i,1}^t | \beta_{i,2}^t | \dots | \beta_{i,l}^t | \end{bmatrix} \quad (4-14)$$

Where l is the number of qubits (the string length of the qubit). In the initial population, all possible states are superposed with the same probability at the beginning.

Step 2 By observation, the states $Q(t)$, binary solutions in $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$, are generated at step t . According to the current probability, either $|\alpha_i^t|^2$ or $|\beta_i^t|^2$ of $q_i^t (i = 1, 2, \dots, l)$, a binary bit 0 or 1 is generated. Thus, a binary solution $x_j^t (j = 1, 2, \dots, n)$ consists of l binary bits.

Step 3 The fitness value for each value for each binary solution $x_j^t (j = 1, 2, \dots, n)$ is calculated by using an evaluation function.

Step 4 In this step, the j th qubit in the i th qubit individual $q_i^t, j = 1, 2, \dots, l, i = 1, 2, \dots, n$ is updated by applying the current q-gate $G_{ij}^t(\theta)$. The quantum rotation gate is used in QIEA as a q-gate:

$$G_{ij}^t(\theta) = \begin{bmatrix} \cos \theta_{ij}^t & -\sin \theta_{ij}^t \\ \sin \theta_{ij}^t & \cos \theta_{ij}^t \end{bmatrix} \quad (4-15)$$

Where θ_{ij}^t is an adjustable q-gate rotation angle. The updated qubit states after applying quantum rotation gate are as follow:

$$\begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix} = G_{ij}^t(\theta) \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix} \quad (4-16)$$

$$\theta_{ij}^t = s(\alpha_{ij}^t, \beta_{ij}^t) \Delta \theta_{ij}^t \quad (4-17)$$

where $s(\alpha_{ij}^t, \beta_{ij}^t)$ is the direction of the quantum gate rotation (the sign of θ) and $\Delta \theta_{ij}^t$ is the magnitude of rotation angle θ .

Step 5 The best solutions among $P(t)$ are selected and stored.

As it was mentioned before in the literature review, QIEAs have a weak local search. In other words, QIEAs have poor performance in searching the neighbourhood of the best solution. The other limitation of these techniques is that the rotation quantum gate's angle and the direction of its movement require prior knowledge to be initialised.

To overcome these shortages, a new quantum gate using Chirikov Standard Map is introduced here to provide a fast and effective local search that does not require any prior knowledge to set the q-gate parameters. The Chirikov Standard Map is a chaotic map that has both characteristics of the quantum gates; it is unitary and reversible. A chaotic map is a mathematical expression with few variables which shows the dynamic of a chaotic system. A chaotic system is a dynamic system that is deterministic intrinsically but reacts random and unpredictable whenever small changes happen in its initial conditions (Ditto and Munakata, 1995; Strevens, 2006; Baranger, 2013). Since chaotic systems are evolving dynamic systems, in their trajectory movement they tend to stabilise to a set of numerical values which is called strange attractors. A chaotic attractor is a collection of an infinite number of periodic orbits, each one being unstable. Due to the ergodic nature of a chaotic attractor, during its temporal evolution, the system ergodically visits the small neighbourhood of every point in each one of the unstable periodic orbits embedded within the chaotic attractor (Boccaletti et al, 2000). Inspiring by these important properties of chaotic systems, the Chirikov Standard map is introduced as a q-gate to improve the QIEA search ability.

4.6.1 Chirikov Standard Map

The Standard map which was proposed by Boris Chirikov (Chirikov, 1971), is an area-preserving map for two dynamical variables momentum and coordinate which is generated by the time-dependent Hamiltonian system. The standard map has many symmetries, the most

important of which is the time-reversibility symmetry (Meiss, 2008; Miguel et al., 2013) that makes it a good candidate for designing a q-gate.

The following is one of the canonical forms of the map:

$$r_{n+1} = r_n + K \sin(\theta_n) \quad (4-18)$$

$$\theta_{n+1} = \theta_n + r_{n+1} \quad (4-19)$$

In polar coordinates the value of r becomes the magnitude of the radial vector, θ is the polar angle, and K is a dimensionless parameter that controls the degree of chaos. The periodic nature of $\sin x$ can result of a cylinder shape dynamic for $\theta \bmod 2\pi$ or a torus shape dynamic for both θ and $r \bmod 2\pi$. Chirikov map is a conservative system that can demonstrate both regular and chaotic dynamics. The map's regular behaviour occurs in the Kolmogorov-Arnold-Moser (KAM) islands of stability which are embedded in the chaotic sea. These islands are stable periodic orbits, and their eigenvalues are equal to one (Kob and Schilling, 1989). Figure 4-5 illustrates examples of Poincare sections (i.e., the periodic and quasi-periodic orbits of a discrete dynamical system with state space of one dimension lower than the original continuous dynamical system) of the Chirikov standard map on a torus that are resulted from different K values. For $K = 0$ the map is linear and by increasing the K value the nonlinearity of the map grows higher. Chirikov standard map shows chaotic behaviour for $K > 1.2$ which is the critical value.

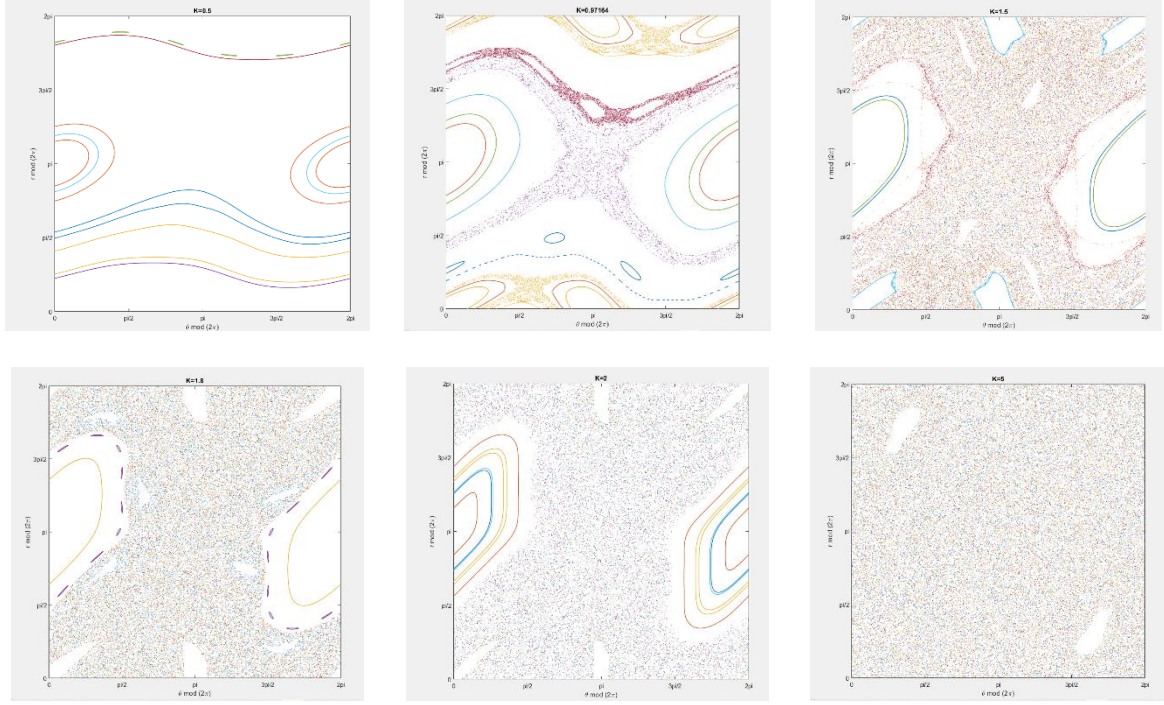


Figure 4-5: Chirikov standard map with different k values, $k=0.5, 0.8, 0.971635, 1.5, 1.8, 2$ (from the left most top to the right most bottom).

4.6.2 The Proposed Quantum Chirikov Standard Map

The Chirikov standard map is generated by a time-dependent Hamiltonian system:

$$H(r, \theta, t) = \frac{r^2}{2} + K \cos(\theta) \delta_1(t) \quad (4-20)$$

where $\delta_1(t)$ is a periodic function with period 1 in time t . The map dynamics describes Poincare's surface of a discrete mechanical system called the kicked rotator. Kicked rotator which is a basic model for chaos and quantum chaos studies describes a particle that is constrained to move on a circle in a system with no friction and gravity (or can be considered as a pendulum swinging with no gravity) and is kicked periodically by a homogeneous field.

Therefore, the Chirikov standard map can be expressed in quantised format to give the quantum map for the wave function ψ :

$$\bar{\psi} = \hat{U}\psi = e^{-i\hat{r}^2/2\hbar} e^{-iK/\hbar \cos \hat{\theta}} \psi \quad (4-21)$$

Where, $\bar{\psi}$ is the new value of the wave function after map iteration, \hbar is the Planck constant, and \hat{U} is the unitary evolution operator. An increase in the K values can increase the energy of the system linearly over time (Giorgilli and Lazutkin, 2000; Meiss, 2008; Gorodetski, 2012; Frahm and Shepelyansky, 2013; Miguel, Simó, and Viero, 2013; Karimov, et al., 2017; Blumenthal, 2021). Eq. 4-21 shows that the Chirikov standard map is a unitary operator that either can be directly used to change the current state of a quantum system in a chaotic manner or the chaotic angular position presented in Eq. 4-18 and 4-19 can be considered as the angle for the rotation gate. In this thesis, we choose the latter approach and leave the former for future work.

4.6.3 The Proposed Chirikov Quantum Gate

As it was mentioned before, a quantum gate is a reversible unitary operator which ensures the sum of probabilities of all possible states of a qubit during the evolution of the quantum systems always equals to one. In the previous sections, it was shown that the Chirikov standard map has all the characteristics required for a quantum gate. Using Chirikov standard map as a q-gate (diversity operator), the problem search space can be explored and exploited more efficiently (see Figure 4-6). As it was mentioned before, the Chirikov standard map can produce both chaotic and regular behaviour with different values of K . Therefore, during the evolution, the dynamic of the system can be controlled efficiently using adaptive values for K . At the beginning of the search/optimisation problem by setting a higher value for K and increasing the chaotic behaviour of the system the search space can be explored more efficiently. By decreasing the values of K over time, the system will produce more regular behaviour than chaotic dynamics which can results in converging to more stable states and exploiting search space to find an optimum solution.

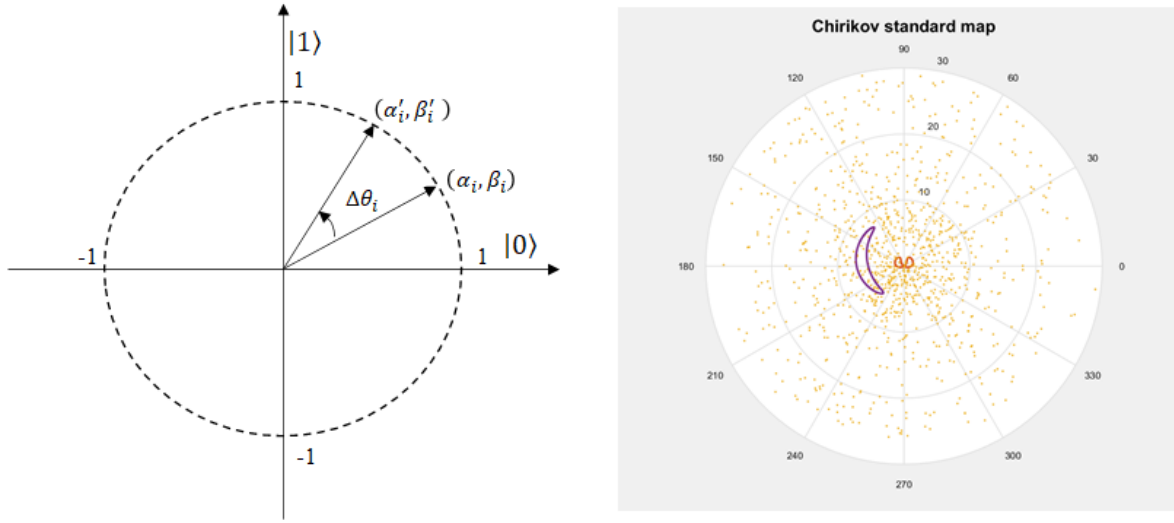


Figure 4-6: Polar plot of the rotation gate and Chirikov map for qubit individuals

The proposed Chirikov quantum gate uses the chaotic angular position presented in Eq. 4-18 and 4-19 as the θ angle for the rotation gate.

4.6.4 The Proposed CQIEA algorithm

In the proposed CQIEA search mechanism by controlling the chaos with an adaptive parameter K , the population movements in the CQIEA will guide to the desired locations during the evolution of the search algorithm. Therefore, the problem search space will be explored and exploited in a more sophisticated way through time. Setting the K parameter to a higher value in the Chirikov quantum gate at the beginning of the search process causes more chaotic behaviour in the system and by decreasing it over time the system behaviour will be more regular and stable. This mechanism K will decrease through the search period by a decay factor of 0.05. Figure 4-7 shows the proposed CQIEA algorithm pseudocode.

Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA) – Discrete Optimisation

```
1:   Set  $t=0$  and  $K = 5$ ,  $\text{population\_size} = 10$ ,  $\text{max\_iteration} = 200$ , and  $\text{decay\_factor} = 0.05$ .

2:   Initialise the  $Q(t)$  population with all the  $\alpha$  and  $\beta$  probabilities set to  $\frac{1}{\sqrt{2}}$ .

3:   Generate  $P(t)$  by observing the state of  $Q(t)$ .

4:   Evaluate  $P(t)$  using the fitness function.

5:   Store the best solutions in  $P(t)$  to  $B(t)$ .

6:   while  $it \leq \text{max\_iteration}$ 

7:        $t \leftarrow t + 1$ 

8:       Generate  $P(t)$  by observing the state of  $Q(t - 1)$ .

9:       Evaluate  $P(t)$  using the fitness function.

10:      Update  $Q(t)$  using the Chirikov q-gate

11:       $K \leftarrow K - \text{decay\_factor}$ 

12:      Store the best solutions among  $B(t - 1)$  and  $P(t)$  to  $B(t)$ .

13:      Store the best solution  $b$  among  $B(t)$ .

14:      if (global migration condition is met)

15:          Migrate  $b$  to  $B(t)$  globally.

16:      else if (local migration condition is met)

17:          Migrate  $b_j^t$  in  $B(t)$  to  $B(t)$  locally.

18:      end if

19:  end while
```

Figure 4-7: CQIEA algorithms pseudocode for discrete optimisation

By adding a simple binary to decimal (real value) and decimal to binary conversion CQIEA algorithm can be used for continuous optimisation too. The pseudocode of the algorithm is depicted in Figure 4-8.

Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA) – Continuous Optimisation

```
1:   Set  $t=0$  and  $K = 5$ ,  $\text{population\_size} = 10$ ,  $\text{max\_iteration} = 200$ , and  $\text{decay\_factor} = 0.05$ .

2:   Initialise the random real-valued population  $R(t)$  bounded between the lower and upper bounds of
      the problem space.

3:   Convert the real-valued initial population  $R(t)$  to the binary-valued  $Q(t)$  population with all the  $\alpha$ 
      and  $\beta$  probabilities set to  $\frac{1}{\sqrt{2}}$ .

4:   Convert the binary-valued  $Q(t)$  population to real-valued population  $R(t)$ .

5:   Generate  $P(t)$  by observing the state of  $R(t)$ .

6:   Evaluate  $P(t)$  using the fitness function.

7:   Store the best solutions in  $P(t)$  to  $B(t)$ .

8:   while  $it \leq \text{max\_iteration}$ 

9:        $t \leftarrow t + 1$ 

10:      Convert the binary-valued  $Q(t - 1)$  population to real-valued population  $R(t)$ .

11:      Generate  $P(t)$  by observing the state of  $R(t - 1)$ .

12:      Evaluate  $P(t)$  using the fitness function.

13:      Update  $Q(t)$  using the Chirikov q-gate

14:       $K \leftarrow K - \text{decay\_factor}$ 

15:      Store the best solutions among  $B(t - 1)$  and  $P(t)$  to  $B(t)$ .

16:      Store the best solution  $b$  among  $B(t)$ .

17:      if (global migration condition is met)

18:          Migrate  $b$  to  $B(t)$  globally.

19:      else if (local migration condition is met)

20:          Migrate  $b_j^t$  in  $B(t)$  to  $B(t)$  locally.

21:      end if

22:  end while
```

Figure 4-8: CQIEA algorithms pseudocode for continuous optimisation

4.7. Benchmark combinatorial optimisation problems

In order to evaluate the CQIEA algorithm for discrete optimisation, two combinatorial problems were chosen here which are close to the application of CQIEA for the methods proposed in the following chapters. The combinatorial optimisation problems are those where mathematical techniques are applied to find optimal solutions within a finite set of possible solutions. The set of possible solutions is generally defined by a set of restrictions that is too large for exhaustive search to solve. The first well-known example is the knapsack problem, where the value of the goods carried in the knapsack must be maximized, while the weight of the goods that can be carried is limited. The second example is the travelling salesman problem, where the total travelling distance must be minimized while each client is visited exactly once.

To evaluate the CQIEA algorithm for continuous optimisation which can be used for model's parameter optimisation, four well-known benchmark functions Sphere, Rastrigrin, Rosenbrock, and Ackley have been chosen and shown in Table 4-2.

4.7.1 Knapsack

Knapsack NP-combinatorial optimisation problem is defined as:

$$\max_x \sum_{j=1}^n p_j x_j \quad x_j \in \{0,1\}, \quad j = 1, \dots, n \quad (4-22)$$

Subject to:

$$\sum_{j=1}^n w_{i,j} x_j \leq c_i, \quad i = 1, \dots, m \quad (4-23)$$

where p_j and $w_{i,j}$ are positive weights, c_i is positive cost parameters, and $x = (x_1, \dots, x_n)$.

4.7.2 Travelling Salesman Problem (TSP)

The TSP NP-combinatorial optimisation problem is defined as:

$$\min \sum_i^n \sum_j^n c_{ij} x_{ij} \quad (4-24)$$

Subject to:

$$\sum_i^n x_{ij} = 1, \quad j = 1, \dots, n \quad (4-25)$$

$$\sum_j^n x_{ij} = 1, \quad i = 1, \dots, n \quad (4-26)$$

$$0 \leq x_{ij} \leq 1, \quad \text{for all } i, j \quad (4-27)$$

where c_{ij} is the cost matrix (or the distance) to travel from city i to city j and x_{ij} value is 1 if the path goes from city i to city j and is 0 otherwise.

4.7.3 Continuous Optimization Benchmarks

Table 4-2: The benchmark functions for continuous optimisation

Function	Mathematical expression	Bounds
Sphere	$f1(x) = \sum_{i=1}^D x_i^2$	(-100, 100)
Rastrigrin	$f2(x) = \sum_{i=1}^n (x_i^2 - 10 \times \cos(2\pi x_i) + 10)$	(-10, 10)
Rosenbrock	$f3(x) = \sum_{i=1}^n (100 \times (x_{i+1} - x_i)^2 + (x_i - 1)^2)$	(-600, 600)
Ackley	$f4(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	(-32, 32)

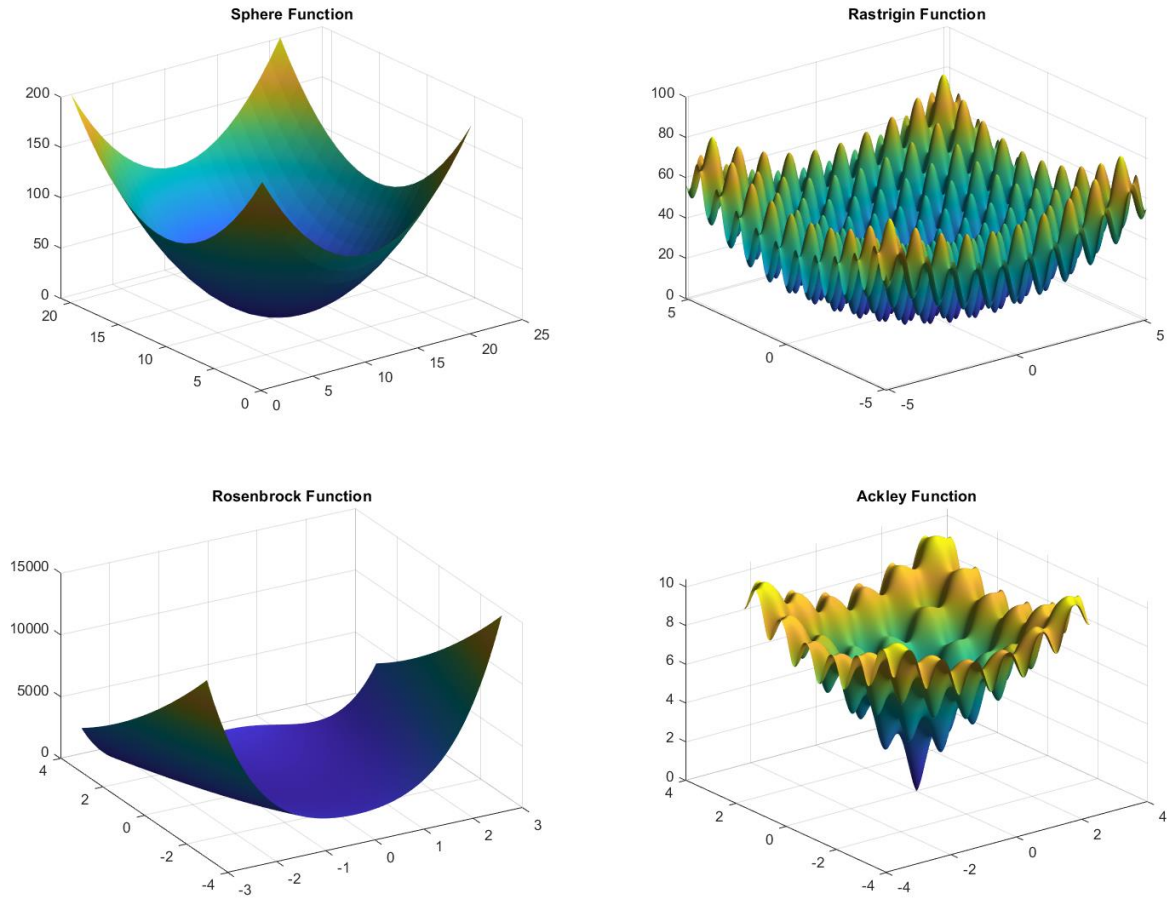


Figure 4-9: 3D representation of benchmark functions

4.8. Implementation and Experimental Analysis

In this section, the experimental results of applying the proposed CQIEA and QIEA on both discrete and continuous environments problems are illustrated and a comparative analysis via tables and plots is provided.

Figure 4-10 to Figure 4-12 show the comparative diagrams of the CQIEA and QIEA performance on the Knapsack problem with 10, 100, and 500 items.

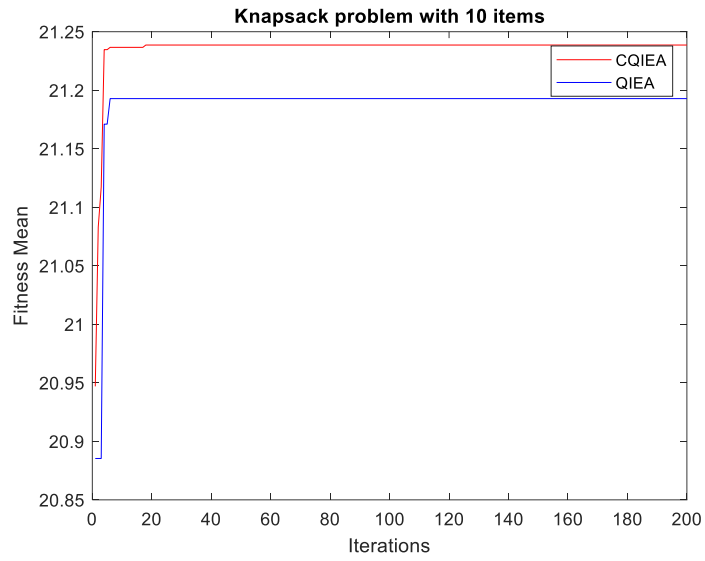


Figure 4-10: The comparison between the CQIEA and QIEA for Knapsack problem with 10 items

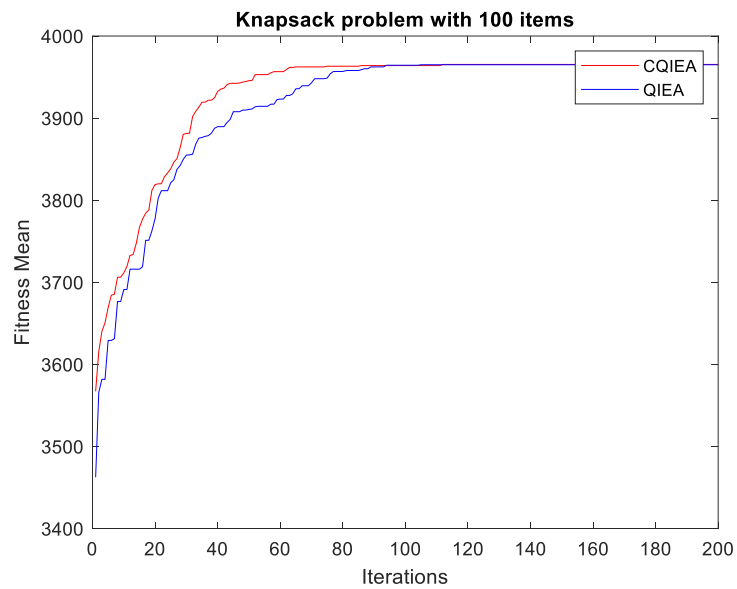


Figure 4-11: The comparison between the CQIEA and QIEA for Knapsack problem with 100 items

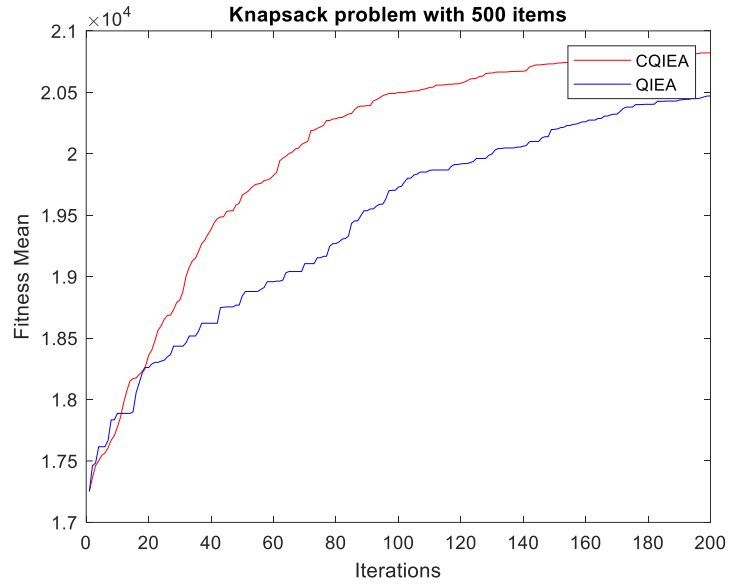


Figure 4-12: The comparison between the CQIEA and QIEA for Knapsack problem with 500 items

As it can be seen the CQIEA algorithm outperforms the QIEA algorithm both in the convergence time and the quality of the final solution. The changes in the chaotic rotation gate angles are shown in Figure 4-13 and Figure 4-14.

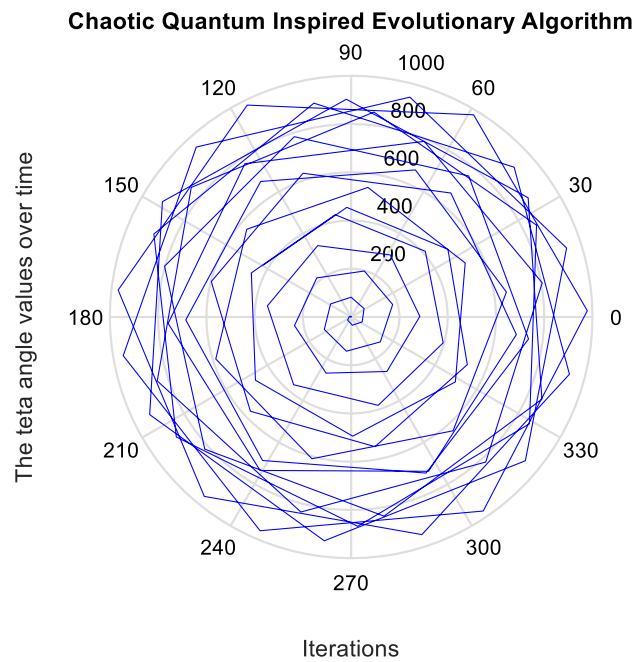


Figure 4-13: The changes of the chaotic rotation gate's angle over time in CQIEA in the unit circle

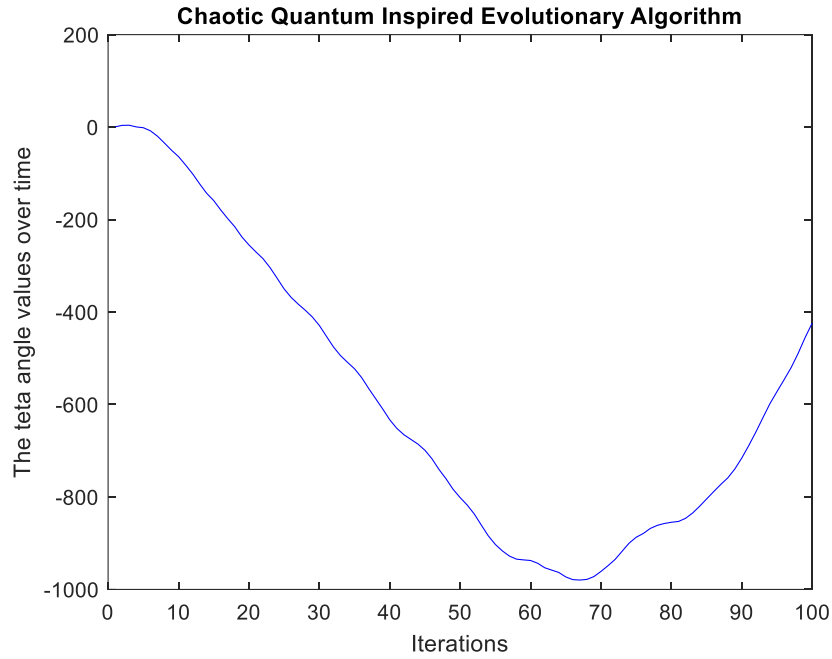


Figure 4-14: The changes of the chaotic rotation gate's angle over time in CQIEA

Table 4-3 provide the numerical analysis of CQIEA and QIEA performance on the Knapsack problem.

Table 4-3: The average results of applying CQIEA and QIEA algorithm on Knapsack problem for 20 runs

Algorithm	Knapsack		
	10 items with capacity = 36.30463	100 items with capacity = 4130	500 items with capacity = 21239
QIEA			
mean	21.1646	3819	19727
std	0.0490	0.0447	0.0533
CQIEA			
mean	21.2319	3966	20397
std	0.0144	0.0250	0.0237

The CQIEA and QIEA algorithms were applied to the TSP problem to visit New Zealand cities, both algorithms achieve the same results. Figure 4-15 to Figure 4-17 show the performance of CQIEA algorithm on the TSP problem for the New Zealand cities.

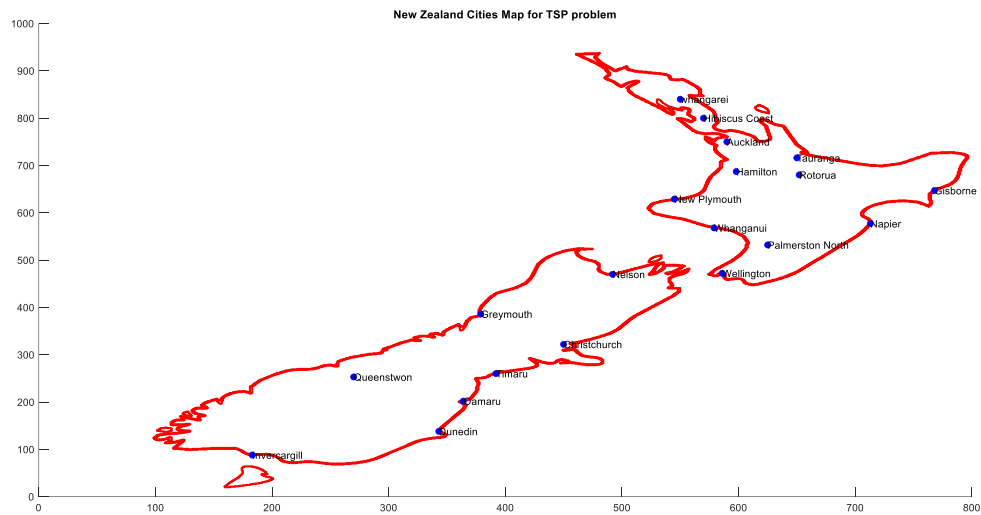


Figure 4-15: New Zealand cities map for TSP problem

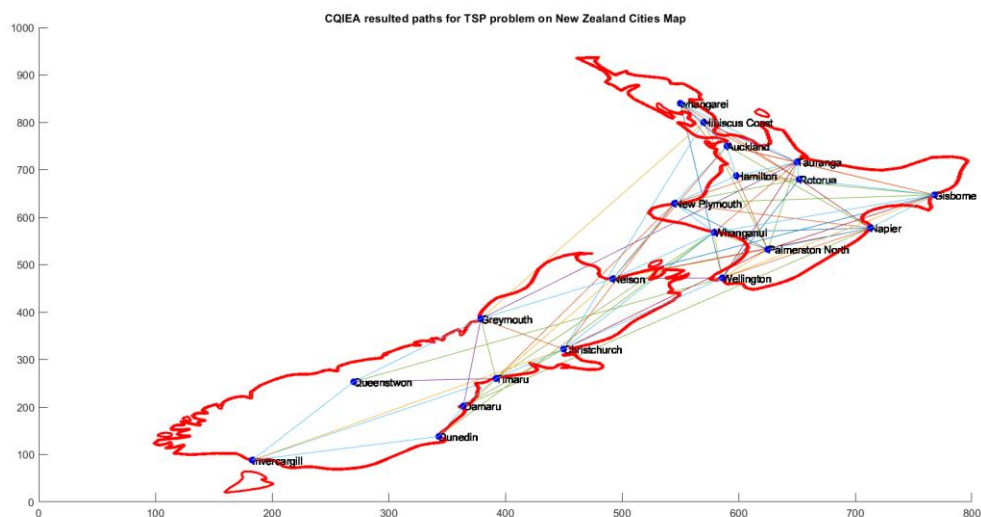


Figure 4-16: CQIEA resulted paths for the TSP problem on New Zealand cites map.

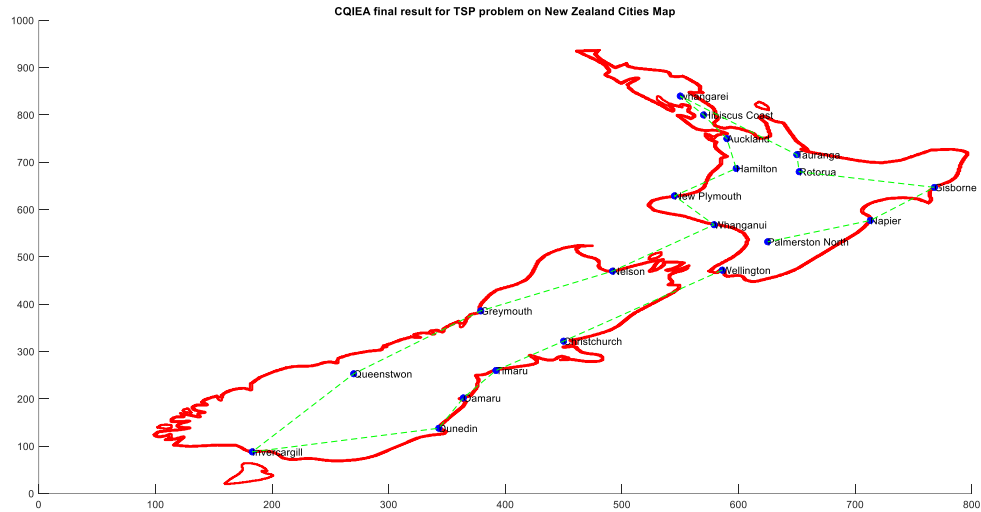


Figure 4-17: CQIEA result for the TSP problem on New Zealand cities map.

Table 4-4 depicts the comparative numerical analysis of applying CQIEA and QIEA to the continuous optimisation problem's benchmarks.

Table 4-4: The average results of applying CQIEA and QIEA algorithm to the continuous benchmark functions for 20 runs

Algorithm	Continuous optimisation			
	Sphere	Rastrigrin	Rosenbrock	Ackley
QIEA				
mean	2.4027×10^{-7}	0.0994	20.4150	1.0738
std	0.0915	0.0866	0.0575	0.0814
CQIEA				
mean	2.0108×10^{-7}	6.9273×10^{-4}	19.3936	0.1395
std	0.0480	0.0823	0.0743	0.0855

The above comparative analysis demonstrates the superiority of the proposed CQIEA algorithm over its former counterpart both in terms of convergence time and the resulting quality.

The CQIEA algorithm improvement can help to boost the proposed model learning mechanism in the following chapters. The convergence time and better-quality results are important capabilities that can be added to improve the EPUSSS framework and the quantum associative memory.

4.9 Chapter Summary

In this chapter, a novel search algorithm is proposed based on chaos theory and the quantum-inspired principals called Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA). The Chirikov chaotic map is used here to improve the convergence rate and the quality of the final solution of the Quantum-Inspired evolutionary algorithm-Using chaos theory to speed up the optimisation process. In this chapter, both discrete and continuous version of CQIEA has been proposed to serve as a learning mechanism to adjust SNN's weights, as an optimisation technique to improve proposed models' performance, and as a search mechanism for the proposed Quantum Associative Memory in the following chapters. The results of experiments prove the effectiveness of the proposed approach in both continuous and discrete environments.

The method is subject to a paper under construction.

Chapter 5 . A Novel Quantum Inspired Spiking Neural Network: QISNN

Abstract

In this chapter, a novel Quantum Inspired Spiking Neural Network (QISNN) is introduced based on the probabilistic spiking neuron model (Kasabov, 2010). The proposed QISNN follows the EPUSSS framework with a two-level hierarchical learning structure. The novel proposed QISNN hierarchical architecture consist of quantum neurons, quantum synapses, and quantum learning rule to perform prediction and classification tasks in a unified framework. The quantum information processing principles are used here to enhance the EPUSSS model performance and flexibility when dealing with Spatio-temporal data.

The following research question has been addressed in this chapter:

RQ5 “How can the biological probabilistic nature of a spiking neuron be demonstrated using quantum mechanics concepts to reinforce all levels of computations in SNN including quantum neurons, quantum synapses and chaotic quantum unsupervised-supervised learning?”

The material is subject to a journal paper in preparation.

5.1 Introduction

In this chapter, a novel Quantum Inspired Spiking Neural Network (QISNN) is introduced based on the probabilistic spiking neuron model (Kasabov, 2010). The proposed QISNN follows the EPUSSS framework with a two-level hierarchical learning structure. As it was mentioned in the previous chapter, although brain behaviour studies historically were led by disciplines such as biology, neuroscience, psychology and cognitive science, there were some

discussions about relating matter and mind using complex system physics and quantum physics (Hameroff and Penrose, 2014). To be more precise, according to the Penrose–Hameroff “Orch OR” model, the brain neurons process information, regulate membrane potential, and synaptic activities using quantum computation. The fact that the brain is a physical system composed of atoms whose behaviour can be explained by quantum physics, motivated the quantum-inspired brain-like model proposed in this chapter. The proposed QISNN model in this chapter combined the biological behaviour of neurons in the macrolevel (molecular level) and the quantum-based behaviour at the microlevel (atomic level) to perform both prediction and classification tasks in a unified framework similar to EPUSSS. The QISNN model extends the probabilistic spiking neuron model proposed by Kasabov (2010) by introducing quantum neurons, quantum synapse, and quantum learning rule in two levels of learning. The local learning in QISNN performs prediction while the global learning carries out the classification task. In the following section, a brief review of quantum spiking neural network ideas is provided.

5.2. Previous Work

Although recently both spiking neural networks and quantum computing attract many researchers’ interest, there is very little research done to couple these two areas together. Kristensen et al. (2021) proposed a quantum spiking neuron to improve SNN performance on custom-designed neuromorphic hardware based on Hamiltonian measurements. The authors introduced a spin system similar to the biological thresholding behaviour of spiking neurons with 3 states. States 1 and 2 are considered as input states and states 3 labelled as output. The chain interaction between the inputs and output cause energy differences among the possible states which results in the output excitation in presence of a certain state. Another research conducted by Sun et al. (2020), encodes the input information to spike trains using quantum

superposition concepts. Then a two-compartment spiking neural network with time convolution kernel synapse is introduced to process the encoded information to perform vision tasks. Chen et al. (2020) suggested a quantum spiking neural network called QSNN that uses quantum systems to measure the neurons voltage changes when a synapse receives temporal-space stimuli. Then the voltage of neurons compares to the threshold voltage in the observation time, through measuring quantum registers. To calculate the quantum probability, the authors used the quantum amplitude estimation (QAE) algorithm combined with the swap-test algorithm and applied the proposed method on the MNIST dataset to perform character recognition.

The above worthwhile research, partially deployed quantum computation to improve SNN performance. However, none of the above models suggested a complete quantum-based framework from network components to the learning algorithms. In this chapter, a novel Quantum Inspired Spiking Neural Network (QISNN) is introduced that not only consists of quantum neurons and quantum synapses but also has a two-level hierarchical quantum-inspired learning algorithm to perform both prediction and classification tasks.

The next section introduces the probabilistic neuron model which is the core idea in the QISNN proposed model.

In the probabilistic spiking neuron model (pSNM), probabilistic parameters determine spikes occurrence and the propagation of spikes across the synaptic connections (Kasabov, 2010). According to the pSNM model, a neuron n_i receives input spikes from pre-synaptic neurons n_j ($j = 1, 2, \dots, m$). The sum of all received inputs from m pre-synaptic neurons determine neuron n_i post-synaptic potential at time t , $PSP_i(t)$. Once $PSP_i(t)$ reaches a firing threshold $\vartheta_i(t)$, neuron n_i emits a spike. In pSNM, synaptic weights $w_{j,i}$ are established during learning using Thrope's rule (Thrope et al., 2001):

$$\Delta w_{j,i} = mod^{order(j)} \quad (5-1)$$

Where mod is a modulation factor in range (0,1), and $order(j)$ is a time-dependent order in which a spike from neuron n_j arrives at synapse $s_{j,i}$. This learning mechanism is a fast method that needs one pass data propagation to adapt the synaptic weights. In pSNN three probabilistic parameters determine neuron response to the input stimuli and its propagation along with post-synaptic connections:

1. $p_{cj,i}(t)$ which is the probability for emitting a spike from a pre-synaptic neuron n_j to a post-synaptic neuron n_i at time t through the synapse $s_{j,i}$. In other words, in pSNN model, each synaptic connection has $p_{cj,i}(t)$ that represents the structural and functional uncertainty observed in biological synapses. $p_{cj,i}(t) = 0$ shows that there is no synaptic connection and subsequently no spike can propagate to the post-synaptic neuron.
2. $p_{sj,i}(t)$ which is the probability of the synapse $s_{j,i}$ to take part in $PSP_i(t)$ after receiving a spike from a pre-synaptic neuron n_j . This probability stimulates the probability for ion channels to open or close during the release of neurotransmitters. $p_{sj,i}(t)$ probability can change during the learning process. In pSNN the assumption is that $p_{sj,i}(t) = 1$ ($j = 1, \dots, m$) for most synaptic connections unless there is a reason to change this parameter in the model.
3. $p_i(t)$ which is the probability for neuron n_i to emit a spike at time t when its $PSP_i(t)$ reaches the firing threshold. This probability represents the probability density for firing in a biological neuron. $PSP_i(t)$ is defined using the following formula:

$$PSP_i(t) = \sum_{p=t_0}^t \sum_{j=1}^m e_j g(p_{cj,i}(t-p)) f(p_{sj,i}(t-p)) w_{j,i}(t) + \eta(t-t_0) \quad (5-2)$$

Where e_j is 1 if neuron n_j emits a spike and 0 otherwise; probability $p_{cj,i}(t)$ can determine $g(p_{cj,i}(t))$ is 1 or 0; probability $p_{sj,i}(t)$ can determine $f(p_{sj,i}(t))$ is 1 or 0; t_0 is the time of the last spike emitted by n_i , and $\eta(t - t_0)$ represents decay in PSP. The pSNN is the generalization of the Integrate and fire (IF) model which if all its probability parameters set to 1 it behaves as IF neuron model.

In pSNN two approaches has been proposed to modify the probability parameter $p_i(t)$:

- a) Traditional probability theory which neuron n_i spiking activity changes the probability parameters. To be more specific, $p_{cj,i}$ ($j = 1, 2, \dots, m$) probability changes by Hebbian learning rule in this sense that if neuron n_i spikes within a time frame after a spike releases from n_j , $p_{cj,i}$ will increase by a small amount (learning rate); $p_{sj,i}$ ($j = 1, 2, \dots, m$) probability increases slightly every time that a spike is received at the synapse $s_{j,i}$ and a PSP contribution is emitted form this synapse.
- b) Quantum information approach which each of the probability parameters p_i , $p_{cj,i}$, and $p_{sj,i}$ is considered as a quantum probability represented by a quantum bit (qubit) to allow spike propagation through the network. The qubit states for pSNN probability parameters are defined by the wave function indicated by equation (4-5). The probability of the qubit to be in state “1” is calculated by $p(1) = \beta^2$. The probability amplitudes of spikes at all m synapses at time t is represented by the following matrix:

$$\begin{bmatrix} \alpha_1 | \alpha_2 | \dots | \alpha_m \\ \beta_1 | \beta_2 | \dots | \beta_m \end{bmatrix} \quad (5-3)$$

To change the qubits corresponding to p_i , $p_{cj,i}$, and $p_{sj,i}$ probabilities in pSNN according to spike propagation within a time interval in the connected neurons n_i and n_j the quantum rotation gate operator is applied:

$$\begin{bmatrix} \alpha_i^j(t+1) \\ \beta_i^j(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^j(t) \\ \beta_i^j(t) \end{bmatrix} \quad (5-4)$$

The qubits will be initialized with a normal distribution with two states 0 and 1, then the quantum rotation gate will change the probability of seeing 0 and 1. All the 2^m combinations of incoming spikes to neuron n_i at time t is represented by an m -qubit vector $p_{cj,i}$ ($j = 1, 2, \dots, m$).

The proposed QISNN is an extended version of pSNN (Kasabov, 2010) which is described in the following section.

5.3. The Proposed Quantum Inspired Spiking Neural Network (QISNN)

As it was mentioned before, the QISNN follows EPUSSS architecture and the two-level hierarchical learning structure. In the QISNN, the neuron model, synaptic connections, and learning rules are a combination of biologically plausible models and quantum-inspired models introduced in chapters 3 and 4, respectively. Therefore, it is fair to say that the QISNN framework is a brain-like quantum-inspired model.

Like EPUSSS, QISNN has local and global learning mechanisms:

- a) In the local learning mechanism, the input data dynamics is learnt and predicted in a semi-supervised manner using the STDP rule and the chaotic quantum learning rule.

- b) In the global learning approach, the classification result at end of each episode of learning is used to adjust the neuron's synaptic weights utilising deSNN structure combined with the chaotic quantum learning rule in a supervised manner.

Although the learning mechanisms of QISNN is fundamentally similar to EPUSSS, the learning algorithms, neuron's firing behaviour, and post-synaptic connections are controlled and governed by quantum computation models.

QISNN like EPUSSS architecture has the same three major modules in its framework: the encoder module which uses BSA algorithm that encodes input data into spike trains, the 3D SNN reservoir module in which the local learning mechanism of QISNN governs the synaptic efficacy in a semi-supervised way, and the output module which deploys the global learning mechanism of QISNN combined with modified deSNN model to update synaptic efficacy in a supervised way. QISNN uses spiking neuron models such as LIF and Izhikevich combined with a quantum model to control neurons' firing behaviour.

QISNN connectivity both in the 3D SNN reservoir and the deSNN output layer is completely akin to EPUSSS. The QISNN model learns from time-series data to predict the next values of the input data in a certain time-window in the reservoir. Predictive modelling is achieved through a semi-supervised local learning mechanism using the error of prediction in a certain time-window and rule that adjust the efficacy of synaptic weights of the reservoir neurons. The output neurons inspired by the deSNN model's concept are created incrementally for each training sample to associate the input data with a class label. For this purpose, all the input neurons and reservoir neurons (or just input neurons) are connected to output neurons using the RO method and their synaptic weights are adjusted based on the QISNN global learning rule using a neuromodulatory signal that is generated by the results of classification in the training phase. The principle behind the algorithm is to perform both prediction and

classification in a two-level hierarchical learning structure. In the proposed model the input neurons contribute strongly to both local and global learning. In local learning, the system learns to predict the input data dynamics through modifying the oscillatory rhythmic patterns of neuronal firing activities. A sliding time-window allows QISNN to capture the firing behaviour of the input neurons for a certain time course in which the input data is exposed to the network. The input neurons have recurrent connections from the last layer of reservoir neurons that help them to learn the activity of the reservoir neurons and influence other input neurons to have optimal learning like the cooperative learning in the brain. Then by cross-matching input neurons firing behaviour with the input data dynamics in one (several) time unite(s) ahead the error will be calculated and backpropagated to modify the synaptic and hypersynaptic connections. This process is similar to synaptic plasticity based on local signals in the brain.

At the end of each episode of learning corresponding to the presentation of the entire input pattern with T lasting duration to the network, a neuromodulatory signal from the output neurons is generated based on the target labels. Using the QICEA learning rule the generated neuromodulatory signal helps to reward or penalise the synaptic and hypersynaptic connections efficacy. This process is the global learning mechanism of QISNN which resembles the feedback loop learning from external stimuli in the brain. The QISNN architecture and its learning principles can be seen in figure 5-1.

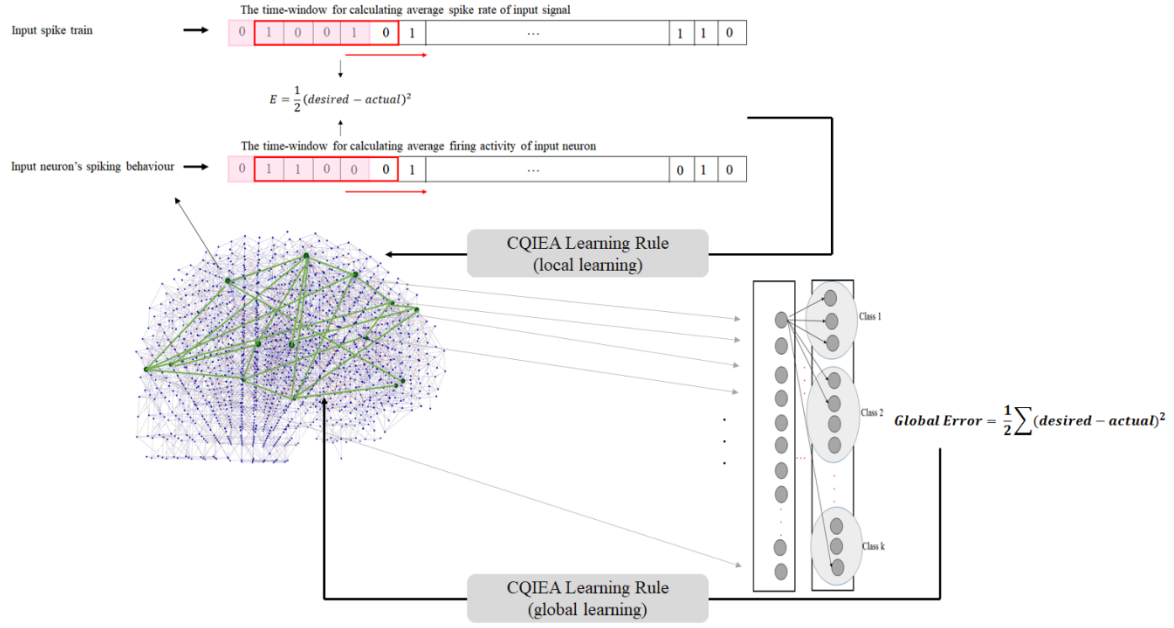


Figure 5-1: The QISNN two-level hierarchical learning framework

5.3.1 Quantum Local Learning

Inspiring by pSNN model, a quantum neuron model is introduced for neurons and their synaptic connections. In this model, the spiking neural network will be presented by two quantum elements: 1) quantum neuron whose firing probability is controlled by a quantum wave function, 2) quantum post-synaptic connections which the probability of forming a connection between neuron n_i and n_j is controlled using a quantum wave function.

A Chaotic quantum rotation gate is used to change the probabilities of firing in a quantum neuron and forming a synaptic connection in quantum post-synaptic connections according to two-level observation. At the first level observation quantum neuron probability of firing is modified using the average firing rate in a predefined time window that slides over the input spike trains. If the spiking activity is not like the firing average of the input spike train in a time window, the quantum gate operator will apply to change the firing probabilities to decrease or increase them.

The predictive accuracy is used as the fitness function to evaluate the quantum solution quality.

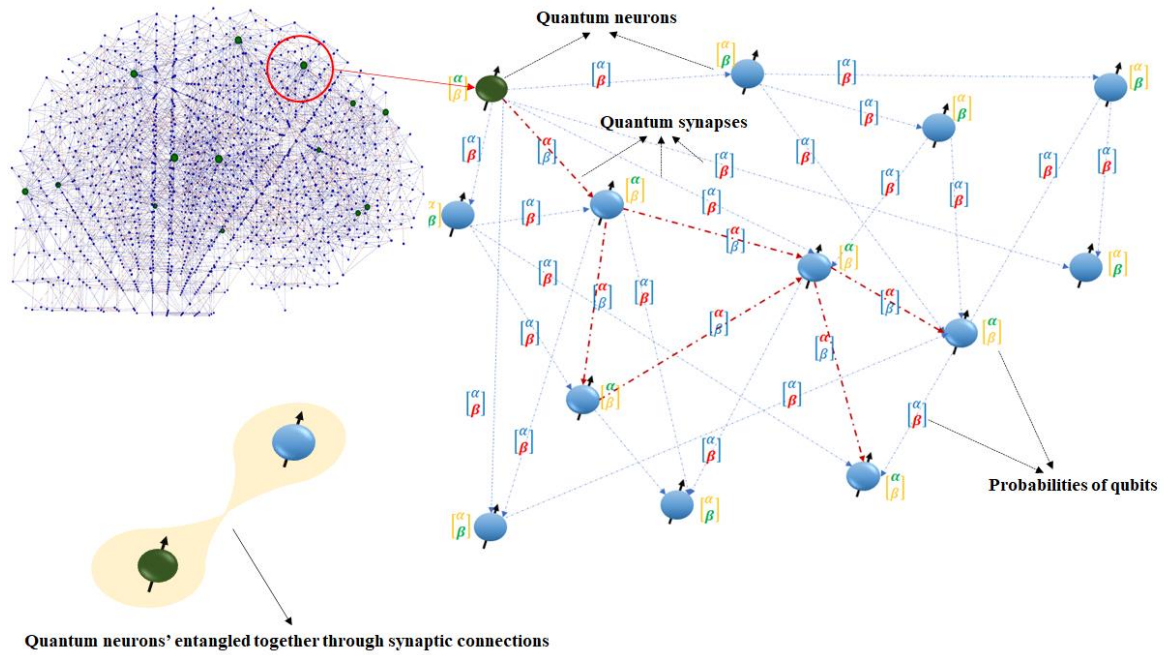


Figure 5-2: The role of quantum state probability amplitudes in the learning mechanisms of QISNN

Since the use of a single qubit cannot underline the power of quantum computation in information processing in spiking neural networks, the quantum entanglement can help to highlight the correlation amongst the quantum neurons in the process of emitting spikes through the synaptic connections.

In the QISNN, the quantum neurons are entangled together through the small-world connectivity. The CQIEA learning rule will control the time that a neuron propagates the information, and it also helps to choose the synapses that contribute to the learning. This approach can help to reduce multiple synaptic paths for propagating information to a few more influential paths that can improve the learning results by reducing noisy paths.

5.3.2 Quantum Neurons and Quantum Synapses

Figure 5-3, shows both quantum neurons and quantum synapses qubits and their wave functions.



Figure 5-3: The local learning quantum neurons and synaptic qubits and the quantum wave function for the probability amplitudes for the entangled neurons.

The probability of amplitudes of both quantum neurons and quantum synapses will be controlled according to the prediction results by CQIEA. At the observation time (end of each time window) the error between the actual firing rate and the input neurons' firing rate will be used as the fitness value to change the states of the qubits towards a better solution in an iterative way. The neurons membrane potential will be calculated using the LIF model, however, it is the CQIEA algorithm that decided if the neuron which has reached the firing threshold can emit the spikes to the post-synaptic neurons or not. Also, although there might be several post-synaptic neurons connected or entangled with the pre-synaptic neuron, only the synapses that have the higher probability amplitude will be chosen to emit the spike to the target neurons. In this way, not only the time of the spike but also the path of information propagation will be improving at each iteration to boost the learning process. This approach helps to create a learning path according to the observed outcome.

5.3.3 Quantum Global Learning

In the level 2 observation (global learning), post-synaptic connections probabilities will be modified according to the result of the classification of the input patterns in the deSNN layer using the CQIEA learning rule.

By introducing spin quantum number to each quantum neuron, a diamagnetic field will be created for those neurons which contribute to form a similar path for similar input patterns and a paramagnetic field for paths for different input patterns. In this way, all the quantum neurons that contribute to forming a certain path for similar input patterns are entangled together. The spin quantum number will be updated based on the spiking neuron activities for input patterns at the end of simulation times for each input sample. Therefore, through the learning time, those neurons that contribute to the same class input will assign the same spin number. This mechanism helps us to extract knowledge of spiking behaviour during learning and provide a clear synaptic connection path for better pattern recognition.

To mimic the hippocampus role in biological learning, a neuron that is connected to presynaptic inhibitory neurons, will not spike even though the probability of firing reaches the threshold. This approach guarantees to balance excitatory and inhibitory activity in the network.

The synaptic weights between pre- and post-synaptic neurons will be updated according to the STDP rule and the probabilities of firings and synaptic connections will be updated using CQIEA. Figure 5-4 provide a schematic illustration of the quantum neurons and quantum synapses in the global learning.

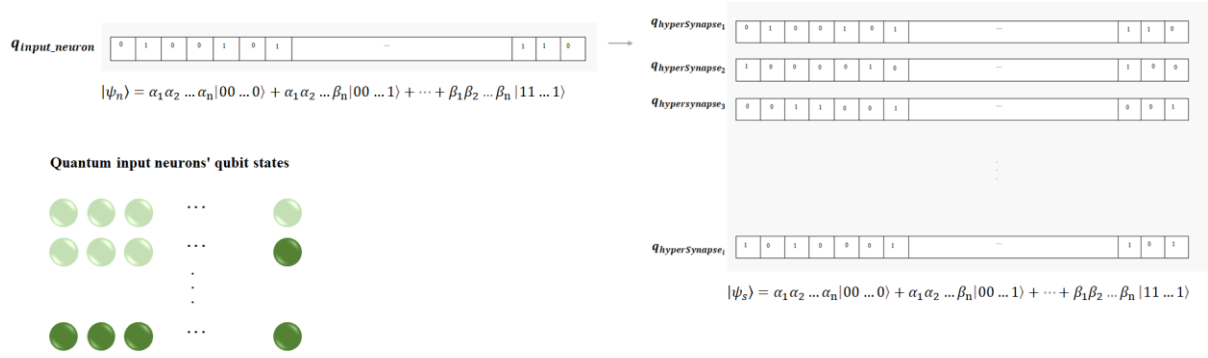


Figure 5-4: The global learning quantum neurons and synaptic qubits and the quantum wave function for the probability amplitudes for the entangled neurons

The same process performed in the local learning will be followed for the global learning too. The only difference is that in global learning the error for the observation time will be the classification error at the end of each episode of training. Figure 5-5 shows the QISNN pseudocode.

Quantum Inspired Spiking Neural Network (QISNN)

- 1: Generate 3D-SNN network including input neurons I , Reservoir neurons R using neurons coordinates (e.g., Talairach map for EEG data).
- Create QISNN network connectivity by establishing hypersynapses and synaptic connections
- 2: amongst input neurons and between input neurons and reservoir neurons, respectively using small world connectivity algorithm (see chapter 2).
- 3: Set 80% of synaptic connections excitatory (+1) and 20% of synaptic connections inhibitory (-1).
- Set neuron firing thresholds ϑ for excitatory and inhibitory neurons (if at least one of the
- 4: postsynaptic connections is inhibitory the neuron is considered as inhibitory neuron).
- 7: Initialise random weights for synaptic and hypersynaptic connections.
- 8: Set *time – window* value (e.g., 10 time units) and *max _iteration* value (e.g., 50)
- 9: Encode all input pattern to spike train using BSA algorithm (see chapter 2).
- 10: while $it \leq \text{max_iteration}$
- 11: for each input pattern P_s
- 12: for each input neuron I_i

```

13:         while  $l \leq \text{length}(\text{time} - \text{window})$ 
14:             Propagate input spikes to reservoir neurons using LIF neuron model (Eq. 3-1)
15:             Adapt synaptic and hypersynaptic weights according to STDP rule (Eq. 2-12).
16:             Calculate  $E_{local_i}$  (Eq. 3-2).
17:             Update weights using CQIEA.
18:             Set the synaptic weights for inactive neurons to zero.
19:              $\text{time} - \text{window} += \text{time} - \text{window}$ 
20:     Initialise deSNN parameters:  $Mod$ ,  $D$ ,  $C$ ,  $\alpha$ , and  $sim$  (similarity threshold)
21:     for each input pattern  $P_i$ 
22:         Create a new output neuron  $i$ 
23:         Initialise synaptic weights  $w_i(0)$  using RO learning rule denoted in Eq. 2-14
24:         Calculate  $PSP_{imax}$  using Eq. 2-16
25:         Calculate the firing threshold  $Th_i$  for neuron  $i$  using Eq. 2-17
26:         if the new wight vector  $w_i$  is similar to an existing output neuron using Euclidean distance
            similarity measure and  $sim$  threshold
27:             if the desired class labels of the similar weight vectors are different
28:                 Generate the corresponding  $d(t)$  reward signal according to Eq. 3-9.
29:                 Update the synaptic weights  $w_i$  for successive spikes on the corresponding synapses
                    using CQIEA rule.
30:                 Add new output neuron to the network
31:             else
32:                 Generate the corresponding  $d(t)$  reward signal according to Eq. 3-9.
33:                 Update the synaptic weights  $w_i$  for successive spikes on the corresponding synapses
                    using CQIEA rule.
34:                 Merge the two neurons by averaging their threshold and synaptic weights
35:         Calculate energy function  $E$  (Eq. 3-11)
36:         Adapt all synaptic weights according to Eq. 3-13.

```

Figure 5-5: The QISNN pseudocode

In Figure 5-6, the modular schematic of the QISNN framework is illustrated.

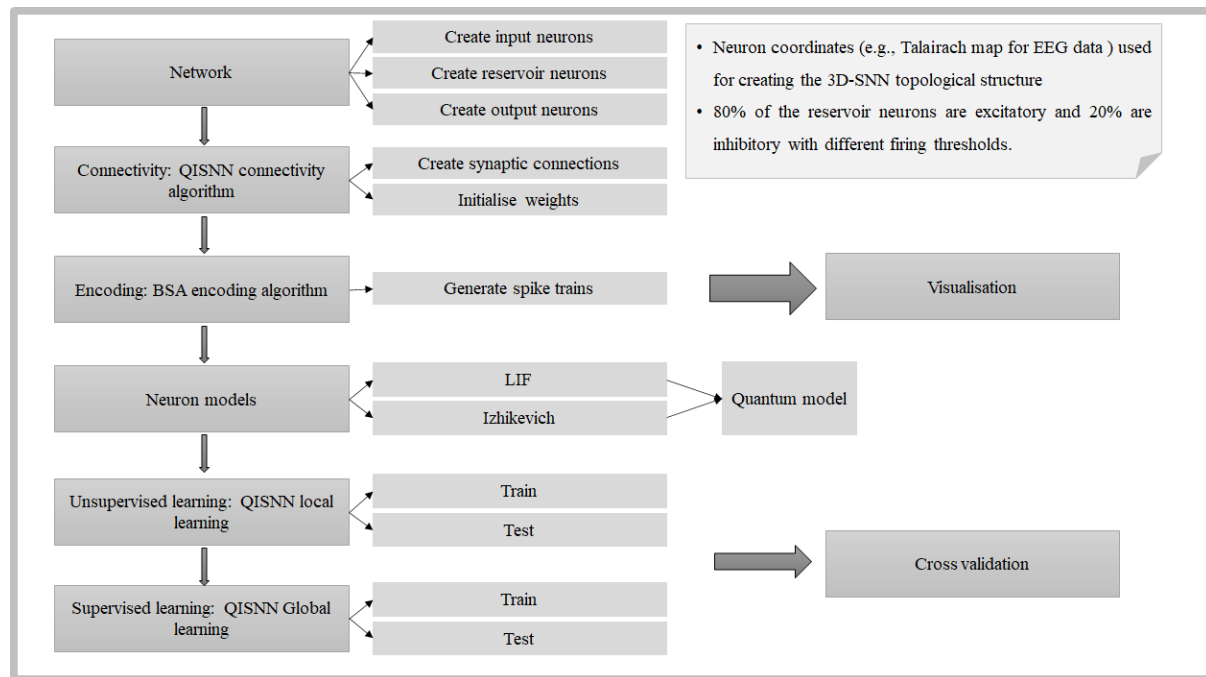


Figure 5-6: The QISNN implementation modular schematic

The proposed QISNN's needs to be investigated by performing more experiments with various data. Also, since the model is a hybrid of the SNN model and quantum computation concepts, there is a potential for designing combined neuromorphic and quantum computing hardware.

Quantum computers, which was first invented in 1998, are immensely difficult to engineer, build and program. Quantum computers are limited errors arising from noise, faults, and loss of quantum coherence in their operations. The loss of coherence caused by vibrations, temperature fluctuations, electromagnetic waves and other interactions with the outside environment can cause damage to the quantum computer's components. Thus, the current quantum computers are not quite reliable even for programs with modest execution times. On the other hand, neuromorphic hardware, which adapts the SNN properties in its architecture, is proven to be fast and energy-efficient and can support high-performance applications. The proposed models in this thesis have the potential to be implemented on quantum computers, neuromorphic hardware, and even hybrid hardware that supports both SNN and quantum

computing architecture. This can be an exciting and promising area to be explored further in the future.

5.4. Conclusion

This chapter demonstrates a quantum-inspired framework for a brain-inspired SNN that uses quantum computations in combination with biologically plausible learning rules to perform both prediction and classification tasks at the same time within a unified model. The proposed QISNN framework is an improved version of the EPUSSS framework to solve the polynomial time complexity of learning mechanisms.

Chapter 6 . Quantum Inspired Associative Memory for Spiking Neural Networks (QIASM-SNN) on Spatio-temporal Data

Abstract

Associative memories (AM) relate to learning some patterns and recalling them from partial information. This is a typical brain function. Creating a neural network for associative memory for static images is what the current state-of-the-art is (see John Holland). This chapter proposes new types of AM for Spatio-temporal data, which is exactly what the brain does, rather than AM for single images. So, it is about Spatio-temporal associative memories (STAM). Two methods are proposed and subject to pending publications: Brain-inspired and STAM Quantum inspired STAM.

In this chapter, the following research questions are addressed:

RQ6 “How can Spatio-temporal spiking patterns be stored and recalled by the spiking neural network for recognition and prediction tasks in the presence of noisy or partial incomplete data by introducing a novel approach using associative memory and quantum computation concepts?”

RQ7 “How can the proposed Quantum Associative Memory for storing and recalling spiking patterns be improved in terms of time and space efficiency using the proposed CQIEA algorithm?”

6.1 Introduction

A human can learn and understand many concepts by finding the relation between apparently unrelated data. In other words, our mind is capable to recognize the patterns in data and label them to classify or using them to predict future events. This ability of our brain is called associative memory.

Inspiring by this human brain capability, associative memory has been introduced to the machine learning field to memorize information and retrieve them from partial or noisy data. Associative memory in neural networks is an unsupervised technique that was first proposed by Hopfield (1982). Inspiring by Quantum Associative Memory proposed by Ventura et al. (2000) and Techapet Najafa and Nana Engo (2018), The Quantum Associative Memory for Spiking Neural Network to store produced patterns by the network and recall them in the presence of partial noisy input data is introduced in this chapter. Before describing the proposed method some basic concepts about quantum mechanics is explained in the following section.

6.2 Previous Work

Associative memory is the ability to learn and remember the relationship between different objects or concepts such as remembering the name of someone we have just met or the aroma of a particular perfume (Suzuki, 2005). In artificial neural networks, associative memory is referred to the ability to retrieve the most similar pattern to the stored memory vector in the network.

Ezhov et al. (2000), discussed a model of quantum associative memory which generalizes the completing associative memory proposed by Ventura and Martinez. The authors suggested the use of a distributed query of general form. Ventura and Martinez (2000), combined quantum

computation with classical neural network theory to produce a quantum associative memory (QuAM). Their method was based on Grover's well-known algorithm for searching an unsorted quantum database. Zhou et al. (2012), introduced a storage method for multiple patterns by constructing the quantum array with the binary decision diagrams. They used the nonlinear search algorithm to increase the pattern recalling speed. Techapet Najafa and Nana Engo (2012), proposed a quantum associative memory model using the quantum matrix with binary decision diagram and the nonlinear search algorithm. Their model can retrieve one of the sought states in the multi-values retrieving scheme. According to the authors, their model is faster and robust to the noise in comparison to Grover's algorithm. Techapet Najafa et al. (2015), proposed an improved quantum associative algorithm with the distributed query. The authors optimized data retrieval of correct multi-patterns by modifying the probability amplitudes of the memory patterns states. Techapet Najafa and Nana Engo (2018), introduced a model of Quantum Associative Memory called QAMDiagnos to diagnose four tropical diseases with similar signs and symptoms. Their proposed memory model can distinguish a single infection from a polyinfection. This model is a combination of the improved version of the original linear quantum retrieving algorithm proposed by Ventura and the non-linear quantum search algorithm. Hunter et al. (2008), compared associative memory function in a spiking neural network with a simple artificial neural network and investigate the biological plausibility of the model to improve the recalling process within a sparsely connected network. Agnes et al. (2012), assembled a synaptic connectivity model on a spiking neuron network to build a dynamic pattern recognition system. In their model, neurons' responses to different current injections are mapped onto a subspace using Principal Component Analysis to form different trajectories. Hu et al. (2015), proposed a spiking associative memory model to encode different memories using different subsets of encoding neurons with temporal codes. The authors used the STDP learning rule to form associative memory by modification of the

synaptic connections between the input layer and hidden layers, and recurrent connections of hidden layers, respectively. Haga and Fukai (2019), introduced a memory system for a neural network based on an attractor network which is a group of connected nodes that display patterns of activity and tend towards certain states. They applied the concept of excitatory and inhibitory nodes to their proposed network to mimic the role of the hippocampus in balancing the network to form new associations. The problem with their approach is using a random sampling of the possible outputs instead of a systematic review of every possible combination.

All these worthwhile works have problems with the computer physical storage capacity, the computation time to retrieve a certain pattern, and deploying an efficient search mechanism for all possible output combinations.

6.3 Brain-inspired STAM based on NeuCube

6.3.1 Introduction

The brain can function as an associative memory (AM) to recall previously learned spatio/spectrotemporal (ST) information even though only partial or related information is provided. In this case, only part of the brain areas involved in the previous learning is activated with the provided input data. It then gets other parts activated through polychronisation and synfire chains until the full information is recalled. Inspired by this amazing brain ability, here we propose a method for the realisation of Spatio-temporal associative memory (STAM) in a brain-inspired spiking neural network (SNN) architecture. We call the proposed method STAM-SNN. It is implemented and experimented here on several STAM case studies, including EEG, fMRI and financial temporal data, manifesting robustness to missing temporal or Spatio-temporal components in the recall data. Future studies will involve the realization of

different types of STAM manifested in the brain for the creation of brain-inspired artificial intelligence.

6.3.2 STAM on NeuCube Using EEG Data

Used data: the Wrist Movement EEG data; length of the EEG signals 128 time points; 3 classes; 20 samples of each class.

Training the SNNcube with all 60 samples of data.

Testing the recall accuracy with all 60 samples, but with partial temporal data (from 90% down to only 5% from the initial part of the EEG signals).

The accuracy achieved: 100% recall of samples and classification of the movements for 5% (just 7 time points from the beginning of each EEG sample) or a higher percentage of the input temporal information used. The accuracy starts to decline from 4% down to 2 % of data used (just 3 time points at the beginning of the signals).

NeuCube SNN parameters: Encoding: SF; Classifier: Mod=0.9; Drift=0; k=1; Firing threshold: 0.5; STDP rate 0.01; training iterations 1; LDC probability 0; Refractory time 6; Potential leak rate 0.02 (see Figure 6-1).

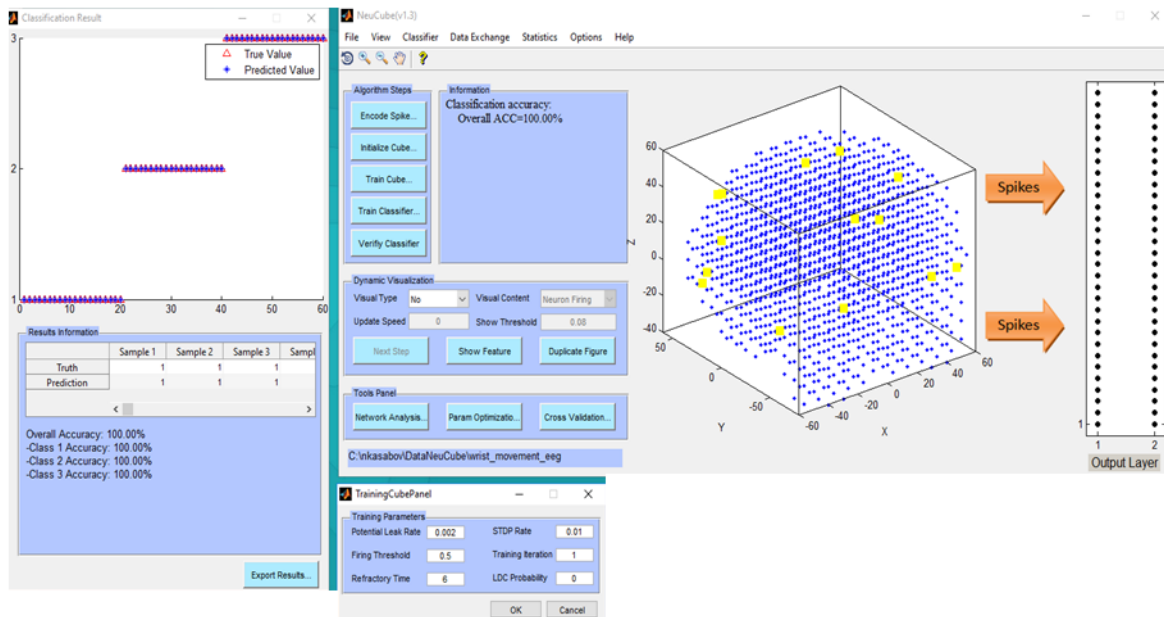


Figure 6-1: Training on full data and recall from 100% down to only 5% of the input temporal data results in 100% classification accuracy.

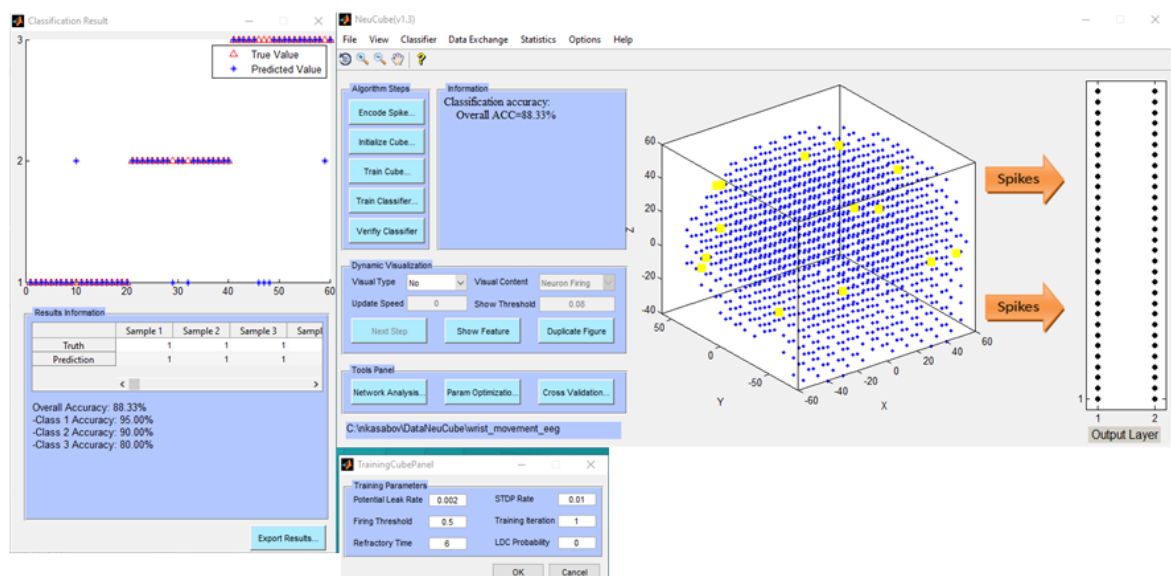


Figure 6-2: Training on full data and recall on only 2% of the temporal input data results in less than 100% accuracy of recall, but still 88% classification accuracy is achieved.

Training with full data (14 channels, full temporal sequences) and validation with 13 EEG channels, full temporal sequences): 100% accuracy (see Figure 6-3).

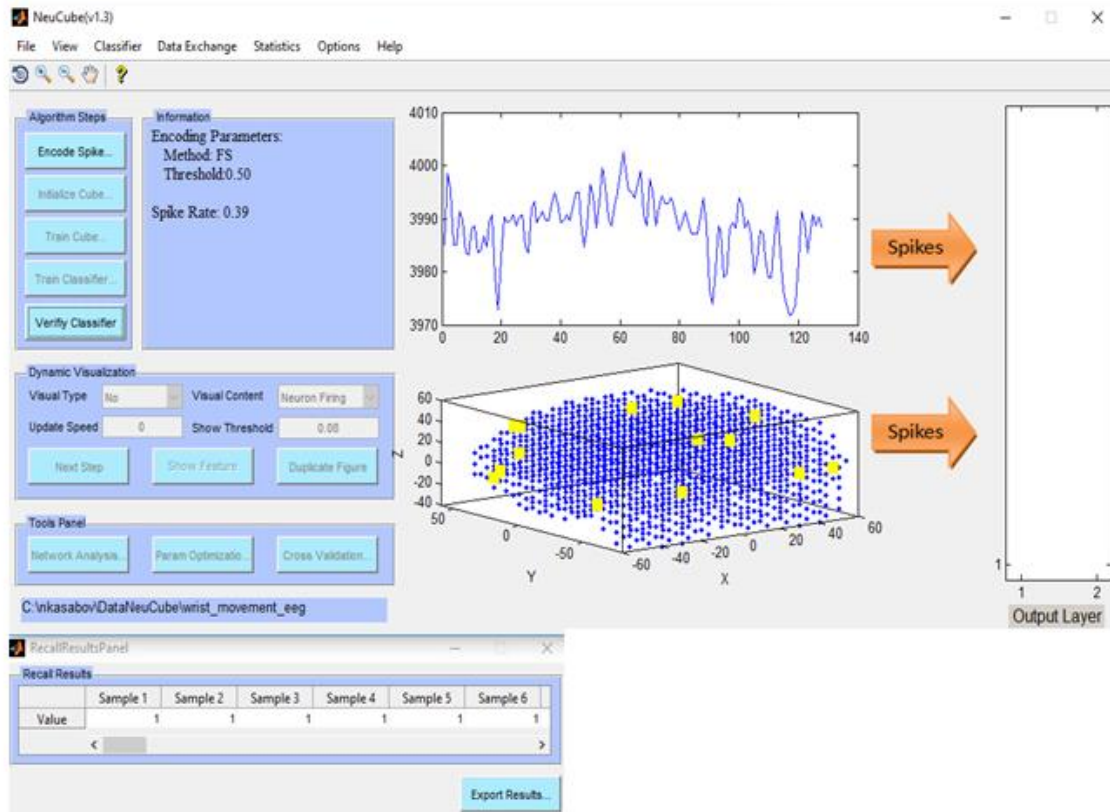


Figure 6-3: Training with full data (14 channels, full temporal sequences) and validation with only 13 EEG channels, full temporal sequences): 100% accuracy.

6.3.3. Challenges For Further Development of STAM-SNN

A challenge in STAM-SNN is to design a system that associates different stimuli during learning so that if one of them is available the other one can be recalled. Can for example a STAM-SNN model learn brain data from *synesthetic* subjects who can hear music when they see a colour? The challenge can be addressed if prior knowledge is used about brain structural and functional pathways, along with the stimuli data and the corresponding ST data from a subject or multiple subjects. At present much is known about structural connectivity and functional pathways during perception (Carter et al, 2009; Stam, 2004; Hellwig, 2000; Braitenberg and Schutz, 1998). This prior knowledge can be used to initialise the SNN Cube connectivity before training.

An open question is how to enter sound-, image and brain response data (e.g., EEG, fMRI) as associated Spatio-temporal patterns into appropriately dedicated groups of neurons? Entering stimulus signals to different parts of the brain is a major principle used in neuroprosthetics (Nicolelis, 2012) so that damaged parts of the brain are ‘skipped’ (Paulun et al, 2019). Could experiments with the STAM-SNN framework provide some insights and ideas about new types of neuroprosthetics?

The STAM-SNN framework makes it possible to use neuromorphic sensory systems, such as the silicon retina DVS camera (Delbruck et al, 2007; Lichetnstein and Delbruck, 2008). It could be done by converting analogue input data into spike trains based on fast AER and then entering these spike trains into appropriate neuronal structures of the model. A trained STAM-SNN model can be developed for the recognition of obscured moving objects.

6.4 The Proposed Quantum-Inspired STAM for SNN

By introducing quantum computation to the machine learning field, especially artificial neural networks, the performance of these techniques has been boosted up both in terms of speed and storage capacity. Quantum associative memory can provide a storage capacity of $O(2^n)$ patterns using only n neurons, which is enormous memory storage, while the conventional associative memory approaches struggle with severe storage restrictions (Ventura et al., 2000).

In this chapter a novel quantum associative memory scheme for spiking neural networks is proposed to perform a fast, online pattern recognition for classification and prediction tasks just by partial pattern exposure.

For the implementation of the quantum associative memory for spiking neural networks, the network requires the ability to store patterns in a medium and the ability to recall those patterns later. In previous work on quantum associative memory, the information needed to be presented

in a binary code to construct quantum states that can be stored or retrieved by the quantum algorithms. By embedding the quantum associative memory into EPUSSS or QISNN models, the spike encoding module of these frameworks will convert the information into the spike trains which is a binary representation. Each input neuron in the EPUSSS or QISNN models can act as a register to store memory of the produced patterns by its locally connected reservoir neurons. To find a pattern in the quantum associative memory, the system should be measured and collapsed with the near certainty to the basis state which corresponds to the sought pattern. To this end, the retrieving quantum algorithm inverts the phase of desired basis state and all the other basis states using the quantum gate operator in a way that the probability amplitude of the desired basis state increases to near unity and the probability amplitude all the other states decrease to near zero.

The proposed local learning in EPUSSS or QISNN simulates the neocortex neurons that are responsible for long-term memory storage and uses excitatory and inhibitory neurons with different types of neurotransmitters to mimic the regulatory role of the hippocampus in forming cognitive patterns in a distributed quantum memory all over the network. Using local learning algorithms introduced in chapters 3 and 5, the memory forms in the proposed network. Then, by introducing quantum memory storage for each input neuron (qubit memories) the learnt patterns can be retrieved in an “Ergodic” manner meaning that the system visits all its possible states to find the pattern that matches to the input (partial or noisy) information.

In the proposed quantum associative memory, we assume that a set of p of m binary patterns of length n are stored. Which n is the number of neurons. We consider the problem of associative pattern completion – learning to produce one of the full patterns when presented with only a partial pattern. Recall a given pattern in an associative fashion allows fill in a reasonable guess as to the rest of the pattern so that even a partial pattern that may never have

been seen during the learning of the pattern set p can be retrieved from the stored memory. Quantum associative memory has a capacity of $O(2^n)$ for storing patterns with length n using n neurons.

In the novel proposed quantum associative memory, the learnt patterns which are produced during the local learning of either EPUSSS model or QISNN by the input neurons (input twin neurons) will be stored as qubits in the quantum memory. In this way, the associative memory is integrated into both models and after local learning when the input spike trains dynamic is captured and learnt by the network, the patterns can be recalled in a more time-efficient manner.

After storing patterns in the local learning phase of the EPUSSS model or the QISNN model two quantum search algorithms can be used to retrieve the sought pattern, Grover's algorithm and the CQIEA algorithm. The schematic of the proposed quantum associative memory for spiking neural network is shown in Figure 6-4.

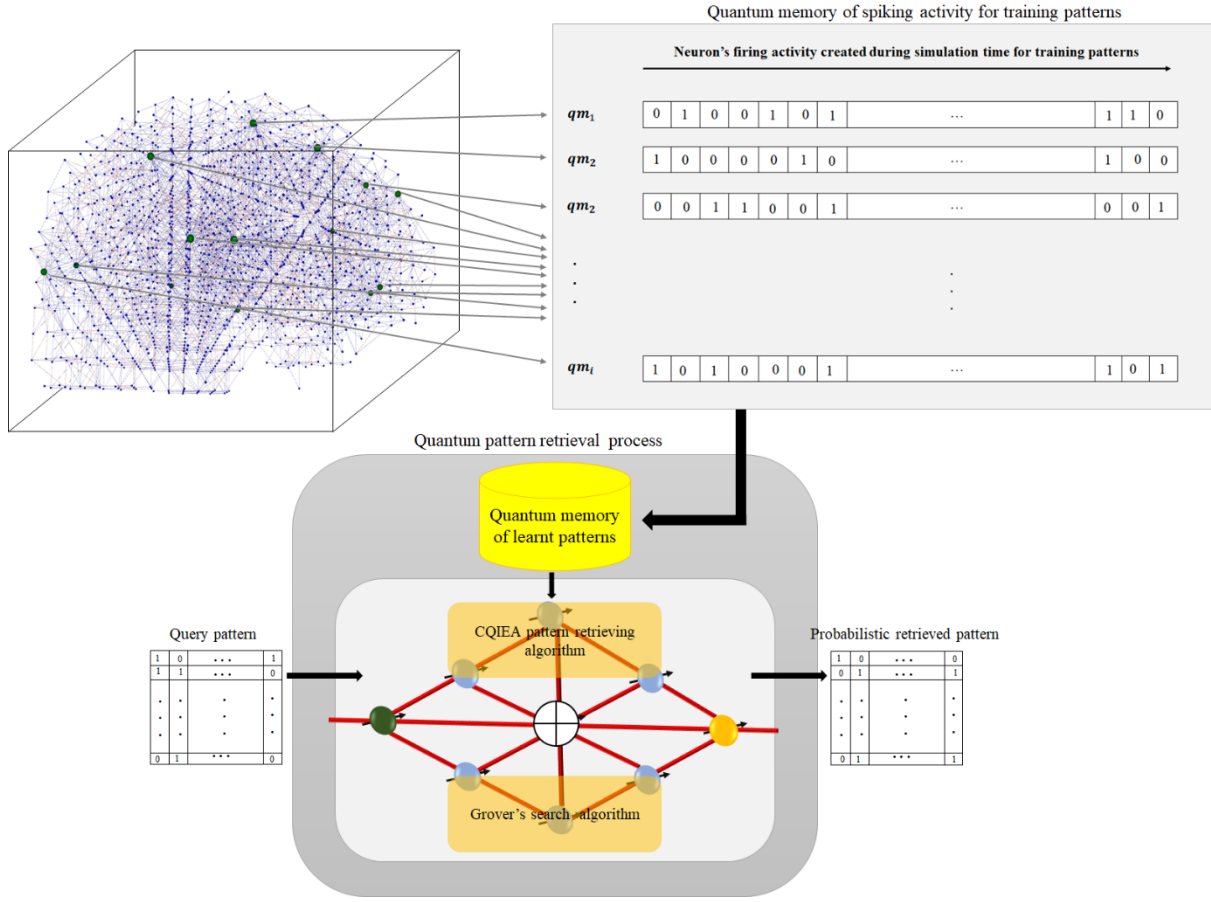


Figure 6-4: The pattern retrieval mechanism in the proposed spiking quantum associative memory

6.4.1 The Grover's Algorithm

The Grover's algorithm (Grover, 1996; Grover, 1998) is a quantum counterpart of classical which offers $O(\sqrt{N})$ time complexity for searching an unstructured database containing N items to find a specific pattern.

Assume that we have an unstructured database containing N patterns binary patterns $f: \{1, 2, 3, \dots, N\} \rightarrow \{0, 1\}$, the search problem is to find $x^*: f(x^*) = 1$. The Grover's algorithm performs the search process in two iterative steps, phase inversion and inversion about the

mean. Grover's algorithm starts with initialising the probability of amplitudes for qubits all to equal superposition states:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \alpha_x |x\rangle \quad (6-1)$$

Afterwards the amplitude α for the sought pattern x^* is changed using the phase inversion operator:

$$\text{Phase inversion} \begin{cases} \sum_{x \neq x^*}^{N-1} \alpha_x |x\rangle & \text{if } x \neq x^* \\ \sum_{x=x^*}^{N-1} -\alpha_x |x\rangle & \text{if } x = x^* \end{cases} \quad (6-2)$$

In the phase inversion, the amplitude of the sought pattern is inverted and the amplitudes of the rest of the patterns remain unchanged.

In the next step, the inversion about the mean operator is applied to flip the amplitudes of the patterns about the mean amplitude.

$$\text{inversion about the mean} = \sum_x^{N-1} (2\mu - \alpha_x) |x\rangle \quad (6-3)$$

Where μ is the mean amplitude and is calculated as follows:

$$\mu = \frac{\sum_{x=0}^{N-1} \alpha_x |x\rangle}{N} \quad (6-4)$$

The phase inversion and the inversion about the mean steps are repeated until the sought pattern is found.

Figure 6-5 illustrates the mathematical representation for the phase inversion and inversion about the mean operators of Grover's algorithm. The iterative process of phase inversion and inversion about the mean helps to increase the probability amplitude of the pattern similar to the sought pattern and decrease the probability of other non-similar patterns.

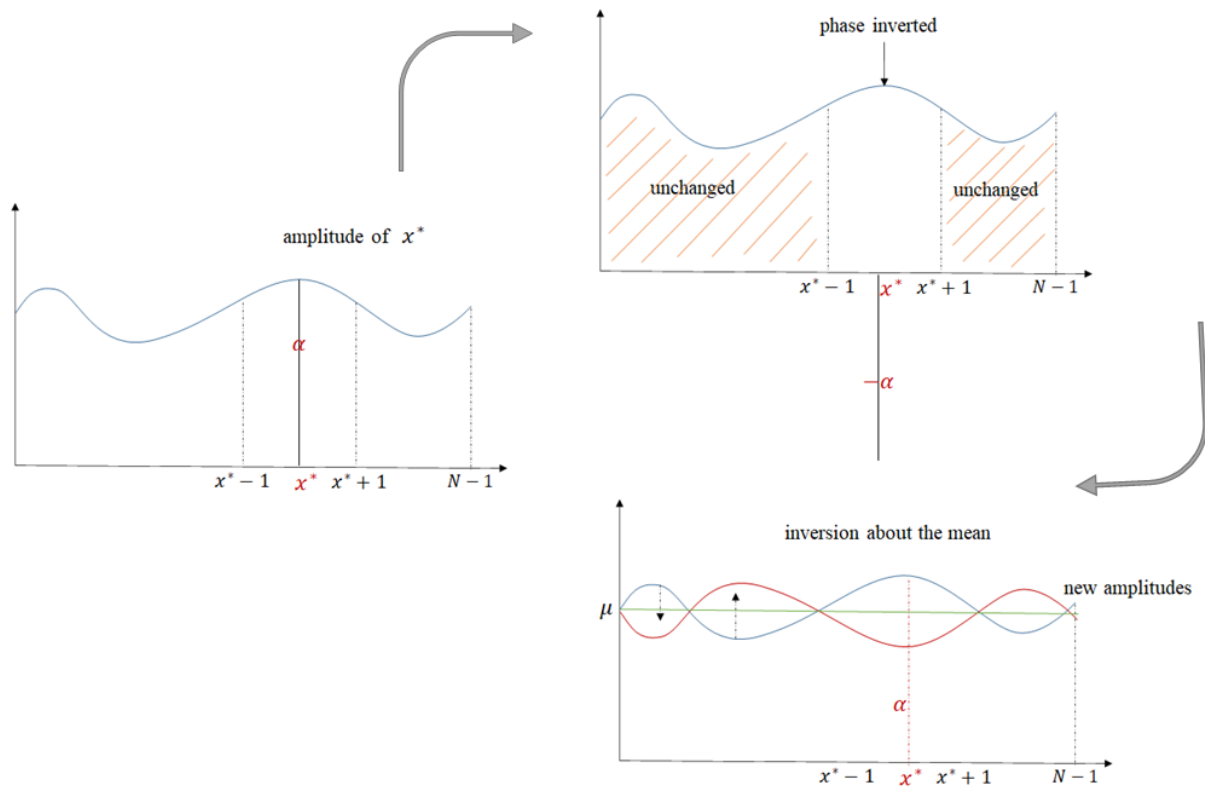


Figure 6-5: The mathematical representation for Grover's algorithm

To implement the Grover's algorithm the following operators are used to perform phase inversion and inversion about the mean.

$$\hat{I}_\phi = \text{identity matrix except for } \phi\phi = -1 \quad (6-5)$$

The \hat{I}_ϕ which inverts the phase of the state $|\phi\rangle$.

$$\hat{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (6-6)$$

The \hat{W} (Walsh or Hadamard transform) operator which performs a special case of the discrete Fourier transform.

Using Grover's algorithm to search for a specific pattern in a database of size $N = 2^n$ where n is the number of qubits, all the states are initialised to $|\bar{0}\rangle$ using \hat{W} operator which results in amplitudes $\frac{1}{\sqrt{N}}$. Then the \hat{I}_τ phase inversion operator is applied to change the amplitude of the sought state $|\tau\rangle$. Ultimately, the \hat{G} operator is applied to invert all the states' amplitude about the mean:

$$\hat{G} = -\hat{W}\hat{I}_\tau\hat{W} \quad (6-7)$$

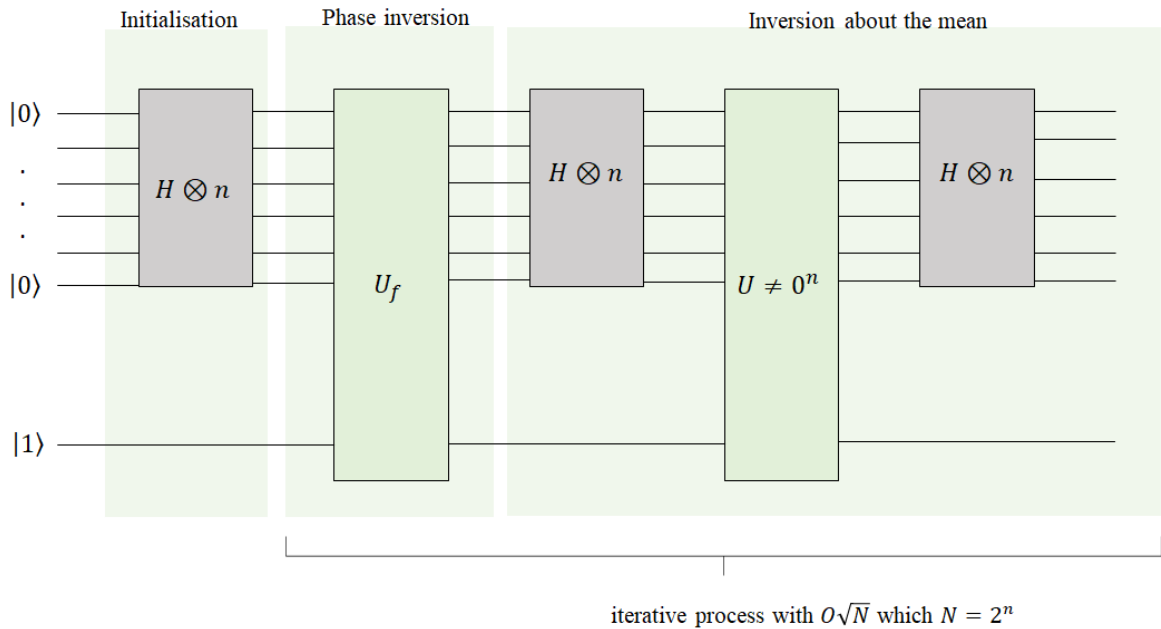


Figure 6-6: The Quantum circuit for Grover's Algorithm

Figure 6-6 shows the quantum circuit for Grover's search algorithm and Figure 6-7 demonstrates the Grover's algorithm pseudocode.

Grover's Algorithm	
1:	Generate the initial state $ \bar{0}\rangle$.
2:	Initialise all the states' amplitude using $\hat{W} \bar{0}\rangle = \bar{1}\rangle = \psi\rangle$
3:	For $i \leq \frac{\pi}{4}\sqrt{N}$
4:	Apply the phase inversion using $ \psi\rangle = \hat{I}_\tau \psi\rangle$
5:	Apply the inversion about the mean using $ \psi\rangle = \hat{G} \psi\rangle$
6:	Observe the system
18:	end for

Figure 6-7: Grover's algorithm pseudocode

6.4.2 The CQIEA Pattern Retrieving Algorithm

In the proposed CQIEA pattern retrieval method by moving a sliding window across the sought pattern and comparing it with the stored firing memories of neurons the highest similarity amongst the neural activities produced by the network is selected and retrieved as the output. The similarity threshold is set to 95%. The performance of the proposed algorithm highly depends on the number of neurons and synaptic connections amongst them and the sparsity of the training patterns.

The CQIEA will be used as a pattern retrieval on the stored quantum memory (the number of qubits depends on the sought patterns length). The state of the patterns will be changed using the Chirikov chaotic quantum gate introduced in chapter 4. Using chaotic quantum gate the

amplitude of the stored quantum patterns will be changed in a way that the similar patterns to the sought pattern have the highest amplitude. This process will be repeated for all the stored patterns until the most similar pattern to the stored patterns is retrieved.

Figure 6-8 illustrates the schematic representation of CQIEA pattern retrieval algorithm for the Quantum Associative Memory in SNNs.

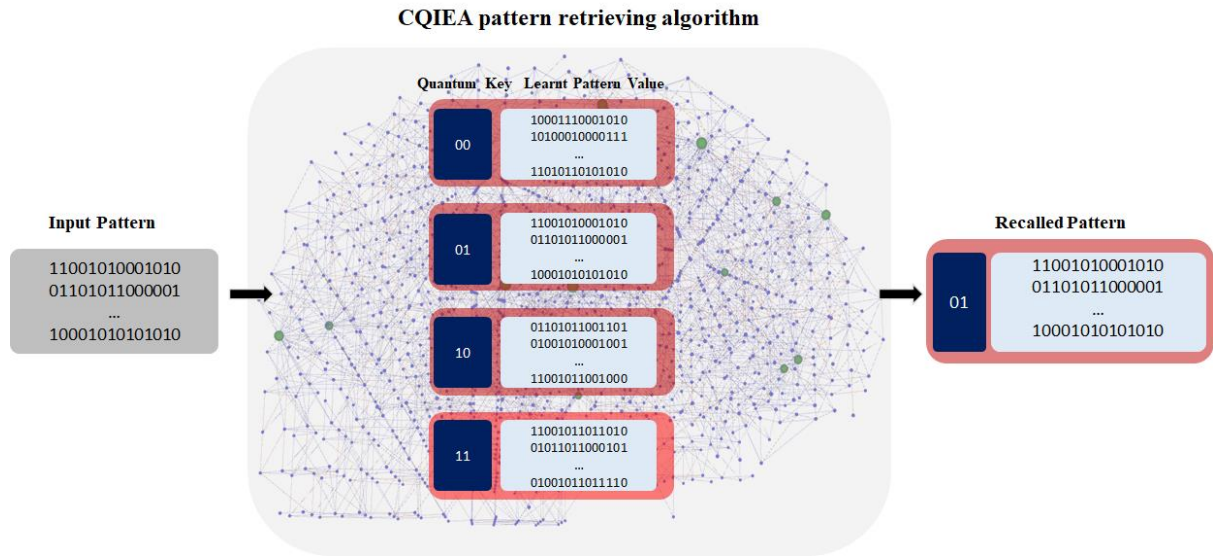


Figure 6-8: Schematic representation of CQIEA pattern retrieving algorithm

Both CQIEA pattern retrieval algorithm and Grover's algorithm use the Jaccard proximity measurement to compute similarity between the sought binary input and the stored patterns.

$$d_{st} = \frac{\#[(x_{sj} \neq y_{tj}) \cap (x_{sj} \neq 0) \cup (y_{tj} \neq 0)]}{\#[(x_{sj} \neq 0) \cup (y_{tj} \neq 0)]} \quad (6-8)$$

Where the d_{st} is the similarity value, the x_{sj} is the sought pattern, and y_{tj} is the stored pattern.

Theoretically, both Grover's algorithm and CQIEA algorithm can search unstructured databases in an acceptable time for the desired patterns and retrieving the sought pattern even in the presence of partial noisy input data. As a future direction and to reveal the power of

quantum associative memory, more experiments with large scale dataset needed to be conduct. As mentioned before, combining the SNN models with quantum computing principals can lead to powerful models with computational advantages and is a promising future for the new generation of quantum computers and neuromorphic hardware.

6.5 Conclusion

In this chapter, a novel quantum associative memory scheme for spiking neural networks is proposed to perform a fast, online pattern recognition for classification and prediction tasks just by partial pattern exposure. Two quantum pattern retrieval algorithms are proposed here to be embedded into EPUSSS or QISNN models: Grover's algorithm and CQIEA algorithm. In the novel proposed quantum associative memory, the learnt patterns which are produced during the local learning of either the EPUSSS model or QISNN by the input neurons (input twin neurons) will be stored as qubits in the quantum memory. In this way, the associative memory is integrated into both models. After local learning, when the input spike trains dynamic is captured and learnt by the network, the patterns can be recalled in a more time-efficient manner. After storing patterns in the local learning phase of the EPUSSS model or QISNN model, Grover and the CQIEA search algorithms are used to retrieve the sought pattern. Both algorithms use Jaccard proximity measurement to compute the similarity between two binary inputs. The proposed methods set a new direction for building AM in the future as STAM.

Publications that present the two methods are under construction.

Chapter 7 . Pre-processing and Feature Extraction from Brain Spatio-temporal EEG Data

Abstract

Almost all the real-world datasets suffer from being incomplete, inconsistent, inaccurate due to data acquisition errors or outliers, and often contain missing values. The quality of the dataset has a great impact on machine learning performance in learning and extracting meaningful insights from data. Therefore, data pre-processing and feature extraction play an important role in enhancing machine learning's efficiency. In this chapter, data pre-processing and feature extraction methods have been applied to increase the proposed SNN models' efficiency.

This chapter addresses the following research question.

RQ 8 “How to pre-process Spatio-temporal data and to extract useful features that can help improve the applicability of neuromorphic computation for a real-world application”.

The application problem here relates to the analysis of EEG brain data collected from dementia patients.

7.1 Introduction

In the last century, dementia diseases become the main concern all over the world due to the increase individuals' life expectations. Dementia occurs when the brain has been affected by a specific disease or condition that causes cognitive impairment (Hampel et al., 2010). Alzheimer's disease (AD), Vascular Dementia (VaD) and Mixed Dementia (MD), which is a combination of Alzheimer's disease and Vascular Dementia, are the most common forms of

dementia (Cedazo-Minguez and Winblad, 2010; DeKosky and Marek 2003; Gorelick, 1997). The AD, VaD, and MD are very similar in their symptom's characteristics. Since the treatment of different types of dementia depends on the correct diagnosis, clinically distinguishing them is very important and challenging for the physician.

Alzheimer Disease (AD) is the most common form of dementia, accounting for two-thirds of all dementia diagnoses. AD causes a gradual and progressive decline in cognitive capabilities including episodic memory loss and executive functions disablement that can lead to social or occupational impairment. Since the early detection of AD may increase the chance of success in therapies and help AD patients to be managed more effectively, an accurate clinical diagnosis in the early stages of the disease becomes very important to neuropsychologists (Galimberti and Scarpini, 2010; Saykin et al, 2010; Linnenbringer et al, 2010; Goldman et al, 2011).

However, since the clinical diagnosis of AD relies on the comparison of a patient's cognitive performance with that of normal non-demented individuals who are matched for age, gender, and education with the subject, may cause failure in capturing some subjects in the early stages of the disease. Moreover, the earliest symptoms of mild impairment can be mistaken with healthy ageing, stress or can be confused with symptoms of other brain disorders. Therefore, diagnosis of AD is a very difficult task that requires the rich clinical experience of physicians.

Due to the non-invasive nature of electroencephalographic (EEG) recording, it is a common technique to collect data from brain disorders patients to understand their mental activities. EEG recordings provide a direct measure of cortical activity with millisecond temporal resolution. Hence, there are various types of data available for analysing brain disorders, specifically AD, including EEG recordings, fMRI images, clinical and demographic information. Most of these collected data are spatio and/or Spatio-temporal data (SSTD) that

include measuring spatial or/and spectral variables over time that making them a suitable application for spiking neural networks.

Despite spiking neural network biological plausibility, fast information processing, and the ability to preserve the Spatio-temporal relationship of the data stream, the pattern recognition and prediction of its occurrence is a difficult task for SNNs. Finding spiking patterns that can represent the learnt information is a challenging task and so far, few attempts have been done to develop a predictive model based on spiking neural networks. To have a good prediction performance, the network behaviour (pattern) for a specific sample should be produced and stored to recall it in the future. It will be more efficient if the desired pattern can be recognized in the presence of partial information, as the human associative memory. In this thesis, to develop a predictive model that can both perform off-line and on-line prediction and recall the whole desired pattern when just some part of the information is exposed to the network, I proposed novel strategies to capture, store and retrieve the patterns.

In this chapter, as a case study, the proposed models are applied to predict the development of brain disorder risk. To this end, a dataset related to Dementia and Alzheimer Disease are used. The dataset is collected by Russian neuroscientists from demented patients. This dataset consists of 67 patients' static clinical data along with their EEG recordings. These patients belong to three different dementia classes: Alzheimer Disease (8 patients), Vascular Dementia (14 patients) and, Mixed Dementia (45 patients). The dataset suffers from several issues:

- EEG recordings suffer from:
 - Different recording times
 - Different EEG channels
 - Highly imbalanced
- Static clinical data suffers from:

- Missing values
- Highly imbalanced

To overcome these issues, data pre-processing methods including data cleaning, data oversampling, and feature extraction techniques have been used which are explained in the following sections.

7.2 The Dementia EEG Recordings Pre-processing

As was stated above, the Dementia dataset needed several pre-processing steps to be ready to be fed into the proposed models in this thesis. This original dataset contained the raw EEG data in “EDF” format. The EDF is the abbreviation of European Data Format which is a simple and flexible format for the exchange and storage of multichannel biological and physical signals. EDF became a standard format for EEG and PSG recordings since 1992 in biological signal recording equipment and research projects (Alvarez-Estevéz, 2021).

The first step before deploying the proposed models on the Dementia dataset is to read the EDF format files into “csv” format which is an acceptable format by all the proposed models implemented in this thesis using a MATLAB script. Afterwards, the above-mentioned pre-processing techniques are performed to increase the performance of the proposed models. Figure 7-1 provides a visual representation of the different pre-processing techniques used on the Dementia dataset.

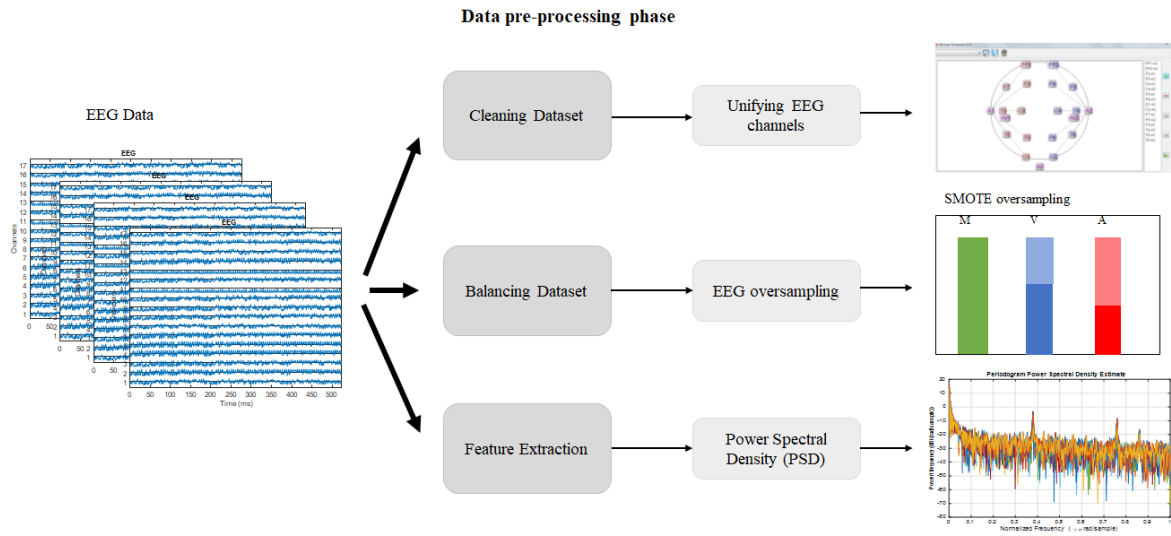


Figure 7-1: Data pre-processing phase diagram for the Dementia dataset

7.2.1 Unifying EEG Channels

During data collection for the Russian Dementia EEG datasets, the EEG “eeg-nevropoligraf” device with two different EEG recording schemes have been used which resulted in different EEG channels for the recorded data. One scheme recorded the data with 20 channels (Figure 7-2) and the other with 17 channels (Figure 7-3). The three extra channels in the data recorded by the first scheme are removed after consulting with an expert to have a uniform format for all EEG recordings.

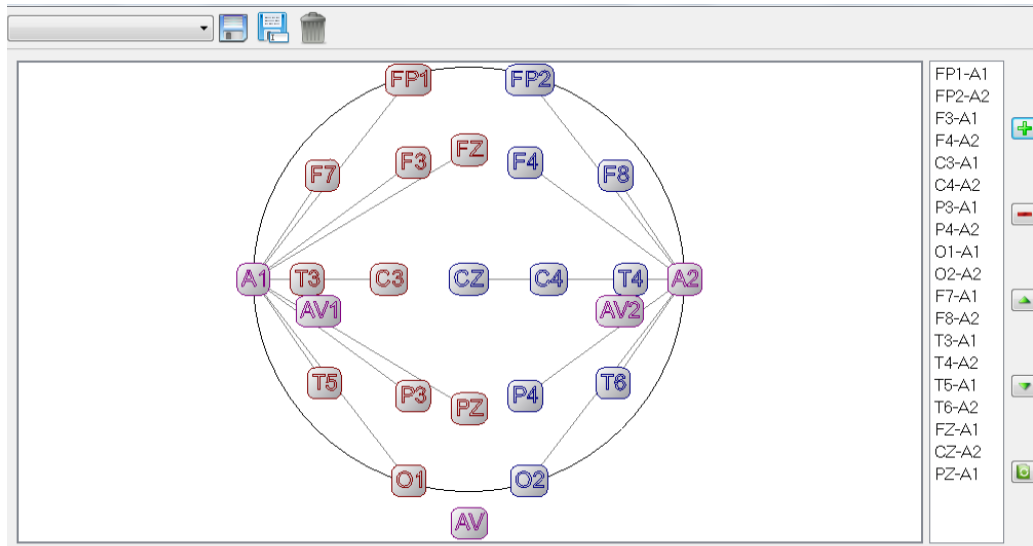


Figure 7-2: EEG recording scheme with 20 electrodes

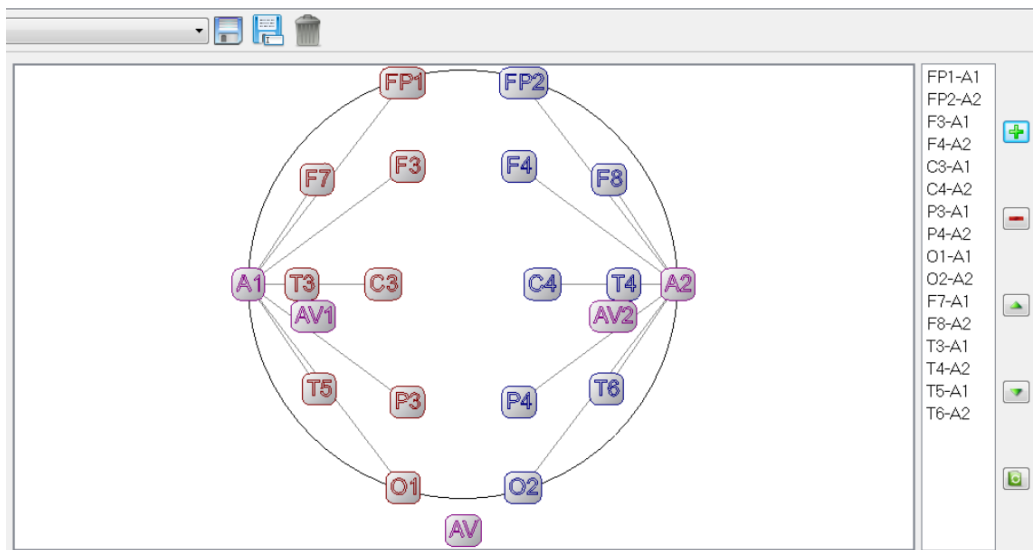


Figure 7-3: EEG recording scheme with 17 electrodes

Table 7-1, shows the EEG channels for the two recording schemes for the Dementia dataset. The highlighted channels are those three extra channels that were removed from the recording scheme with 20 electrodes to create a uniform format for all EEG recordings in the dataset.

Table 7-1: The EEG channels for the two recording schemes.

EEG channels for scheme 1	EEG channels for scheme 2
PZA1	A2A1
A2A1	O1A1
O1A1	C3A1
FZA1	P3A1
C3A1	FP1A1
P3A1	F3A1
FP1A1	T3A1
F3A1	T5A1
T3A1	F7A1
T5A1	F4A2
F7A1	Fp2A2
F4A2	P4A2
Fp2A2	C4A2
P4A2	O2A2
C4A2	F8A2
O2A2	T6A2
CZA2	T4A2
F8A2	
T6A2	
T4A2	

7.2.2 EEG Feature Extraction using Power Spectral Density

EEG recordings are notable for their high temporal precision and are often recorded with sampling rates around 1000 Hz or higher. This is good when precise timing of events is important to the experimental design or analysis plan, but also consumes more memory and computational resources when processing the data. In cases where high-frequency components of the signal are not of interest and precise timing is not needed downsampling, the signal can be a useful time-saver. The Dementia EEG dataset was recorded with different recording times varying between 35500 milliseconds up to 204500 milliseconds. Learning and analysing these long varying recordings are a very memory intensive function and computationally expensive task for almost all the machine learning models. Besides, the different recording length of each sample makes the implementations of the proposed models challenging. To overcome these issues Power Spectral Density is used as the feature extraction method to extract the most representative features of the signals to have more informative signals with the same length without losing important information. The Power spectral density (PSD) filter which describes the distribution of average signal power in the frequency domain has been widely used for stationary signal processing and is suitable for narrowband signals. PSD is a common signal processing technique for EEG recordings that distributes the signal power over frequency and show the strength of the energy as a function of frequency (Al-Fahoum and Al-Fraihat, 2014; Bascil et al., 201; Hong et al., 2018). The PSD is calculated by applying Fast Fourier Transform (FFT) to the estimated autocorrelation sequence using nonparametric methods like the Welch's method. In Welch's method, a moving window slides over the data segments, then FFT is applied to each segment to compute the periodogram, finally the average of all periodograms results in a modified periodogram or the Welch PSD of signals.

Let the input signal be $x[n]$ where $n = 0, 1, 2, \dots, N - 1$, then the signal is split into N_s overlapping segments (e.g., 50% overlap) with length L . The i th segment of the signal can be represented as follow:

$$x_i = x[i + \frac{L}{2} + n] \quad (7-1)$$

Where $i = 0, 1, \dots, L - 1$ and $n = 0, 1, \dots, N - 1$. The sampling length N can be calculated as below:

$$N = L + (L - N_D)(N_s - 1) \quad (7-2)$$

where N_D is the number of overlapping points, N_s is the number of segments, and L is the length of the segment. After dividing the signal into the N_s segments, the Hamming window $w(n)$ is applied to each segment:

$$w(n) = 0.54 - 0.46 \cos \left[\frac{2n\pi}{L} \right] \quad (7-3)$$

The Hamming window function prevents the spectral leakage (Xiong et al., 2020). The window function can apply to the i th segment of the signal as follow:

$$W_i = x_i(n) \times w(n) \quad (7-4)$$

Then, the Fourier transform of each windowed segment is computed as below:

$$A_i(k) = x_i(n) \times w(n) e^{-i\left(\frac{2\pi}{N}\right)nk} \quad (7-5)$$

Where $A_i(k)$ is the Fourier transform of the i th windowed segment. Thereupon, the periodogram of each Fourier transformed windowed segment is computed as follow:

$$\phi_i = \frac{1}{LU} |A_i(k)|^2 \quad (7-6)$$

Where $U = \frac{1}{L} \sum_{n=0}^{L-1} w^2(n)$ indicates the mean power of the window $w(n)$ and accordingly the LU is the energy of the window function $w(n)$ with length L .

Eventually, the PSD Welch of the signal can be obtained by averaging all the computed periodograms:

$$S(k) = \frac{1}{L} \sum_{i=0}^{L-1} \phi_i(k) \quad (7-7)$$

Power Spectral Density (PSD) normalizes the amplitudes by the frequency resolution to give the amplitudes a similar appearance. Therefore, it can preserve the shape of the signal after extracting the most informative features.

Power Spectral Density (Welch's method)

- 1: Divide data into overlapping segments.
 - 2: Add specified windows to each segment.
 - 3: Apply FFT to windowed segments.
 - 4: Compute periodogram of each windowed segment.
 - 5: Compute average of all periodograms to obtain Welch PSD
 - 6: End
-

Figure 7-4: Power Spectral Density Welch algorithm pseudocode

Applying this method for feature extraction from the described dementia data set is illustrated below (Figure 7-5 to Figure 7-8).

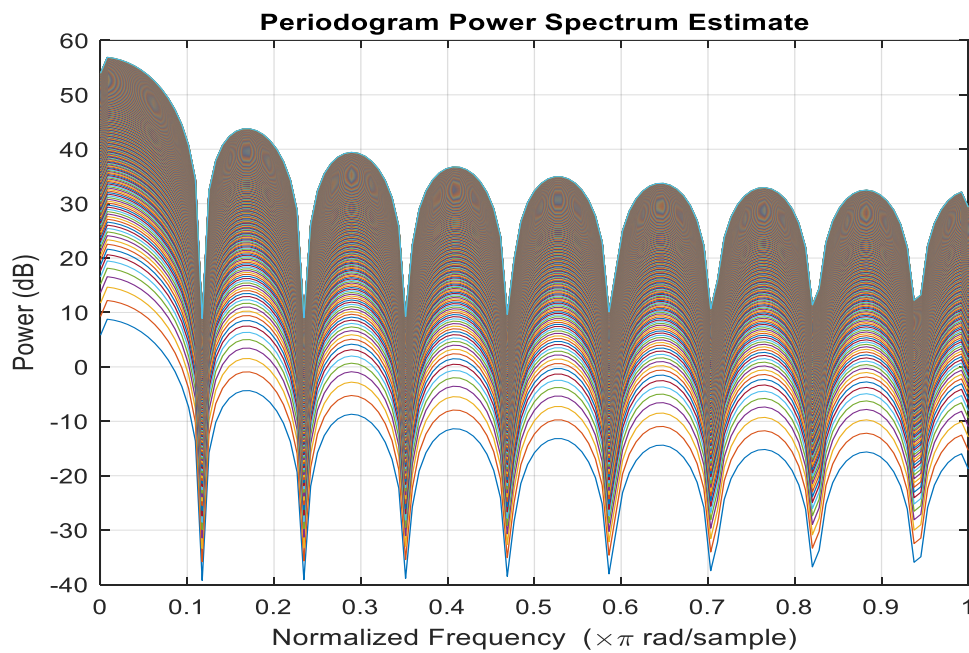


Figure 7-5: Power spectrum estimator

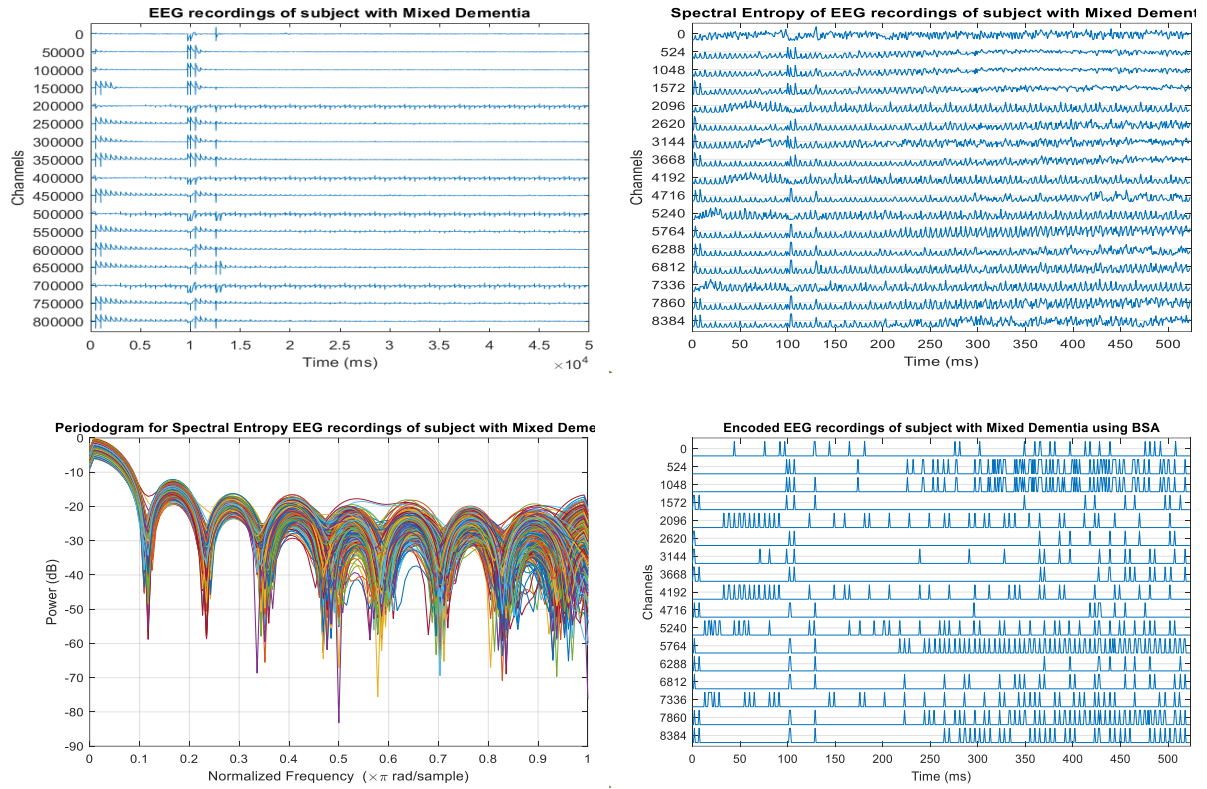


Figure 7-6: Preprocessing of Mixed Dementia EEG recordings

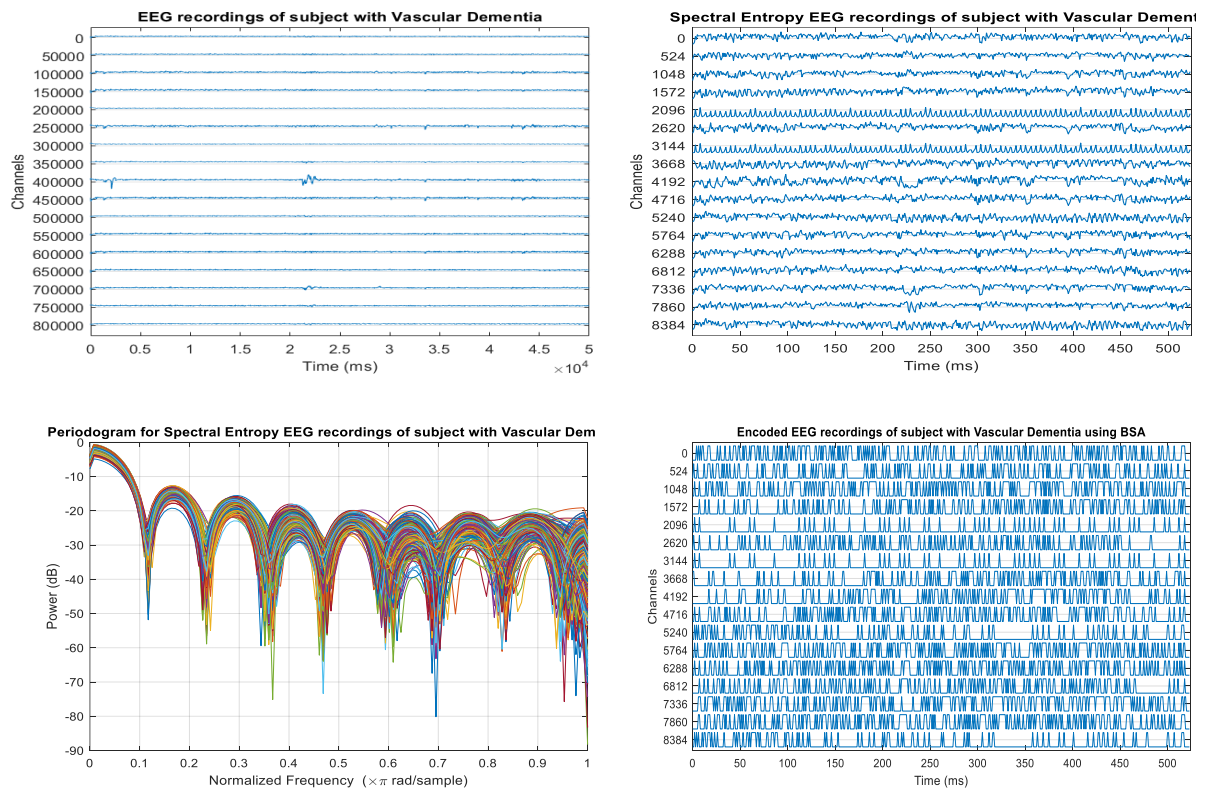


Figure 7-7: Preprocessing of Vascular Dementia EEG recordings

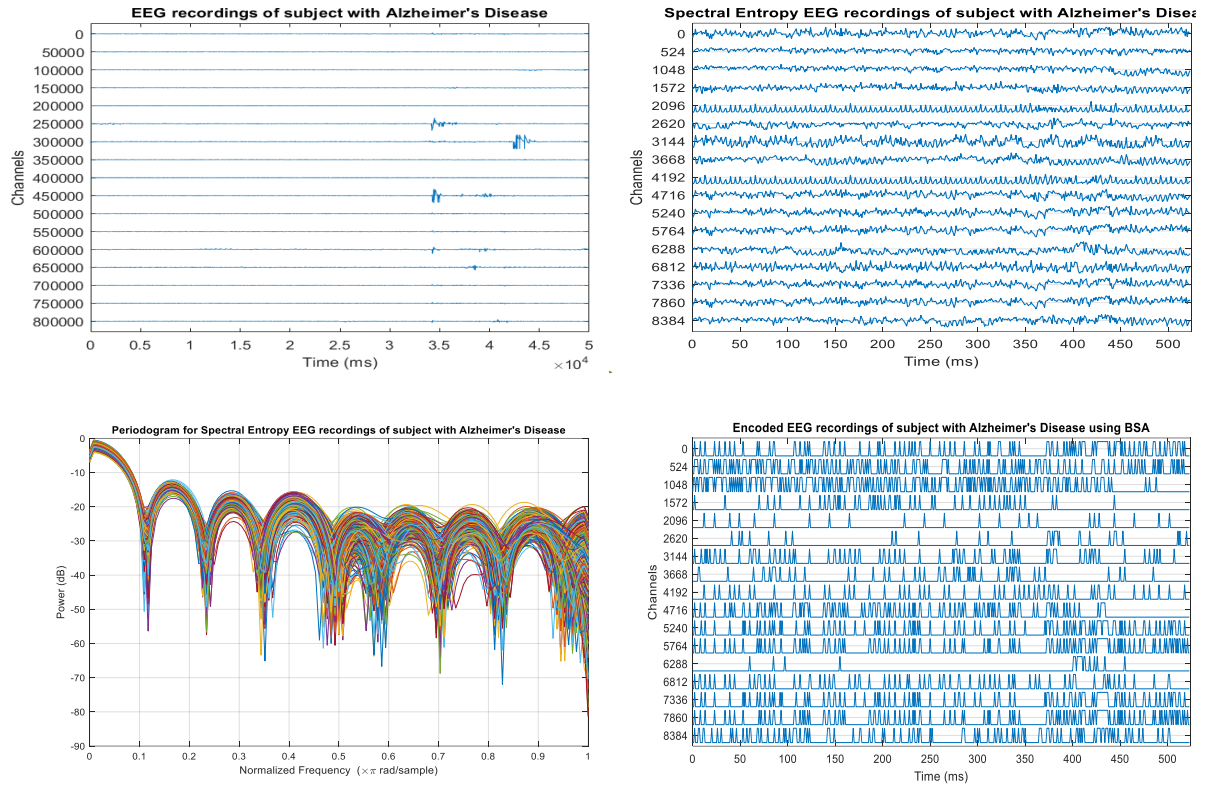


Figure 7-8: Preprocessing of Alzheimer's Disease EEG recordings

After applying the PSD Welch method on the Dementia EEG recordings, the time points are reduced to 524. In the proposed PSD Welch method, the frequency is set to 100 and the size of the window is set to 250.

7.2.3 Balance Dementia EEG Dataset

As it was mentioned before the Dementia dataset was highly imbalanced which could affect the reliability of the outcome of learning using the proposed models in this thesis. To balance the EEG recordings dataset, the Synthetic Minority Oversampling Technique (SMOTE) is used to generate artificial data based on the similarities between the feature spaces of the existing minority class samples. The algorithm generates some random points in between the two specified vector points and thus more generalises the minority class decision region (Han, Wang, and Mao, 2005). After applying SMOTE oversampling technique, the Dementia dataset

consists of 135 data including 45 Mixed Dementia, 45 Vascular Dementia, and 45 Alzheimer's Disease subjects.

7.3. Experimental Result of Applying EPUSSS Model to the Dementia Dataset

In this section, the comparative analysis of applying EPUSSS, NeuCube and LSTM on the Dementia dataset is provided. Figure 7-9 to Figure 7-18 represent the EPUSSS local learning, global learning, and the Energy function behaviour in the process of training with the Dementia dataset.

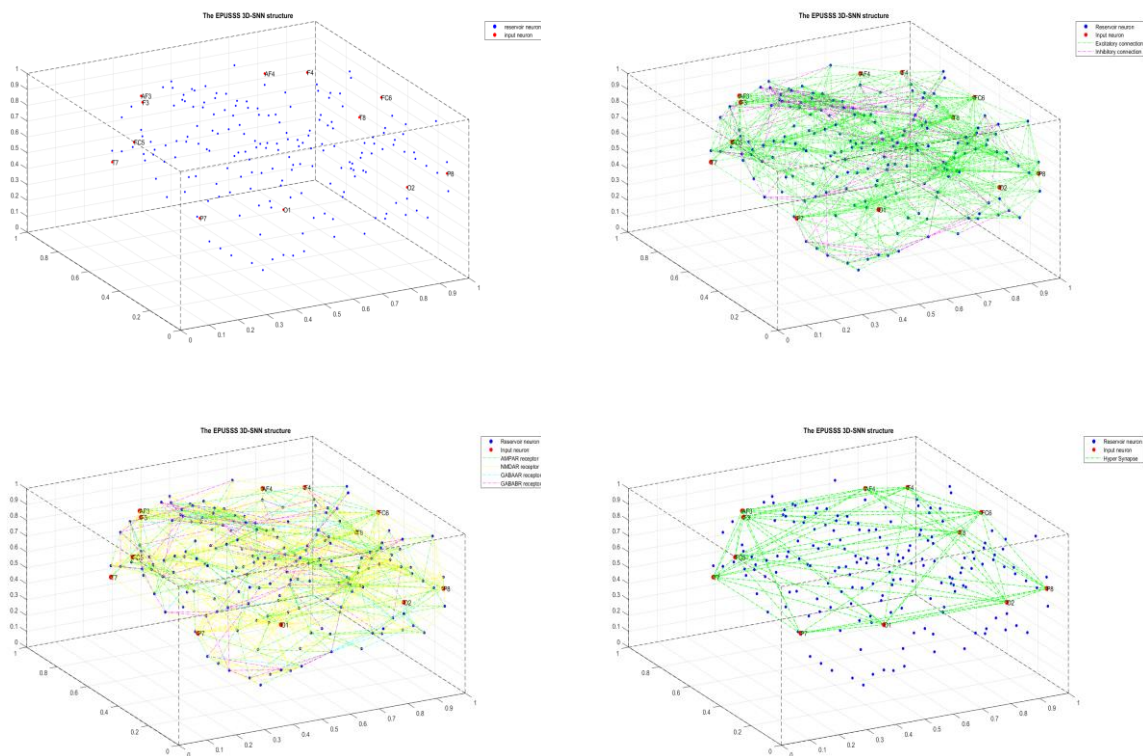


Figure 7-9: EPUSSS SNN initialisation for Dementia

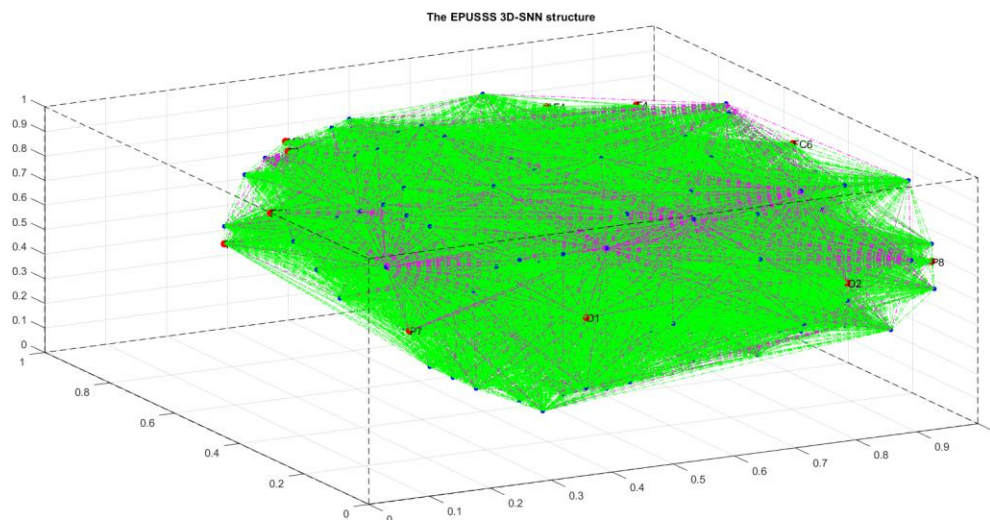
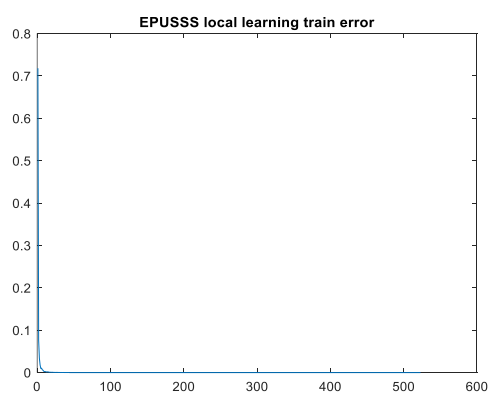


Figure 7-10: EPUSSS SNN learnt weights and connectivity for Dementia

a)



b)

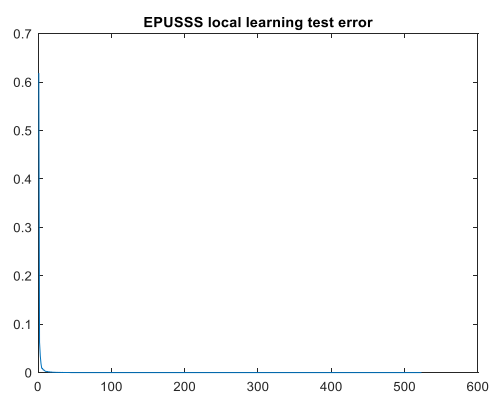


Figure 7-11: EPUSSS local learning errors convergence, a) training error, b) test error

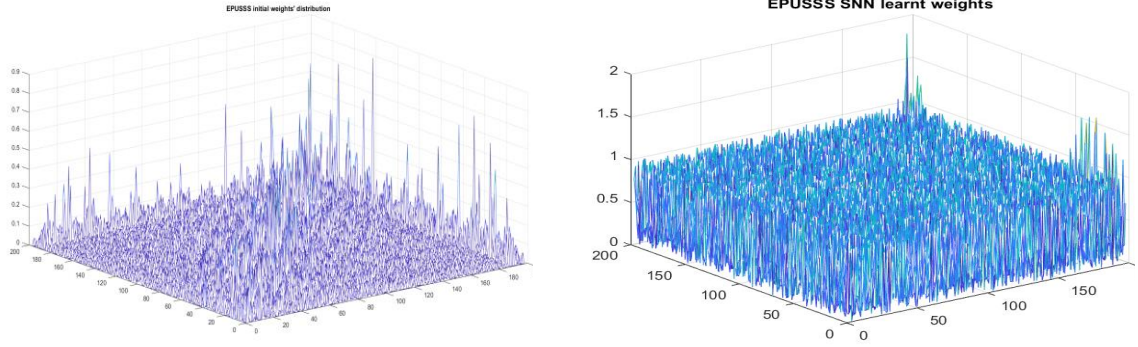
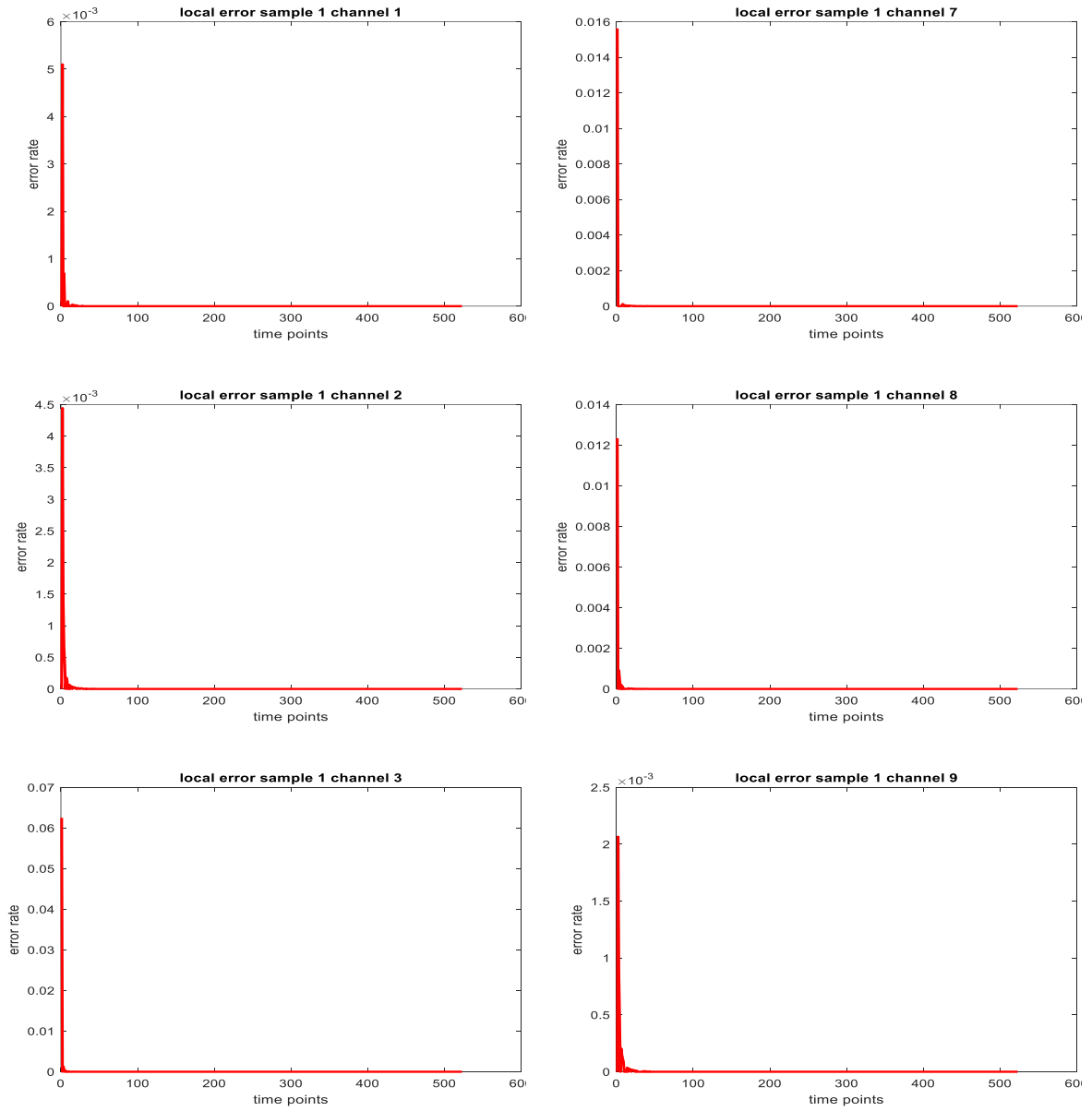


Figure 7-12: EPUSSS SNN weights distributions: a) initial weights, b) learnt weights



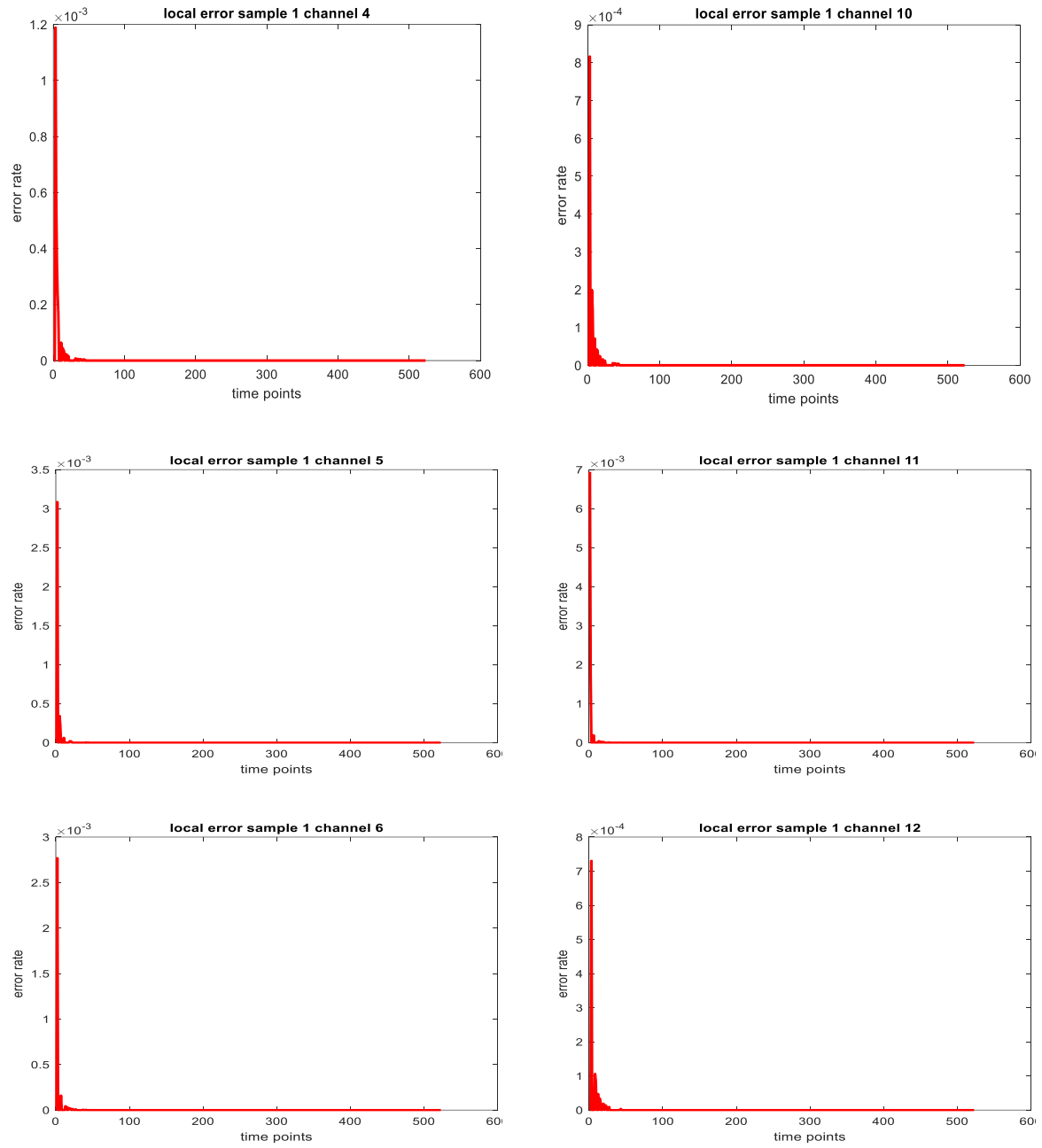


Figure 7-13: EPUSSS local error for each input in training phase for sample 1

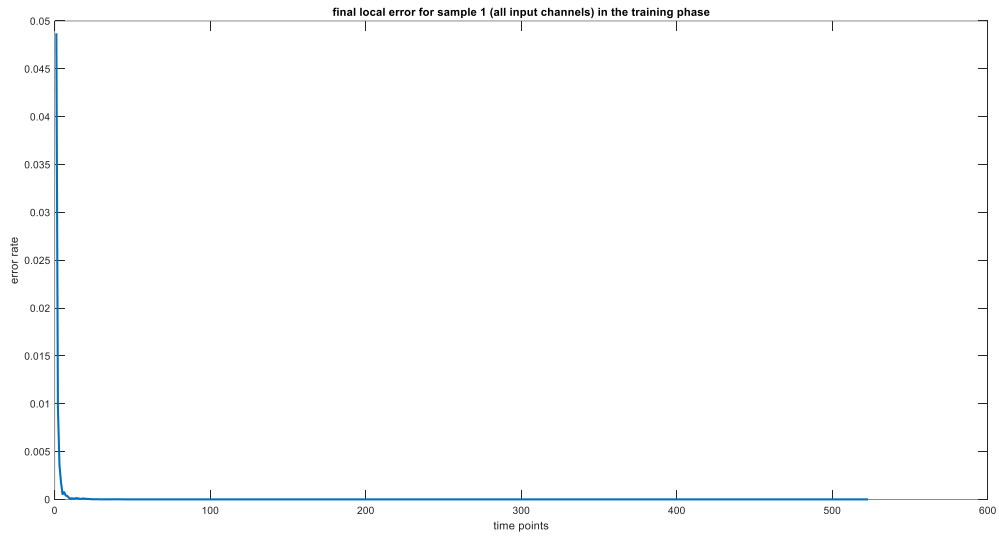
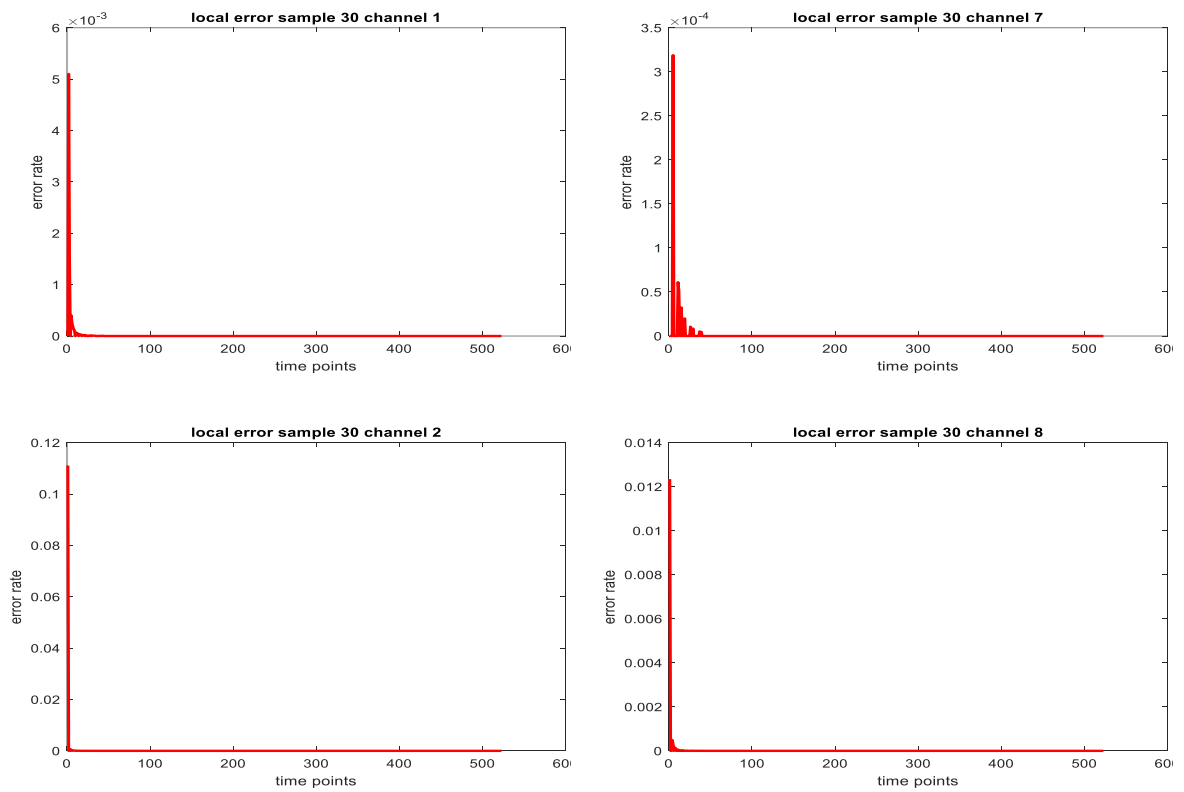


Figure 7-14: EPUSSS final local error for sample 1 (all input channels) in the training phase



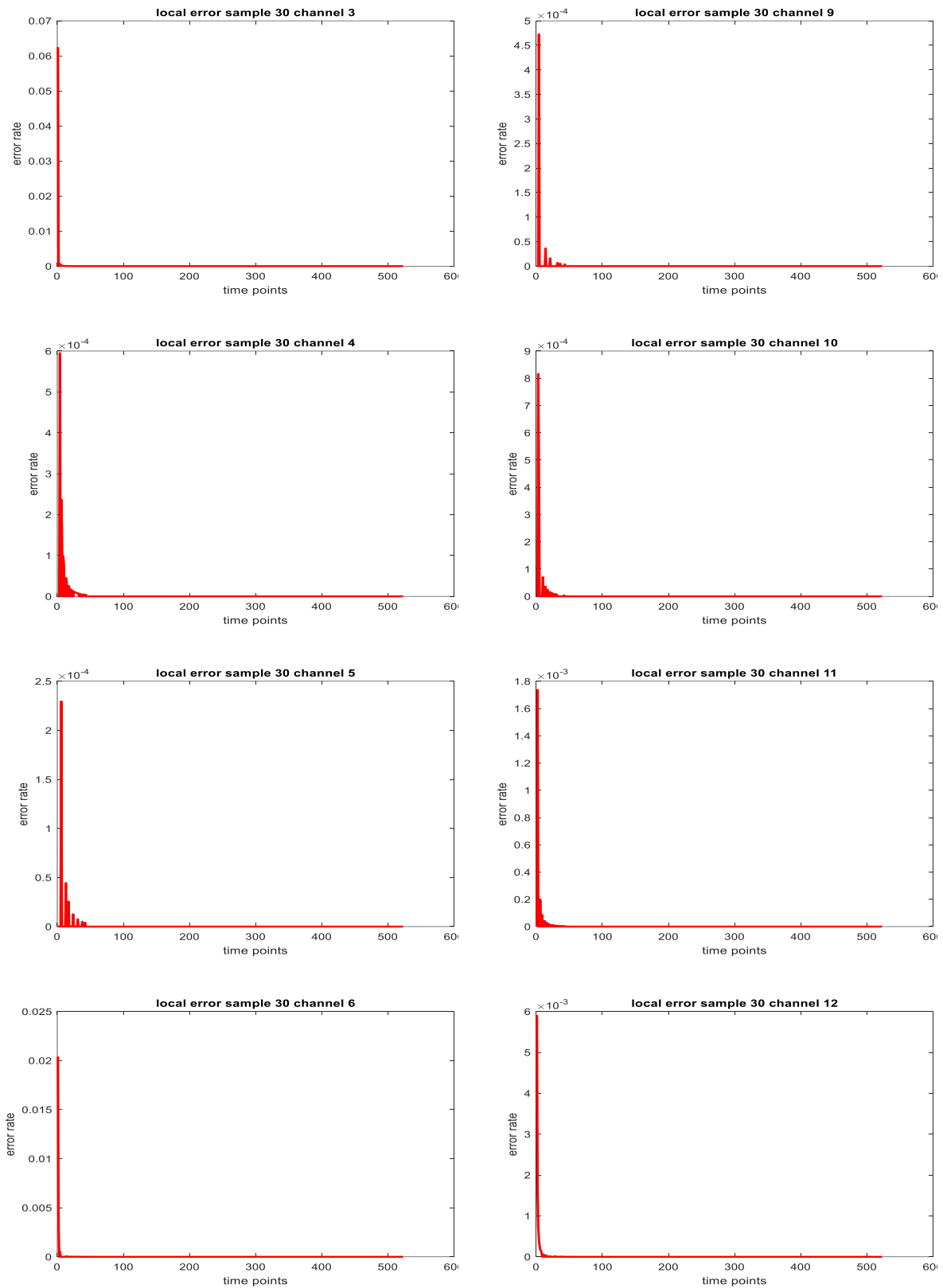


Figure 7-15: EPUSSS local error for each input in testing phase for sample 30

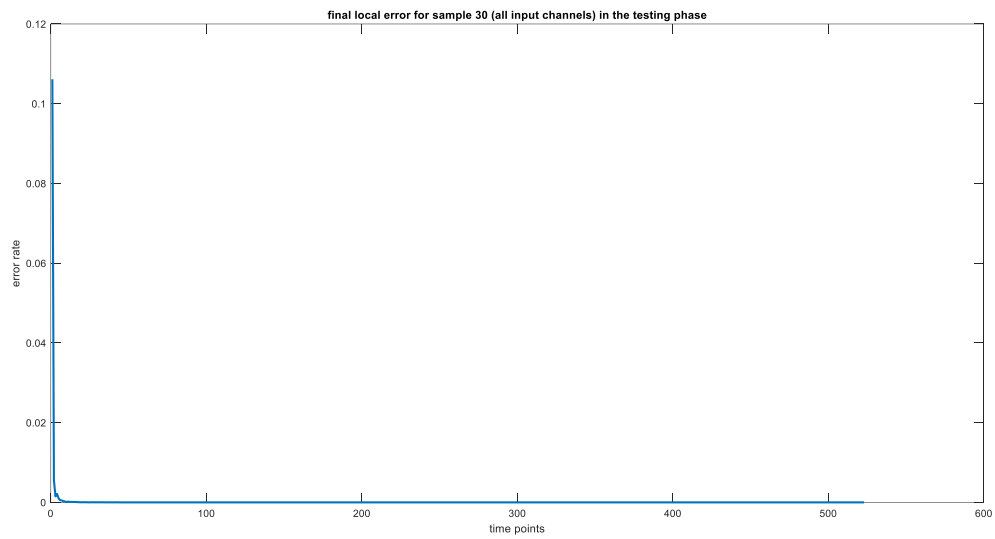


Figure 7-16: EPUSSS final local error for sample 30 (all input channels) in the testing phase

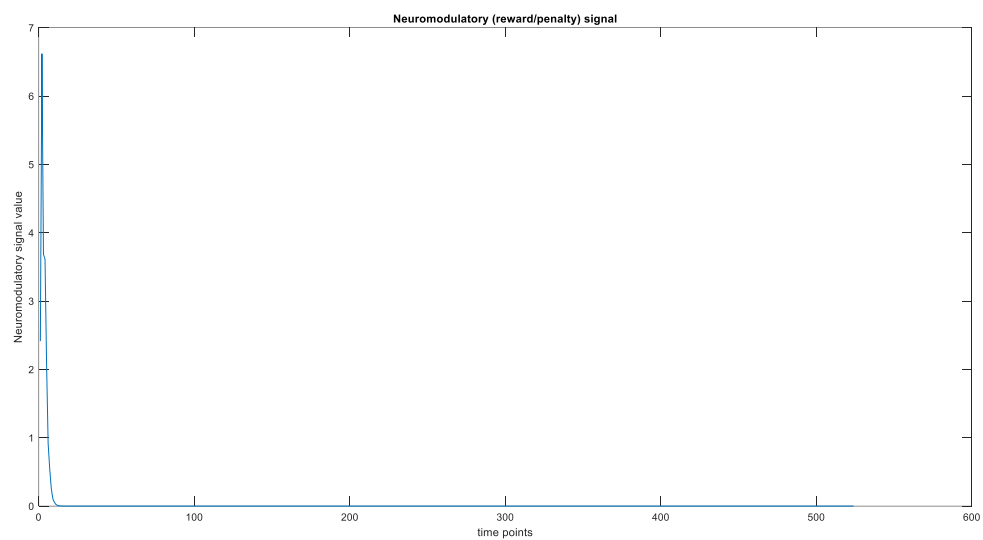


Figure 7-17: Neuromodulatory (reward/penalty) signal in the EPUSSS global training

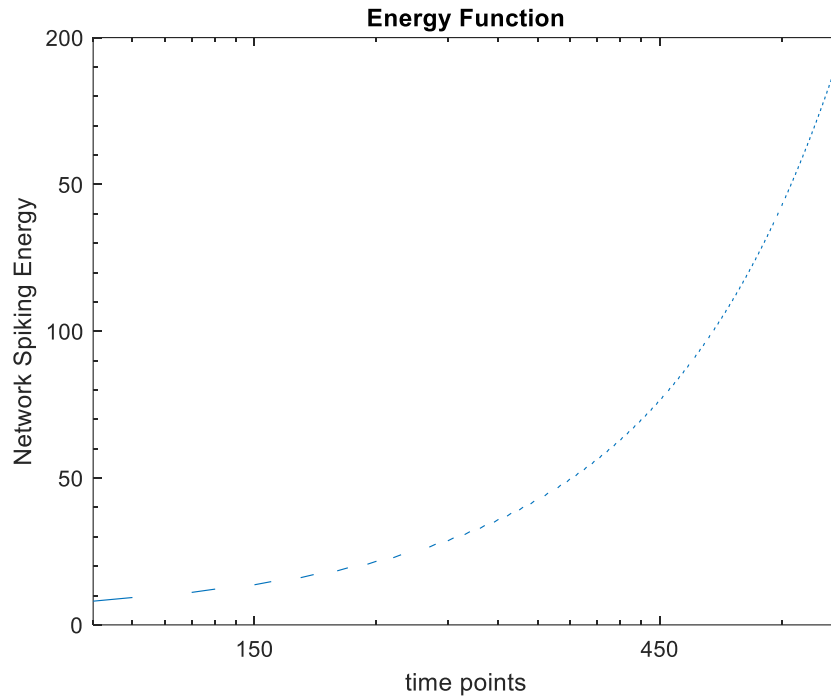


Figure 7-18: Spiking energy of the EPUSSS SNN network during training for Dementia Dataset

The increase in the Energy function proves that the network learnt the input patterns by demonstrating more excitations in neuronal activities.

Like chapter 3, NeuCube and LSTM models have been used for the comparative analysis. The same parameter setting as it was mentioned in chapter 3 was applied to EPUSSS and NeuCube. Following the same parameter setting of LSTM model in chapter 3, the LSTM deep learning recurrent model was designed with 5 layers including a sequence input with 14 dimensions, a BiLSTM layer with 100 hidden units, and three fully connected layers. The SoftMax model was used as the activation function and the cross-entropy loss was used for measuring the classification error.

The comparative results of applying NeuCube and LSTM on the Dementia dataset can also be found in Figure 7-2 and Figure 7-3.

Table 7-2: The comparative analysis of Global Learning Classification Results for Dementia Dataset

	Classification	Accuracy	Precision	Recall	F-score
EPUSSS	Class 1	82%	0.84	0.73	0.78
	Class 2	81%	0.66	0.75	0.70
	Class 3	83%	0.73	0.76	0.75
NeuCube	Class 1	80%	0.80	0.66	0.72
	Class 2	79%	0.68	0.70	0.69
	Class 3	81%	0.62	0.77	0.69
LSTM	Class 1	80%	0.80	0.66	0.72
	Class 2	77%	0.64	0.67	0.65
	Class 3	82%	0.66	0.78	0.72

Table 7-3: The comparative analysis of Global Learning Classification Results Confusion Matrix for Dementia Dataset

	Confusion Matrix	Class 1	Class 2	Class 3
EPUSSS	Class 1	38	3	4
	Class 2	9	30	6
	Class 3	5	7	33
NeuCube	Class 1	36	5	4
	Class 2	10	31	4
	Class 3	8	9	28
LSTM	Class 1	36	7	2
	Class 2	10	29	6
	Class 3	8	7	30

NeuCube and LSTM learning process for Dementia classification can be seen in the following figures.

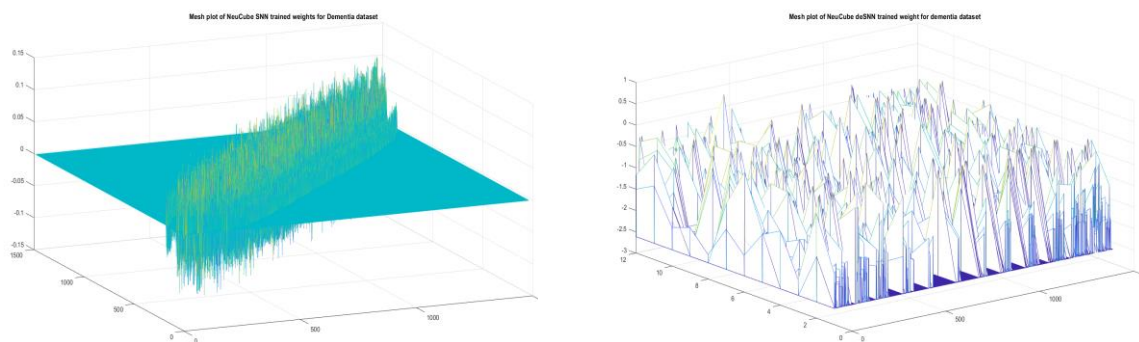
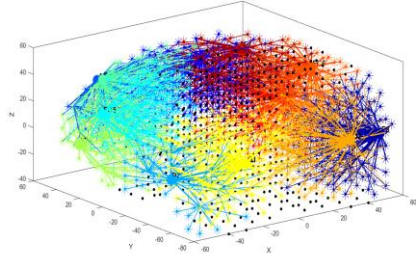
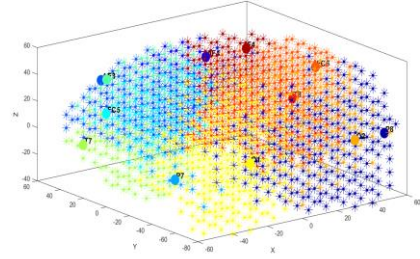


Figure 7-19: NeuCube SNN learnt weight distribution for Dementia

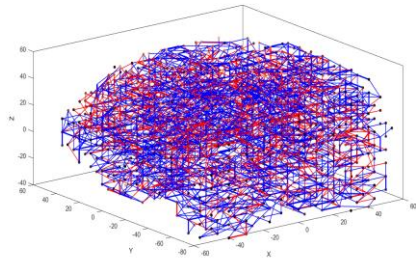
a)



b)



c)



d)

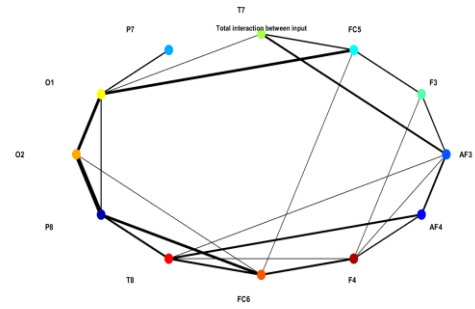


Figure 7-20: NeuCube SNN spiking behaviour for Dementia: a) spiking behaviour of input channels connected to reservoir, b) reservoir neurons regions according to their connectivity to EEG input channels c) excitatory and inhibitory connections, d) EEG channels correlations.

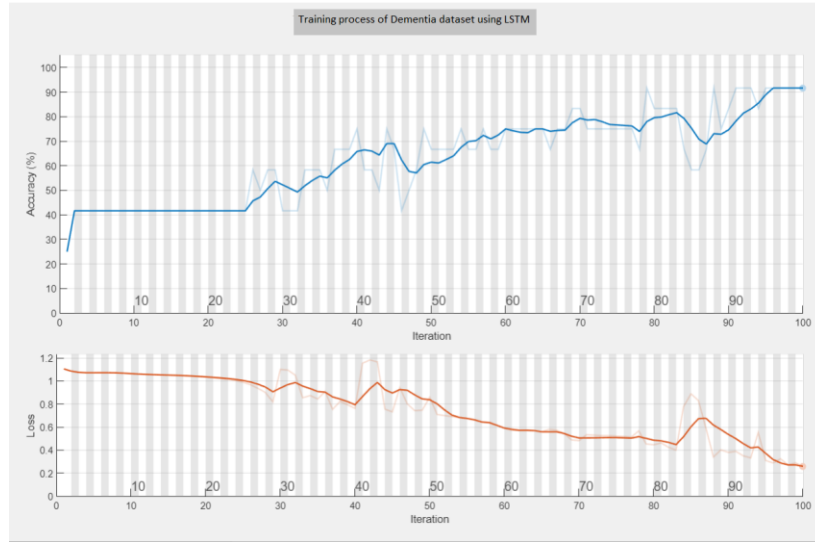


Figure 7-21: LSTM training accuracy and error plot for Dementia

The results of classification of the Dementia using EPUSSS outperform NeuCube and LSTM. However, both NeuCube and LSTM demonstrated acceptable performance in the training and testing phase using Dementia EEG data.

In the following section, a novel feature extraction method is introduced based on a combination of the proposed Chaotic Quantum-Inspired Evolutionary Algorithm and fuzzy rough set theory for clinical static data related to the EEG recordings. The purpose of this approach is to provide a potential for weighted learning of Spatio-temporal data based on the clinical and demographical observations of the subjects in the proposed methods.

7.4. Using CQIEA to Improve Traditional Fuzzy Rough Set Feature Selection Method to Better Tackle the Curse of Dimensionality Problem

7.4.1. Fuzzy Rough Sets

The rough set theory is a method to extract hidden patterns in a dataset with discretised attribute values without requiring any additional information about the data. However, most of the real-world datasets may have continuous attributes or contain both symbolic and real-valued (numerical and categorical) features where the rough set theory is not efficient enough to handle the situation. In other words, for these kinds of datasets, rough set theory cannot help to determine whether two attributes' values are similar and to what extent they are the same. Some approaches suggest discretising the continuous datasets beforehand and considering different sets of numerical and categorical attributes for the mixed datasets. However, these approaches fail to determine the membership degrees of the discretised values or degree of similarities of numerical and categorical features resulting in information loss and poor classification performance. Applying fuzzy concepts can empower rough set theory to deal with these kinds of complexity in the datasets.

7.4.2. Fuzzy Rough Membership Function

In the classical set theory, an element either belongs to a set or not, while fuzzy set theory determines a membership degree for the element so that it can be partially included in a set. The rough set theory uses approximations to determine whether the element belongs to a set or not. In other words, the fuzzy set uses infinite-valued logic to describe the belongingness of an element of the universe of discourse while the classical (crisp) set employs bi-valued logic or “yes / no” logic (Bandemer and Gottwald,1995).

The fuzzy rough membership function can be defined as the degree of relative overlap between the set X and the equivalence $[x]_B$ class to which x belongs (Skowron and Rauszer 1992; Pawlak and Skowron, 1994):

$$\mu_X^B: U \rightarrow [0,1] \text{ and } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

Where

(7-1)

$$\mu_X^B(x) \in [0,1]$$

The value of the membership function $\mu_X^B(x)$ is a conditional probability that represents the degree of certainty to which x belongs to X .

The rough set approximations and the boundary region of a set can be redefined using fuzzy rough membership as follow:

$$\underline{B}(X) = \{x \in U: \mu_X^B(x) = 1\} \quad (7-2)$$

$$\overline{B}(X) = \{x \in U: \mu_X^B(x) > 0\} \quad (7-3)$$

$$BN_B(X) = \{x \in U: 0 < \mu_X^B(x) < 1\} \quad (7-4)$$

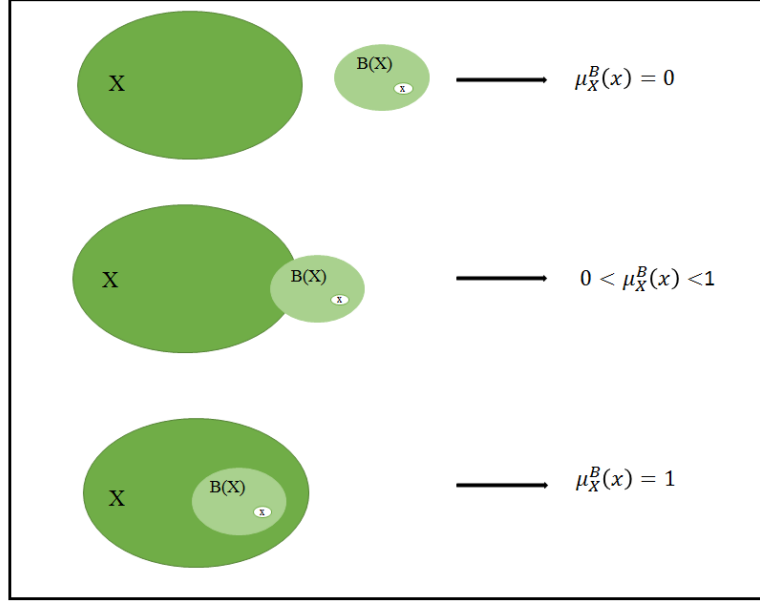


Figure 7-22: Fuzzy rough set membership

As it can be seen in Figure 7-22:

- $\mu_X^B(x) = 1$ *iff* $x \in \underline{B}(X)$
- $\mu_X^B(x) = 0$ *iff* $x \in \overline{-B}(X)$
- $0 < \mu_X^B(x) < 1$ *iff* $x \in BN_B(X)$

Some of the fuzzy rough membership properties are:

- $\mu_{U-X}^B(x) = 1 - \mu_X^B(x)$ *for any* $x \in U$
- $\mu_{X \cup Y}^B(x) \geq \max(\mu_X^B(x), \mu_Y^B(x))$ *for any* $x \in U$
- $\mu_{X \cap Y}^B(x) \leq \min(\mu_X^B(x), \mu_Y^B(x))$ *for any* $x \in U$

As can be seen from the above properties, the fuzzy rough membership is a generalization form of the fuzzy membership and has a probabilistic nature. In other words, the fuzzy rough membership function is a conditional probability $\Pr(x \in X|u)$ which determines object x belongs to set X , giving knowledge u of x regarding attributes B .

7.4.3. Fuzzy Rough Equivalence Classes

The fuzzy rough equivalence class $[x]_B$ can also be defined using indiscernibility relation on objects of the universe of discourse. In this definition, B is a fuzzy set that determines the extent to which objects of the universe are close to x .

The usual fuzzy properties of reflexivity ($\mu_B(x, x) = 1$), symmetry ($\mu_B(x, y) = \mu_B(y, x)$), and transitivity ($\mu_B(x, z) \geq \mu_B(x, y) \wedge \mu_B(y, z)$) are preserved here too. The fuzzy rough equivalence class can be defined as follow:

$$\mu_{[x]_B}(y) = \mu_B(x, y) \quad \forall y \in X \quad (7-5)$$

The following principles should be satisfied for the fuzzy equivalence class $[x]_B$:

1. $\mu_{[x]_B} , \exists x, \mu_{[x]_B}(x) = 1$
2. $\mu_{[x]_B}(x) \wedge \mu_B(x, y) \leq \mu_{[x]_B}(y)$
3. $\mu_{[x]_B}(x) \wedge \mu_{[x]_B}(y) \leq \mu_B(x, y)$

The first principle indicates that an equivalence class is non-empty. The second principle states that elements in y 's neighbourhood are in the equivalence class of y . The final principle shows that any two elements in $[x]_B$ are related via B . Using these definitions, the concepts of the rough-fuzzy and fuzzy rough set can be introduced.

Using fuzzy equivalence class definition which means that the decision and the conditional attributes may all be fuzzy, the lower and upper approximations can be redefined as follow (Yao, 1998; De Cock et al., 2007):

$$\mu_{\underline{X}}([x]_B) = \inf_x \max\{1 - \mu_{[x]_B}(x), \mu_X(x)\} \quad \forall i \quad (7-6)$$

$$\mu_{\bar{X}}([x]_B) = \sup_x \min\{\mu_{[x]_B}(x), \mu_X(x)\} \forall i \quad (7-7)$$

Where $[x]_B$ denotes a fuzzy equivalence class and tuple $\langle \underline{X}, \bar{X} \rangle$ is a fuzzy rough set. If all the equivalence classes are crisp, then these definitions can represent traditional rough sets. In the following section, applying the fuzzy rough set to reduce dataset dimensionality is explained.

7.4.4. Dimensionality Reduction Using Fuzzy Rough Set

Consider the crisp partitioning $U/D = \{\{P_1, P_2, P_3, P_6\}\{P_4, P_5, P_7\}\}$ for the dementia information system presented in table 2 which contains two equivalence classes $Dementia = \{P_1, P_2, P_3, P_6\}$ and $No Dementia = \{P_4, P_5, P_7\}$. Using the rough set concept, the elements of the universe of discourse belong to these two classes with a membership of one or zero, e.g., for the class $Dementia$, the objects P_4, P_5 and P_7 have a membership of zero. By extending the crisp membership of these objects to fuzzy values in the interval $[0,1]$ the crisp partitioning U/D becomes a fuzzy partition.

In order to calculate the *reducts*, the lower approximations are calculated using equation (7-6). Similar to the rough set approach, the positive region is determined using the equivalence classes in U/D . The membership of each object x in the universe of discourse to the fuzzy equivalence class $[x]_B$ is obtained from the definition of $[x]_B$. For example, for the first equivalence class $X = \{P_1, P_2, P_3, P_6\}$, both $\mu_{\{P_1, P_2, P_3, P_6\}}(Dementia_D)$ and $\mu_{\{P_1, P_2, P_3, P_6\}}(NoDementia_D)$ should be calculated, because both $Dementia_D$ and $NoDementia_D$ are the fuzzy equivalence classes in U/D . The membership of each object x in the universe to the fuzzy equivalence class $Dementia_D$ is obtained from the definition of $Dementia_D$. Since U/D in this case is crisp, the extent to which x belongs to $\{P_1, P_2, P_3, P_6\}$ is 1 only if $x \in \{P_1, P_2, P_3, P_6\}$, and 0 otherwise.

As it was discussed before, the crisp positive region in the rough set theory is defined as the union of the lower approximations. Therefore, using the fuzzy extension principle, the fuzzy positive region of a fuzzy equivalence class $[x]_B \in U/D$ can be defined as:

$$\mu_{POS_D}([x]_B) = \sup_{x \in U/D} \mu_{\underline{x}}([x]_B) \quad (7-8)$$

In the same way, using the fuzzy extension principle, the membership of an object $x \in U$ to the fuzzy positive region can be calculated using equation (7-8).

$$\mu_{POS_D}(x) = \sup_{[x]_B \in U/D} \min(\mu_{[x]_B}(x), \mu_{POS_D}([x]_B)) \quad (7-9)$$

According to the equation (7-9), the extent to which object x belongs to POS_D is the degree to which it belongs to each fuzzy equivalence class and the degree to which the equivalence class belongs to POS_D .

Eventually, using the definition of the fuzzy positive region, the fuzzy dependency function can be defined as follows:

$$\gamma'_D(Q) = \frac{|\mu_{POS_D}(x)|}{|U|} = \frac{\sum_{x \in U} \mu_{POS_D}(x)}{|U|} \quad (7-10)$$

The fuzzy dependency, similar to the rough set approach, is the proportion of objects that are discernible in the universe of discourse using only the information in D . In other words, the fuzzy dependency is the cardinality of $\mu_{POS_D}(x)$ divided by the total number of objects in the universe of discourse.

Since objects may belong to many equivalence classes in the fuzzy rough set, to find the fuzzy dependency for all various subsets of the original attribute set, the cartesian product of all indiscernible sets of objects should be calculated:

$$U/D = \otimes\{a \in D: U/IND(a)\} \quad (7-11)$$

For example, consider $D = \{A, B\}$ and $U/A = \{Dementia_A, NoDementia_A\}$, $U/B = \{Dementia_B, NoDementia_B\}$:

$$\begin{aligned} U/D &= U/A \otimes U/B = \{Dementia_A, NoDementia_A\} \otimes \{Dementia_B, NoDementia_B\} \\ &= \left\{ \begin{aligned} &\{Dementia_A, Dementia_B\}, \{Dementia_A, NoDementia_B\}, \{NoDementia_A, Dementia_B\}, \\ &\{NoDementia_A, NoDementia_B\} \end{aligned} \right\} \end{aligned}$$

Each set in U/D is an equivalence class and the extent to which an object belongs to such a class is calculated by the conjunction of the membership degrees to all sets in the class:

$$\mu\{[x]_{B_1}, \dots, [x]_{B_n}\}(x) = \min(\mu_{[x]_{B_1}}(x), \mu_{[x]_{B_2}}(x), \dots, \mu_{[x]_{B_n}}(x)) \quad (7-12)$$

7.4.5. Computational Reduction Using Fuzzy Rough Set

As it was mentioned before, reducing irrelevant or redundant attributes from the dataset can improve the classification performance. The QUICKREDUCT algorithm is a conventional method that uses rough set theory to find the minimal subset of attributes that preserve the same classification power as the original dataset.

As it can be seen, the algorithm starts with an empty set R and adds those attributes that result in the greatest increase in $\gamma_R(D)$, until it reaches its maximum possible value for the dataset (usually 1). The algorithm is a non-optimal heuristic and suffers from information loss in the discretization procedure. The fuzzy version of the QUICKREDUCT algorithm is introduced to improve the computation time and decrease information loss. Fuzzy QUICKREDUCT uses the dependency function γ' to choose which attributes to add to the current reduct candidate in the same way as the original QUICKREDUCT process. However, the algorithm terminates when the addition of any remaining attributes does not increase the dependency.

QUICKREDUCT algorithm

```
1:    $R \leftarrow \{\}, \gamma'_{best} \leftarrow 0, \gamma'_{prev} \leftarrow 0$ 
2:   Do
3:     Set  $T \leftarrow R$ 
4:      $\gamma'_{prev} \leftarrow \gamma'_{best}$ 
5:      $\forall x \in (C - R)$ 
6:     If  $\gamma'_{R \cup \{x\}}(D) > \gamma'_T(D)$ 
7:        $T \leftarrow R \cup \{x\}$ 
8:        $\gamma'_{best} \leftarrow \gamma'_T(D)$ 
9:      $R \leftarrow T$ 
10:  Until  $\gamma'_{best} = \gamma'_{prev}$ 
18:  Return  $R$ 
```

Figure 7-23: Fuzzy rough QUICKREDUCT algorithm

The minimal reduct in fuzzy QUICKREDUCT is found using the information on the degree of membership. Although the introduction of some threshold values to terminate the algorithm if the change in dependency is not significant enough could alleviate this problem, still the computation time is a major drawback.

7.5. Conclusion

In this chapter, several methods for pre-processing and feature extraction of Spatio-temporal EEG data are proposed and illustrated on a real-world problem related to brain neurodegenerative disease. Also, a novel feature extraction method is introduced based on a combination of the proposed Chaotic Quantum-Inspired Evolutionary Algorithm and fuzzy rough set theory for clinical static data related to the EEG recordings. The purpose of this approach is to provide a potential for weighted learning of Spatio-temporal data based on the clinical and demographical observations of the subjects in the proposed methods.

Chapter 8 . Conclusions and Recommendations for Future Work

Abstract

This chapter summarises the main contributions of this thesis along with paving ways for further research. The chapter also addresses the following research question related to new ideas for further development.

RQ9 “How can the mechanism of producing spiking patterns be improved in a self-adaptive manner according to the spiking neural network activities by introducing a novel neuron model based on nuclear physics concepts?”

8.1. Main Contributions of The Thesis

The main contributions of the thesis are in the proposed new theoretical methods for deep learning in SNN inspired by brain-and quantum principles, namely:

1. Brain-like computational unified framework EPUSSS to perform both prediction and classification tasks by introducing a hierarchical two-level learning, excitatory/inhibitory neurons, 4 different synaptic neurotransmitter receivers with different delays and functionality in propagating information, energy function, and pruning mechanism to support life-long learning.
2. Chaotic quantum gate to improve the QIEA search mechanism to help parameter optimisation and developing learning algorithm for a quantum inspired SNN and search mechanism for associative memory.

3. Quantum inspired brain-like computational unified framework QISNN to perform both prediction and classification tasks by introducing hierarchical two-level learning, quantum neurons, quantum synapses, and quantum learning.
4. Quantum associative memory with two different quantum search mechanisms using Grover's algorithm and CQIEA for pattern retrieval.
5. Prep-processing and feature extraction from Spatio-temporal data for real-world applications.

In this thesis, I have explored the human brain's physiological components and behaviour in the learning process and forming memories of past events and quantum physics principles in computational models. I have proposed two novel frameworks for deep learning in spiking neural networks, including eight new algorithms inspired by the brain physiological learning behaviour, quantum computation principles, and a combination of brain-like and quantum-inspired principles to learn, predict, classify, and recall Spatio-temporal data.

These two novel frameworks have a 3D spiking neuronal network topology according to the Talairach brain atlas to predict several steps ahead of the brain data and classify them using a 2-level hierarchical structure. The proposed frameworks EPUSSS and QISNN benefit from biologically plausible characteristics and quantum computational principles to provide robust, fast, and accurate computational models.

Both EPUSSS and QISNN can learn the spatial and temporal correlation of the input data by mapping them into a 3D space and capturing their temporal relationship in a two-level learning hierarchy using hybrid learning mechanisms. Then a memory of this learnt pattern is created using two different Quantum

Inspired Associative memory models and later used to recall and match new patterns to the data stored in the quantum memory.

These novel frameworks were tested on neurorehabilitation of Wrist Movement and Dementia disease EEG data for prediction and classification as a proof of concept.

In the following, a brief overview of my PhD study contributions is provided.

8.1.1 Chapter 3 Contributions and Future Directions

In chapter 3 of this thesis, a novel framework for Evolving Predictive Unsupervised Supervised deep learning algorithms for Spike Streams (EPUSSS) is proposed.

EPUSSS is inspired by biological learning behaviour in the human brain and performs both prediction and classification of spatio-temporal data in a two-level hierarchical fashion under one unified framework. The EPUSSS architecture is influenced by the NeuCube framework and has three major modules:

1. The encoder module which uses BSA algorithm that encodes input data into spike trains.
2. A 3D SNN reservoir module in which the local learning mechanism of EPUSSS governs the synaptic efficacy in a semi-supervised way.
3. The output module which deploys the global learning mechanism of EPUSSS, combined with a modified deSNN model to provide a neuromodulatory feedback to modify synaptic efficacy in a supervised way.

In the EPUSSS 3D SNN reservoir, the input neuron is not only connected to the reservoir neurons by synaptic connections but also is connected to other input neurons by “hypersynaptic” connections. This ability of EPUSSS, allows spatial correlation as well as temporal correlation amongst input patterns’ channels to be captured and the collaboratively

impact on the learning process like the biological learning in which different parts of the brain with various inputs empower the learning experience in an individual.

The input neurons and reservoir neurons are mapped in a 3D SNN structure according to their 3D spatial coordinates. This visual representation allows observing the interaction between input neurons and reservoir neurons and amongst input neurons themselves. In the case of brain data, the coordinates are set according to the Talairach brain atlas, which helps to perceive the neuronal behaviour in the presence of input patterns in a brain-like simulation. This feature of EPUSSS is not only interesting from a computational aspect but also can help neuroscientists to have a better understanding of brain data (e.g., EEG, fMRI, MRI, etc.).

The learning mechanism in EPUSSS is designed in a two-level hierarchy to mimic physiological learning locally and globally to reinforce learning behaviour in the network.

EPUSSS local learning captures the Spatio-temporal correlation in the input data space by adapting to the statistical properties of the neuronal firing behaviours. The global learning mechanism of EPUSSS uses the concept of neuromodulatory signals in the brain, which influence the synaptic activities by reward and punishment to help learn and create memories. Using neuromodulatory concepts, neurons in the proposed model get feedback of reward or punishment to modify their spiking activities to form memories and recognize the learnt pattern in the presence of the external stimuli (i.e., target class labels).

The EPUSSS framework is theoretically capable of life-long learning with the ability of transfer learning. To achieve the life-long learning ability, EPUSSS has several important characteristics that help to learn from the input data continuously without being saturated and are listed as follow:

- EPUSSS has 80 percent excitatory and 20 percent inhibitory neurons in its spiking neural network in such a way that the inhibitory neurons spike faster than excitatory neurons to balance the network spiking behaviour.
- The synaptic connections in EPUSSS have four types of neurotransmitter receptors:
 - Excitatory neurons have two types of neurotransmitter receptors:
 - AMPAR: fast excitatory synaptic transmission receptors
 - NMDAR: slow excitatory synaptic transmission receptors
 - Inhibitory neurons have two types of neurotransmitter receptors:
 - GABAAR: fast inhibitory synaptic transmission
 - GABABR: slow inhibitory synaptic transmission

By introducing synaptic delays, the synaptic transmission time can be controlled to mimic fast and slow signal transmission like biological neurotransmitters behaviour.

- Synaptic pruning which is part of the brain development is another feature of EPUSSS in which neurons that do not show activity for a certain time-window will lose their synaptic connections gradually by decreasing their post synaptic weights. This mechanism helps to eliminate noisy neurons from the learning process and improve the model prediction and classification performance.

Another important feature of EPUSSS is using a Lyapunov energy function as an extra control mechanism over synaptic weight changes to guarantee life-long learning without the risk of network saturation.

The EPUSSS framework has been tested on two real-world datasets containing EEG recordings of neurorehabilitation of Wrist Movement and Dementia disease and its results and performance compared to NeuCube and LSTM. The EPUSSS model outperforms the two

models. However, due to a large number of neurons, complicated connectivity structure, and the iterative process of local and global learning, the computation time of EPUSSS increases polynomially by the size of the problem. This limitation of the EPUSSS model has been addressed in chapter 5 by introducing a quantum inspired SNN model to reduce the computation time. Another aspect of the EPUSSS framework that needs to be explored more is the encoding algorithm. Currently, the BSA algorithm has been used to encode the input data to spike train. Although BSA has been proved to be less sensitive to changes in the filter and threshold and as a result provides a more accurate spike train even for bipolar cases, the nature of the problem should be considered to choose a proper encoding algorithm. As a future direction, this area needs to be investigated by performing more experiments with various data and different encoding methods.

8.1.2 Chapter 4 Contributions and Future Directions

In chapter 4, a novel search algorithm is proposed based on chaos theory and the quantum-inspired principals called Chaotic Quantum-Inspired Evolutionary Algorithm (CQIEA). The Chirikov chaotic map is used here to improve the convergence rate and the quality of the final solution of the Quantum-Inspired evolutionary algorithm. Chirikov standard map is generated by a time-dependent Hamiltonian system and has all the requirements that a quantum gate must have such as being a reversible unitary operator. The Chirikov standard map is used as a q-gate (diversity operator) for QIEA to explore and exploit the problem search space more efficiently. In this thesis, it was shown that the Chirikov standard map can produce both chaotic and regular behaviour with different values of K . Therefore, during the evolution, the dynamic of the system can be controlled efficiently using adaptive values for K . At the beginning of the search/optimisation problem by setting a higher value for K and increasing the chaotic behaviour of the system the search space can be explored more efficiently. By decreasing the values of K over time, the system will produce more regular behaviour than chaotic dynamics

which can result in converging to more stable states and exploit search space to find an optimum solution. In this thesis, both discrete and continuous version of CQIEA has been proposed to serve as a learning mechanism to adjust SNN's weights, as an optimisation technique to improve proposed models' performance, and as a search mechanism for the proposed Quantum Associative Memory. The results of experiments prove the effectiveness of the proposed approach in both continuous and discrete environment. As a future direction, the Chirikov standard map will be investigated more to be directly used to change the current state of a quantum system in a chaotic manner instead of serving as the chaotic angular position for the rotation gate.

8.1.3 Chapter 5 Contributions and Future Directions

In chapter 5, a novel Quantum Inspired Spiking Neural Network (QISNN) is introduced based on the probabilistic spiking neuron model (Kasabov, 2010). The QISNN framework, which is a brain-like quantum-inspired model, follows EPUSSS architecture and the two-level hierarchical learning structure. In the QISNN, the neuron model, synaptic connections, and learning rules are a combination of biologically plausible models and quantum-inspired models introduced in chapters 3 and 4, respectively.

QISNN like EPUSSS architecture has the same three major modules in its framework: the encoder module which uses BSA algorithm that encodes input data into spike trains, the 3D SNN reservoir module in which the local learning mechanism of QISNN governs the synaptic efficacy in a semi-supervised way and the output module which deploys the global learning mechanism of QISNN combined with modified deSNN model to update synaptic efficacy in a supervised way. QISNN uses spiking neuron models such as LIF and Izhikevich combined with a quantum model to control neurons' firing behaviour. Despite the learning mechanisms of QISNN being fundamentally similar to EPUSSS, the learning algorithms, neuron's firing

behaviour, and post-synaptic connections are controlled and governed by quantum computation models.

The QISNN model learns from a time series data to predict the next values of the input data in a certain time-window in the reservoir as part of local learning process. Predictive modelling is achieved through a semi-supervised local learning mechanism using the error of prediction in a certain time-window and rule that adjust the efficacy of synaptic weights of the reservoir neurons. The output neurons inspired by deSNN model's concept are created incrementally for each training sample to associate the input data with a class label in the global learning stage. For this purpose, all the input neurons and reservoir neurons (or just input neurons) are connected to output neurons using RO method and their synaptic weights are adjusted based on QISNN global learning rule using a neuromodulatory signal that is generated by the results of classification in the training phase. The principle behind the QISNN is to perform both prediction and classification in a two-level hierarchical learning structure. In the proposed model the input neurons contribute strongly to both local and global learning.

The QISNN framework benefits from biologically plausible learning principals and quantum computation concepts. Therefore, theoretically, it is a potent tool compared to other machine learning models and can improve the computation complexity of EPUSSS to overcome the slow learning process speed.

As a future direction, QISNN's needs to be investigated by performing more experiments with various data. Also, since the model is a hybrid of the SNN model and quantum computation concepts, there is a potential for designing combined neuromorphic and quantum computing hardware.

Quantum computers, which was first invented in 1998, are immensely difficult to engineer, build and program. Quantum computers are limited errors arising from noise, faults, and loss

of quantum coherence in their operations. The loss of coherence caused by vibrations, temperature fluctuations, electromagnetic waves and other interactions with the outside environment can cause damage to the quantum computer's components. Thus, the current quantum computers are not quite reliable even for programs with modest execution times. On the other hand, neuromorphic hardware, which adapts the SNN properties in its architecture, is proven to be fast and energy-efficient and can support high-performance applications. The proposed models in this thesis have the potential to be implemented on quantum computers, neuromorphic hardware, and even hybrid hardware that supports both SNN and quantum computing architecture. This can be an exciting and promising area to be explored further in the future.

8.1.5 Chapter 6 Contributions and Future Directions

In chapter 6, a novel quantum associative memory scheme for spiking neural networks is proposed to perform a fast, online pattern recognition for classification and prediction tasks just by partial pattern exposure. To model the quantum associative memory for spiking neural networks, the network requires the ability to store patterns in a medium and the ability to recall those patterns later. By embedding the quantum associative memory into EPUSSS or QISNN models, the spike encoding module of these frameworks will convert the information into spike trains which is a binary representation. Each input neuron in the EPUSSS or QISNN models can act as a register to store memory of the produced patterns by its locally connected reservoir neurons. To find a pattern in the quantum associative memory, the system should be measured and collapsed with the near certainty to the basis state which corresponds to the sought pattern. To this end, the retrieving quantum algorithm inverts the phase of desired basis state and all the other basis states using the quantum gate operator in a way that the probability amplitude of the desired basis state increases to near unity and the probability amplitude all the other states decrease to near zero.

In the novel proposed quantum associative memory, the learnt patterns which are produced during the local learning of either the EPUSSS model or QISNN by the input neurons (input twin neurons) will be stored as qubits in the quantum memory. In this way, the associative memory is integrated into both models. After local learning, when the input spike trains dynamic is captured and learnt by the network, the patterns can be recalled in a more time-efficient manner. After storing patterns in the local learning phase of the EPUSSS model or QISNN model, two quantum search algorithms can be used to retrieve the sought pattern, Grover's algorithm and the CQIEA algorithm. Both algorithms use Jaccard proximity measurement to compute the similarity between two binary inputs. Theoretically, both Grover's algorithm and CQIEA algorithm can search unstructured databases in an acceptable time for the desired patterns and retrieving the sought pattern even in the presence of partial noisy input data. As a future direction and to reveal the power of quantum associative memory, more experiments with large scale dataset needed to be conduct. As mentioned before, combining the SNN models with quantum computing principals can lead to powerful models with computational advantages and is a promising future for the new generation of quantum computers and neuromorphic hardware.

8.1.6 Chapter 7 Contributions and Future Directions

In chapter 7, several methods for pre-processing and feature extraction of Spatio-temporal EEG data are proposed and illustrated on a real-world problem related to brain neurodegenerative disease. Also, a novel feature extraction method is introduced based on a combination of the proposed Chaotic Quantum-Inspired Evolutionary Algorithm and fuzzy rough set theory for clinical static data related to the EEG recordings. The purpose of this approach is to provide a potential for weighted learning of Spatio-temporal data based on the clinical and demographical observations of the subjects in the proposed methods.

To produce precise spiking activities, the idea of a novel neuron model inspired by the nuclear physic concept of Microtron Accelerator is presented as a future study to add a self-adaptable delay mechanism to the SNNs.

Due to the extensive research, this thesis has a great potential for future work. One original idea by the author is explained in this section addressing the below research question:

“How can the mechanism of producing spiking patterns be improved in a self-adaptive manner according to the spiking neural network activities by introducing a novel neuron model using nuclear physics concepts?”

8.2 Microtron Neuron Model

Assigning the fixed random synaptic delays to neurons in polychronous spiking networks restricts the PNGs to produce fixed random patterns. This blind inefficient mechanism for producing patterns ignores network real spiking activities.

Despite biological plausibility and acceptable performance of previous neuron models (Hodgkin and Huxley, 1952; Gerstner and Kistler, 2002; Izhikevich, 2003) in spiking neural networks, these models are not capable to produce adaptable delays for emitting spikes in the proposed polychronous spiking neural network. To have an efficient adaptable delay mechanism for PNGs, I have proposed a novel neuron model using the nuclear physics concept of Microtron Accelerator.

The Microtron is a cyclic particle accelerator for low to intermediate energies (up to some tens of MeV). In fact, Microtron is a linear accelerator that accelerates particles using a high-frequency electric field to pass through a resonant cavity. The Microtron can only accelerate particles whose kinetic energy can be increased by an amount comparable to their rest energy

during one single pass through the accelerating cavity. Therefore, the only particles that can be accelerated by the Microtron are electrons and positrons (Lidbjork, 1994; Sobenin et al., 1996; Kostin et al., 1999; Froelich and Manca, 1975; Penner et al., 1981; Kaiser, 1956).

The layout of the classical Microtron can be seen in Figure 8-2.

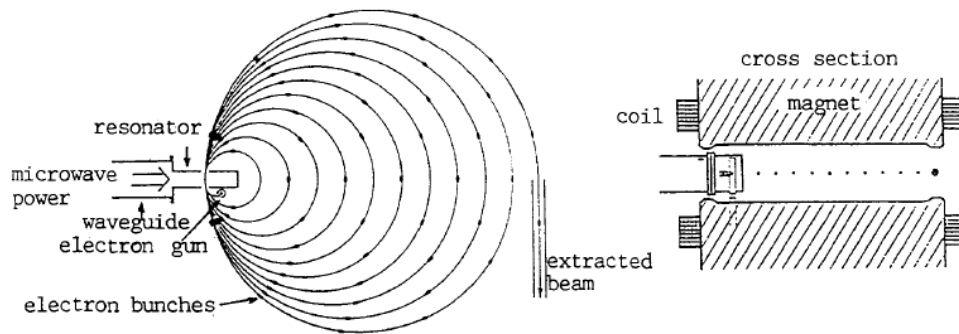


Figure 8-1: The classical Microtron (Lidbjork, 1994)

Microtron accelerates a particle (electron/positron) in a relativistic motion to pass it through a gap. In this gap, there is a (spatially) uniform time-dependent electric field that can either accelerate or decelerate the particle which results in a change of particle kinetic energy. The particle paths are different for each pass because of their increasing momentum. Therefore, the time needed for the particle to pass through the gap is proportional to the pass numbers. The slow particles need one electric field oscillation while the faster ones need an integer multiple of this oscillation. Outside the gap the particle orbits around due to a uniform magnetic field B .

The acceleration mechanism in the Microtron is based on a relation between the magnetic field and the frequency and amplitude of the accelerating voltage. The magnetic field, B , and the energy gain, ΔE are adjusted so that the revolution time in the first orbit can be calculated by:

$$T_1 = (\frac{2\pi}{ec^2B})E_1 \quad (8-1)$$

Where T_1 is the revolution time and E_1 is the total energy in the first orbit, e is the particle energy and c is the light speed. For all other orbits the revolution time must increase relative to the previous orbit:

$$\Delta T = (\frac{2\pi}{ec^2B})\Delta E \quad (8-2)$$

Where ΔT is the difference in revolution time between two orbits and ΔE is the energy gain in the resonator. The total energy in the first orbit is:

$$E_1 = E_0 + E_i + \Delta E \quad (8-3)$$

Where E_0 is the particle rest energy and E_i is the injection energy.

The total energy in orbit n and the magnetic field are then:

$$E_n = \Delta E(T_n/\Delta T + n - 1) \quad (8-4)$$

$$B = (\frac{2\pi}{ec^2})\Delta E/\Delta T \quad (8-5)$$

According to the above explanation about the Microtron working principle, I devised the Microtron Neuron Model as follow:

In my proposed scheme each neuron in the spiking neural network is a Microtron accelerator that can emit electrons (excitatory spikes) or positrons (inhibitory spikes) that respectively depolarizes or hyperpolarizes post-synaptic neurons. Each Microtron neuron has a magnetic

field as a memory of spiking activity which can be weakened or strengthened by the injection energy (pre-synaptic spikes) based on the type of emitted particles. Equation (8-3) can be considered as neural integration equilibrium in conventional neuron models. The memory will be updated with a modified STDP or EPUSSS learning rule to capture temporal spiking activities by the post-synaptic neuron (see equations (8-4 and 8-5)). The time of emitting a spike and the frequency of firing is controllable according to the Microtron neuron energy as it can be seen in equation (8-2). Therefore, the new Microtron model can control spike emitting time and frequency in order to improve polychronous network performance in a more effective manner using network activity information to produce self-adaptive patterns.

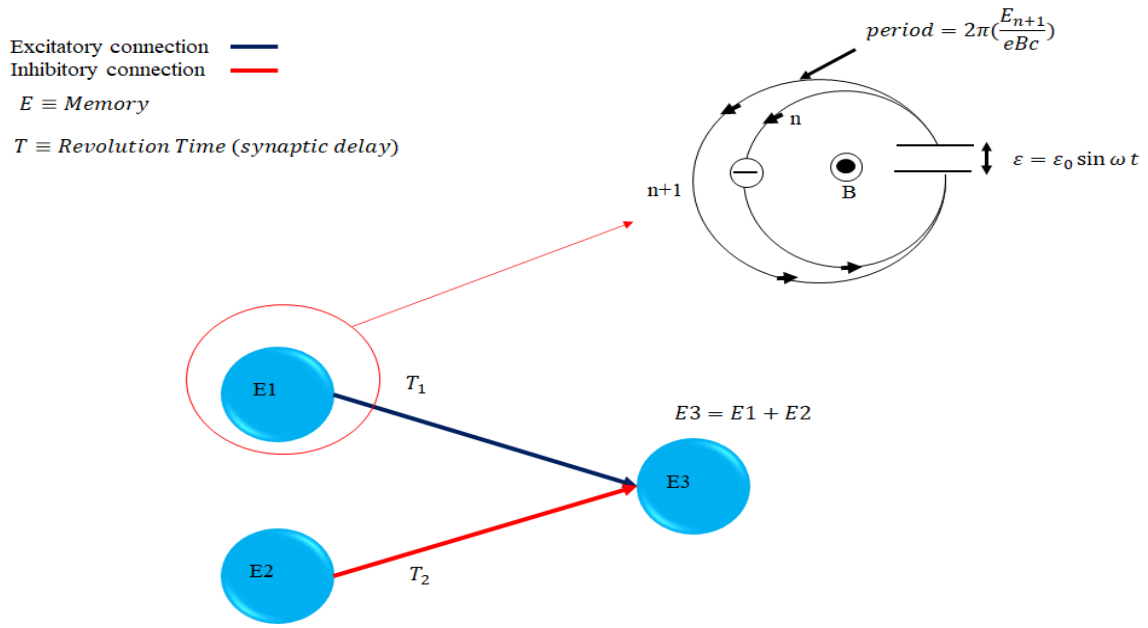


Figure 8-2: The novel Microtron model

Figure 8-2, shows the Proposed Microtron neuron model. The performance of the new model in the proposed network will be compared with LIF and Izhikevich neuron models in terms of network ability to perform prediction tasks.

8.2.2 Other Directions for Further Research

Other directions for further research include:

- Full implementation of the proposed in this thesis methods.
- Experimentation of the proposed methods on large scale data.
- Evaluation of the applicability of the proposed methods for what of class of Spatio-temporal data and problems they will be most efficient.
- Application of the proposed methods on real-world problems.

References

- Agnes, E. J., Erichsen, Jr. R., & Brunnet, L. G. (2012). Model architecture for associative memory in a neural network of spiking neurons. *Physica A: Statistical Mechanics and its Applications*, 391(3), 843-848.
- Al-Fahoum, A. S., & Al-Fraihat, A. A. (2014). Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains. *International Scholarly Research Notices*, (2014), 7 pages, <https://doi.org/10.1155/2014/730218>.
- Alvarez-Estevéz, D. (2021). European Data Format. Retrieved from <https://www.edfplus.info>.
- Anwani, N. & Rajendran, B. (2020). Training multi-layer spiking neural networks using NormAD based Spatio-temporal error backpropagation. *Neurocomputing*, 380, 67-77, doi: <https://doi.org/10.1016/j.neucom.2019.10.104>.
- Aswolinskiy, W., & Pipa, g. (2015). RM-SORN: a reward-modulated self-organizing recurrent neural network. *Frontiers in Computational Neuroscience*, 9, 15 pages, doi: 10.3389/fncom.2015.00036.
- Alatas, B. & Akin, E. (2009). Multi-Objective Rule Mining Using a Chaotic Particle Swarm Optimisation Algorithm. *Journal of Knowledge-Based Systems*, 22, 455-460.
- Bandemer H. & S. Gottwald (1995). Fuzzy Sets, Fuzzy Logic, Fuzzy Methods, with Applications, Wiley, Chichester.
- Baranger, M. (2013). Chaos, Complexity, and Entropy: A Physics Talk for Non-Physicists. <http://necsi.edu/projects/baranger/cce.pdf>.
- Bascil, M. S., Tesneli A. Y., & Temurtas, F. (2016). Spectral feature extraction of EEG signals and pattern recognition during mental tasks of 2-D cursor movements for BCI using SVM and ANN. *Australas Phys Eng Sci Med*. 39(3), 665-76.
- Bassett, D. S., & Bullmore, E. T. (2017). Small-World Brain Networks Revisited. *Neuroscientist*, 23(5), 499–516, doi: 10.1177/1073858416667720.
- Beiser, A. (1987). Concepts of Modern Physics (4th ed.). *McGraw-Hill (International)*. ISBN 0-07-100144-1, pages, 559.
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1), 1-127, <http://dx.doi.org/10.1561/22000000006>.
- Bhagat, Y. (2007). Diffusion Tensor Imaging of the Human Brain (Doctoral dissertation in Biomedical Engineering). Retrieved from https://www.researchgate.net/publication/291346848_Diffusion_Tensor_Imaging_of_the_Human_Brain.

- Blanco A., D., & Ramirez, R. (2019). Evaluation of a Sound Quality Visual Feedback System for Bow Learning Technique in Violin Beginners: An EEG Study. *Frontiers in Psychology*, 10, doi: 10.3389/fpsyg.2019.00165.
- Blumenthal, A., (2021). Statistical properties for compositions of standard maps with increasing coefficient. *Ergodic Theory & Dynamical Systems*, 41(4), 981-1024.
- Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H., & Maza, D. (2000). The Control of Chaos: Theory and Applications. *Journal of Physics Report*, 329, 103-197.
- Bohte, S. M., Kok, J. N., & Poutré, H. L. (2000). SpikeProp: Backpropagation for Networks of Spiking Neurons. *Symposium conducted at the meeting of the ESANN 2000, 8th European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 26-28, 2000, Proceedings.
- Bohte, S. M., Kok, J. N., & Poutre, H. L. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48, 17–37. [https://doi-org.ezproxy.aut.ac.nz/10.1016/S0925-2312\(01\)00658-0](https://doi-org.ezproxy.aut.ac.nz/10.1016/S0925-2312(01)00658-0).
- Brusca, S., Capizzi, G., Sciuto, G. L., & Susi, G. (2017). A New Design Methodology to Predict Wind Farm Energy Production by Means of a Spiking Neural Network–Based System. *International Journal of Modelling*, 14 pages, <https://doi.org/10.1002/jnm.2267>.
- Bullmore, E., & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10, 186–198.
- Cai, M., & Liu, J. (2016). Maxout Neurons for Deep Convolutional and LSTM neural networks in Speech Recognition. *Journal of Speech Communication*, 77, 53-64.
- Cao, Y., Chen, Y., & Khosla, D. (2015). Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition. *International Journal of Computer Vision*, 113, 54-66.
- Cedazo-Minguez, A., & Winblad, B. (2010). Biomarkers for Alzheimer’s Disease and Other Forms of Dementia: Clinical Needs, Limitations and Future Aspects. *Experimental Gerontology*, 45(1), 5–14.
- Celecia, Alimed & Vellasco, Marley. (2020). Chaotic Quantum-inspired Evolutionary Algorithm: enhancing feature selection in BCI. 1-8. 10.1109/CEC48606.2020.9185608.
- Chen, G., Liu, L., Song, P., & Du, Y. (2014). Chaotic Improved PSO-Based Multi-Objective Optimisation for Minimization of Power Losses and L index in Power Systems. *Journal of Energy Conversion and Management*, 86, 548-560.
- Chen, Y., Guo, H., Gao, X. & Wu, J. (2020). Quantum Spike Neural Network. arXiv:2011.05062v2 [quant-ph].
- Chen, Z., Yuan, X., Ji, B., Wang, P., & Tian, H. (2014). Design of a Fractional Order PID Controller for Hydraulic Turbine Regulating System Using Chaotic Non-Dominated Sorting Genetic Algorithm II. *Journal of Energy conversion and Management*, 84, 390-404.

Chirikov, B. V. (1971). Research Concerning the theory of nonlinear resonance and stochasticity. *Perprint N 267, Institute of Nuclear Physics, Novosibirsk, 1969, (Engl. Trans., CERN Trans. 71-40 (1971))*.

Chirikov, B.V., Izrailev, F., & Shepelyanskij, D.L. (1981). Dynamical stochasticity in classical and quantum mechanics. *Soviet Scientific Reviews*. Section C. 2.

Chuang, L. Y., Tsai, S. W., & Yang, C. H. (2011). Chaotic Catfish Particle Swarm Optimisation for Solving Global Numerical Optimisation Problems. *Journal of Applied Mathematics and Computation*, 217, 6900-6916.

Clark, D. P., & Pazdernik, N. J. (2013). Molecular Biology (Second Edition), *Academic Press*, ISBN 9780123785947, <https://doi.org/10.1016/B978-0-12-378594-7.00052-4>.

Columbia University School of Engineering and Applied Science. (2019). Specific neurons that map memories now identified in the human brain. ScienceDaily. Retrieved November 12, 2021, from www.sciencedaily.com/releases/2019/11/191112095740.htm.

De Cock, M., Cornelis, C., & Kerre, E. E. (2007). Fuzzy Rough Sets: The Forgotten Step. *IEEE Transaction on Fuzzy Systems*, 15(1), 121-130.

Defoin-Platel, M., S.Schliebs, N.Kasabov, Quantum-inspired Evolutionary Algorithm: A multi-model EDA, *IEEE Trans. Evolutionary Computation*, vol.13, No.6, Dec.2009, 1218-1232

DeKosky, S. T., & Marek, K. (2003). Looking Backward to Move Forward: Early Detection of Neurodegenerative Disorders. *Science*, 302(5646), 830–834.

Deng, L., & Yu, D. (2013). Deep Learning: Methods and Application. *Foundations and Trends in Signal Processing*, 7(3-4), ISSN: 1932-8346.

Dhoble, K., Nuntalid, N., Indiveri, G., & Kasabov, N. (2012). Online Spatio-temporal Pattern Recognition with Evolving Spiking Neural Networks Utilising Address Event Representation, Rank Order, and Temporal Spike Learning. *IEEE World Congress on Computational Intelligence*, (pp. 1-7). Brisbane, Australia.

Dias, Eduardo & Vellasco, Marley & Abs da Cruz, Andre. (2020). Quantum-inspired neuro coevolution model applied to coordination problems. *Expert Systems with Applications*. 167. 114133. 10.1016/j.eswa.2020.114133.

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., & Pfeiffer, M. (2015). Fast-Classifying, High-Accuracy Spiking Deep Networks Through Weight and Threshold Balancing. *IEEE International Joint Conference on Neural Networks (IJCNN)*, (pp. 1-8), Killarney.

Diehl, P. U., et al. (2016). Conversion of Artificial Recurrent Neural Networks to Spiking Neural Networks for Low-Power Neuromorphic Hardware. *arXiv preprint arXiv:1601.44187v1*.

Ditto, W., & Munakata, T. (1995). Principles and Applications of Chaotic Systems. *Journal of Communication of the ACM*, 38(11), 96-102.

- Dommelen, L. V., (2018), Fundamental Quantum Mechanics for Engineers. Retrieved from http://www.dommelen.net/quantum2/style_a/index.html.
- Eisberg, R. (1974). Quantum Physics of Atoms, Molecules, Solids, Nuclei and Particle. New York: John Wiley & Sons Inc. ISBN-978-0-471-234647.
- Enayatifar, R., Abdullah, A. H., & Isnin, I. F. (2014). Chaos-Based Image Encryption Using a Hybrid Genetic Algorithm and a DNA Sequence. *Journal of Optics and Lasers in Engineering*, 56, 83-93.
- Esposito, S. & Naddeo, A. (2013). The Genesis of the Quantum Theory of the Chemical Bond, *arXiv:1309.4647v1 [physics.hist-ph]* 18 Sep 2013.
- Esser, S. K., Appuswamy, R., Merolla, P. M., Arthur, J. V., & Modha, D. S. (2015). Backpropagation for Energy-Efficient Neuromorphic Computing. *Advanced in Neural Information Processing Systems 28 (NIPS)*, 1117-1125.
- Esser, S. K., et al. (2016). Convolutional Networks for Fast, Energy-Efficient Neuromorphic Computing. *Proceeding of the National Academy of Science of the United States of America*, 113(41), 11441-11446.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S., Blau, H., & Thrun, S. (2017). Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. *Nature*, 542, 115–118.
- Ezhov, A., Nifanova, A., & Ventura, D. (2000). Distributed Queries for Quantum-inspired associative memory. *Information Sciences*, 3-4, 271-293.
- Feynman, Richard (June 1982). Simulating Physics with Computers. *International Journal of Theoretical Physics*. 21 (6/7): 467–488.
- Fraday, E.P., & Sommer, F.T. (2019). Robust computation with rhythmic spike patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 116(36), 18050-18059.
- Farries, M. A., & Fairhall, A. L. (2007). Reinforcement Learning with Modulated Spike Timing–Dependent Synaptic Plasticity. *Journal of Neurophysiology*, 98, 3648–3665, doi:10.1152/jn.00364.2007.
- Frahm, K. M., & Shepelyansky, D. L. (2013). Poincare recurrences and Ulam method for the Chirikov standard map. *European Physical Journal B*, 86(7), 11 pages.
- Frémaux, N., & Gerstner, w. (2015). Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Frontiers in Neural Circuits*, 9(85), 19 pages, doi: 10.3389/fncir.2015.00085.
- Froelich, H. R., & Manca, J. J. (1975). Performance of Multiactivity Racetrack Microtron. *IEEE Transaction on Nuclear Science*, 22(3), 1758-1762.

- Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-Enhanced Accelerated Particle Swarm Optimisation. *Journal of Commun Nonlinear Sci Number Simulat*, 18, 327-340.
- Gardner, B., Sporea, I., & Gruning, A. (2014). Classifying Spike Patterns by Reward-Modulated STDP. *International Conference on Artificial Neural Networks-ICANN 2014*, (pp. 749-756), Lecture Notes in Computer Science, 8681, Springer, Cham, doi: https://doi.org/10.1007/978-3-319-11179-7_94.
- Gashler, M. S., & Ashmore, S. C. (2016). Modelling Time Series Data with Deep Fourier Neural Networks. *Journal of Neurocomputing*, 188, 3-11.
- Gaspard, P., & Wang, X. J., (1993). Noise, Chaos, and (ϵ, τ) -Entropy per Unit Time. Physics Reports, *Review Section of Physics Letters*, 235(6), 291-343.
- Gerstner, W., & Kistler, W. (2002). Spiking Neuron Models: Single Neurons, Populations, Plasticity. *Cambridge University Press*, ISBN: 978-0521890793.
- Gibson, T., Henderson, J.A., & Wiles, J. (2014). Predicting Temporal Sequences Using an Event-Based Spiking Neural Network Incorporating Learnable Delays. *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2014*, (pp. 3213-3220).
- Gibson, T., Heath, S., Quinn, R. P., Lee, A. H., Arnold, J. T., Sonti, T. S., Whalley, A., Shannon, G. P., Song, B. T., Henderson, J. A., & Wiles, J. (2014). Event-Based Visual Data Sets for Prediction Tasks in Spiking Neural Networks. *Springer 24th International Conference on Artificial Neural Networks, ICANN 2014, LNCS 2014*, LNCS 2014, (pp. 635-642).
- Gilra, A., & Gerstner, W. (2017). Predicting Non-Linear Dynamics by Stable Local Learning in a Recurrent Spiking Neural Network. *eLife 2017*, 6: e28295, doi: 10.7554/eLife.28295, 43 pages.
- Giorgilli, A., & Lazutkin, V.F. (2000). Some remarks on the problem of ergodicity of the standard map. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 272(5-6), 359-367.
- González, René & Vellasco, Marley & Figueiredo, Karla. (2019). Resource optimization for elective surgical procedures using quantum-inspired genetic algorithms. 777-786. 10.1145/3321707.3321786.
- Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning, *MIT Press*.
- Gorelick, P. B. (1997). Status of Risk Factors for Dementia Associated with Stroke. *Stroke*, 28(2), 459-463.
- Gorodetski, A. (2012). On Stochastic Sea of the Standard Map. *Communications in Mathematical Physics*, 309(1), 155-192.
- Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 212-19.

Grover, L. K. (1998). Quantum Search on Structured Problems. *Proceedings of the 1st NASA International Conference on Quantum Computation and Quantum Communications*.

Haga, T. & Fukai, T. (2019). Extended Temporal Association Memory by Modulations of Inhibitory Circuits. *Physical Review Letters*, 123 (7), doi: 10.1103/PhysRevLett.123.078101.

Hameroff, S., & Penrose, R. (2014). Consciousness in the universe: A review of the ‘Orch OR’ theory. *Physics of Life Reviews*. 11(1), 39-78.

Hamano, G., Lowe, A., & Cumin, D. (2017). Design of Spiking Neural Networks for Blood Pressure Prediction during General Anesthesia: Considerations for Optimizing Results. *Evolving Systems*, 8(3), 203-210.

Hampel, H., Frank, R., Broich, K., et al. (2010). Biomarkers for Alzheimer’s Disease: Academic, Industry and Regulatory Perspectives. *Nature Reviews Drug Discovery*, 9(7), 560–574.

Han, K., & Kim, J. (2000). Genetic Quantum Algorithm and its Application to combinatorial optimisation problem. *IEEE Proceeding of the 2000 Congress on Evolutionary Computation*, 2, (pp. 1354-1360).

Han, K. H., & Kim, J. H. (2002). Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimisation. *IEEE Transactions on Evolutionary Computation*, 6(6), 580-587.

Han, K., & Kim, J. (2004). Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, H-Epsilon Gate, and Two-Phase Scheme. *IEEE Transaction on Evolutionary Computation*, 8(2), 156-169.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Advances in Intelligent Computing*, 878-887.

He, H. D., Lu, W. Z., & Xue, Y. (2014). Prediction of Particulate Matter at Street Level Using Artificial Neural Networks Coupled with Chaotic Particle Swarm optimisation. *Journal of Building and Environment*, 78, 111-117.

Henderson, J. A., Gibson, T. A., & Wiles, J. (2015). Spike Event Based Learning in Neural Networks. *arXiv preprint arXiv: 1502.05777v1*.

Hirsch, M. W., Smale, S., & Devaney, R. L. (2004). Differential Equations, Dynamic Systems & An Introduction to Chaos. *Elsevier Academic Press*, 60, 2end edition, USA, P. A. Smith and S. Eilenberg, ISBN: 0-12-349703-5.

Hong, K. S., Khan M. J., & Hong, M. j. (2018). Feature Extraction and Classification Methods for Hybrid fNIRS-EEG Brain-Computer Interfaces. *Frontiers in Human Neuroscience*. 12(246), doi: 10.3389/fnhum.2018.00246.

Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Scientists*, 79, (pp. 2554-2558).

- Hu, J., Hou Z.-G., Chen, Y.-X., Kasabov, N., & Scott, N. (2014). EEG-Based Classification of Upper-Limb ADL Using SNN for Active Robotic Rehabilitation. *The 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, (pp. 409-114).
- Hu, Q., Liu, J., & Yu, D. (2008). Mixed Feature Selection Based on Granulation and Approximation. *Knowledge-Based Systems*, 21, 294-304.
- Hu, J., Tang, H., Tan, K. C., & Gee, S. B. (2015). A Spiking Neural Network Model for Associative Memory Using Temporal Codes. *Springer Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 1, (pp 561-572).
- Huang, G., Huang, G. B., Song, S., & You, K. (2015). Trend in Extreme Learning Machines: A review, *Journal of Neural Networks*, 61, 32-48.
- Huang, X., et al. (2018). Revealing Alzheimer's Disease Genes Spectrum in the Whole-Genome by Machine Learning. *BMC Neurology*, 18(5), 8 pages, doi: 10.1186/s12883-017-1010-3.
- Hunsberger, E., & Eliasmith, C. (2015). Spiking Deep Networks with LIF Neurons. *arXiv preprint arXiv: 1510.08829*.
- Hunter, R., Cobb, S. R., & Graham, B. (2008). Improving Associative Memory in a Network of Spiking Neurons. In: *Kurkova V, Neruda R, Koutnik J (ed.) Artificial Neural Networks - ICANN 2008, 18th International Conference*, (pp. 636-645), Prague, Czech Republic, September 3-6, 2008, Proceedings, Part II, Berlin, Germany, Springer-Verlag. 18th International Conference on Artificial Neural Networks –ICANN 2008, 3.9.2008 - 6.9.2008, Prague, Czech Republic.
- Indiveri, G., Corradi, F., & Qiao, N. (2015). Neuromorphic Architectures for Spiking Deep Neural Networks. *IEEE International Electronic Devices Meeting (IEDM)*, (pp. 421-424), Washington, DC, USA.
- Izhikevich, E. M. (2003). Simple Model of Spiking Neurons. *IEEE Transactions on Neural Networks*, 14(6), 1569-1572.
- Izhikevich, E.M. (2006). Polychronization: Computation with Spikes. *MIT Press Journal, Neural Computation*, 18, 245-282.
- Jankovic, M. V., Gajic, T., & Reljin, B. D. (2014). Applications of probabilistic model based on main quantum mechanics concepts. *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, 33–36.
- Jascur, M. (2013). Quantum Theory of Magnetism. Pavol Jozef Safarik University in Kosice, ISBN 978-80-8152-049-5.
- Kaiser, H. F. (1956). The Microtron, A Nuclear and Electronic Research Instrument. *IEEE Transaction on Nuclear Science*, 3(2), 17-26.

Kaplan, B.A., Lansner, A., Masson, G.S., & Perrinet, L.U. (2013). Anisotropic Connectivity Implements Motion-Based Prediction in a Spiking Neural Network. *Frontiers in Computational Neuroscience*, doi: 10.3389/fncom.2013.00112.

Karimov, A. I., Butusov, D. N., Rybin, V. G., & Karimov, T. I. (2017). The study of the modified Chirikov map. *IEEE International Conference on Soft Computing and Measurements (SCM)*, 341-344.

Kasabov, N. (2010). Neural Networks Letter: To Spike or Not to Spike: A Probabilistic Spiking Neuron Model. *Neural Networks*, 23(0893-6080), 16019.

Kasabov N. (2010). Integrative Probabilistic Evolving Spiking Neural Networks Utilising Quantum Inspired Evolutionary Algorithm: A Computational Framework. In: *Koronacki J., Raś Z.W., Wierzchoń S.T., Kacprzyk J. (eds) Advances in Machine Learning II. Studies in Computational Intelligence*, 263. Springer, Berlin, Heidelberg, doi: https://doi.org/10.1007/978-3-642-05179-1_19.

Kasabov, N. K., (2019), Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, *Springer Series on Bio- and Neurosystems*, 7, pages 742, doi: 10.1007/978-3-662-57715-8.

Kasabov, N., Dhoble, K., Nuntalid, N., & Indiveri, G. (2013). Dynamic Evolving Spiking Neural Networks for Online Spatio- and Spectro-Temporal Pattern Recognition. *Neural Networks*, 41, 188-201.

Kasabov, N., Feigin, V., Hou, Z.-G., Chen, Y., Liang, L., Krishnamurthi, R., Othman, M., & Parmar, P. (2014). Evolving Spiking Neural Networks for Personalised Modelling, Classification and Prediction of Spatio-temporal Patterns with a Case Study on Stroke. *Neurocomputing*, 134, 269-279.

Kasabov, N. (2014). NeuCube: A Spiking Neural Network Architecture for Mapping, Learning and Understanding of Spatio-temporal Brain Data. *Neural Networks*, 52, 62-76.

Kaveh, A., Sheikholeslami, R., Talataheri, S., & Keshvari-Ilkhichi, M. (2014). Chaotic Swarm of Particles: A New Method for Optimisation. *Journal of Advances in Engineering Software*, 67, 136-147.

Kazem, A., Sharifi, E., Hussain, F. K., & Saberi, M. (2013). Support Vector Regression with Chaotic Based Firefly Algorithm for Stock Market Price Forecasting. *Journal of Applied Soft Computing*, 13, 947-958.

Knight, B. W. (1972). Dynamics of Encoding in a Population of Neurons. *The Journal of General Physiology*, 59(6), 734-766.

Kob, W. & Schilling, R., (1989). KAM tori, chaotic motion and exactly integrable islands for a non-differentiable Hamiltonian system. *Journal of Physics A: Mathematical and General*, 22(13), 633-639.

Kobuchi, Y. (1991). State evaluation functions and Lyapunov functions for neural networks, *Neural Networks*, 4(4), 505-510.

Kostin, D. V., Shevedunov, V. I., Sobenin, N. P., & Trower, W. P. (1999). A Novel Racetrack Microtron Accelerating Structure. *IEEE Proceeding of the 1999 Particle Accelerator Conference*, (pp. 910-912).

Kristensen, L. B., Degroote, M., Wittek, P., Aspuru-Guzik, A., & Zinner, N. T. (2021). An artificial spiking quantum neuron. *Nature, npj Quantum Information*. <https://doi.org/10.1038/s41534-021-00381-7>.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proc. NIPS, MIT Press*.

Lancaster, J., L., Woldorff, M., G., Parsons, L., M., Liotti, M., Freitas, C., S., Rainey, L., Kochunov, P., V., Nickerson, D., Mikiten, S., A., & Fox, Peter T. (2000). Automated Talairach Atlas Labels for Functional Brain Mapping. *Human Brain Mapping*, 10, 120–131.

Langkvist, M., Karlsson, L., & Loutfi, A. (2014). A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modelling. *Journal of Pattern Recognition Letters*, 42, 11-24.

LeCun, Y., Bengio, Y.Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.

Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training Deep Spiking Neural Networks using Backpropagation. *arXiv preprint arXiv: 1608.08782v1*.

Lee, J. W., Park, J. B., & Choi, Y. H. (2010). Robust flight control system using neural networks: Adaptive dynamic surface design approach. *ICCAS 2010, Gyeonggi-do, Korea (South)*, 657-662, doi: 10.1109/ICCAS.2010.5669861.

Legenstein, R., Pecevski, D., & Maass, W. (2008). A Learning Theory for Reward-Modulated Spike-Timing-Dependent Plasticity with Application to Biofeedback. *PLoS Computational Biology*, 4(10), 27 pages, doi: 10.1371/journal.pcbi.1000180.

Leuengo, J., Garca, S., & Herrera, F. (2012). On the Choice of the best Imputation Methods for Missing Values Considering Three Groups of Classification Methods. *Knowledge and Information Systems*, 32, 77-108.

Lidbjork, P. (1994). Microtron. *CERN 94-01*, 2, 971-980.

Lu, Y., Zhou, J., Qin, H., Wang, Y., & Zhang, Y. (2011). Chaotic Differential Evolution Methods for Dynamic Economic Dispatch with Valve-Point Effects. *Journal of Engineering Application of Artificial Intelligence*, 24, 378-387.

Maass, W. (1994). On the Computational Complexity of Networks of Spiking Neurons. *Proceedings of the 7th International Conference on Neural Information Processing Systems*, Denver, Colorado, 183–190.

Magnasco, V. (2007). Valence Bond Theory and the Chemical Bond. *Elementary Methods of Molecular Quantum Mechanics*, 473-575.

Mastin, L., (2010). Types of Memory - The Human Memory. Human-Memory.Net. Retrieved from <http://www.human-memory.net/types.html>.

Matsuda, S. (2016). BPSpike: A backpropagation learning for all parameters in spiking neural networks with multiple layers and multiple spikes. *2016 International Joint Conference on Neural Networks (IJCNN)*, 293.

Meiss, J. (2008). Visual Exploration of Dynamics: The Standard Map, *Pramana. Journal of Physics*, 1-23.

Miguel, N., Simo, C., & Vieiro, A. (2013). From the Henon Conservative Map to the Chirikov Standard Map for Large Parameter Values. *Regular and Chaotic Dynamics*, 18(5), pp. 469-489.

Misra, P. K. (2012). Magnetic Ordering. *Physics of Condensed Matter*, 409-449.

Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., Thorpe, S. J., & Masquelier, T. (2018). Combining STDP and Reward-Modulated STDP in Deep Convolutional Spiking Neural Networks for Digit Recognition. *arXiv:1804.00227v1 [cs.CV]*.

Narayanan, A., & Moore, M. (1996). Quantum-Inspired Genetic Algorithms. *Proceeding of IEEE International Conference on Evolutionary Computation*, (pp. 61-66).

Nowotniak, R., & Kucharski, J. (2014). Higher-Order Quantum Inspired Genetic Algorithms. *arXiv:1407.0977v1[cs.NE]*.

O'Connor, P., & Welling, M. (2016). Deep Spiking Networks. *arXiv preprint arXiv:1602.08323v2*.

Ogorzalek, M. J. (1993). Taming Chaos – Part1: Synchronization. *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, 40(10), 693-699.

Othman, M., Kasabov, N., Tu, E., Feigin, V., Krishnamurthi, R., Hou, Z. G., Chen, Y., & Hu, J. (2014). Improved Predictive Personalised Modelling with the use of Spiking Neural Network System and a Case Study on Stroke Occurrences Data. *IEEE, 2014 International Joint Conference on Neural Networks (IJCNN)*, (pp.3197-3204), Beijing, China.

Pawlak, Z. & Skowron, A. (1994). Rough Membership Function. in: *R. E. Yeager, M. Fedrizzi and J. Kacprzyk (eds.), Advances in the Dempster-Schafer of Evidence*, Wiley, New York, 251-271.

Peleg, Y., Pnini, R., Zaarur, E., & Hecht, E. (2010). Quantum mechanics. McGraw-Hill Companies. Retrieved from <http://search.ebscohost.com.ezproxy.aut.ac.nz/login.aspx?direct=true&db=cat05020a&AN=aut.b14208520&site=eds-live>.

Penner, S., et al. (1981). The NBS-LASL CW Microtron. *IEEE Transaction on Nuclear Science*, 28(2), 1526-1530.

- Petro, B., Kasabov, N., & Kiss R. M. (2020). Selection and Optimisation of Temporal Spike Encoding Methods for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2), 358-370, doi: 10.1109/TNNLS.2019.2906158.
- Phillips, A. C. (2003). Introduction to Quantum Mechanics, *Wiley & Sons Ltd, The Manchester Physics Series, Department of Physics and Astronomy, University of Manchester*, pages 248.
- Pickel, V. M., & Segal, M. (Eds.). (2014). The Synapse: Structure and Function. *ProQuest Ebook Central* <https://ebookcentral.proquest.com>.
- Platel, M. D., Schliebs, S., & Kasabov, N. (2007). A Versatile Quantum-Inspired Evolutionary Algorithm. *IEEE Congress on Evolutionary Computation (ECE2007)*, (pp. 423-430).
- Platel, M. D., Schliebs, S., & Kasabov, N. (2009). Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. *IEEE Transactions on Evolutionary Computation*, 13(6), 1218-1232.
- Pluhacek, M., et al. (2013). On the Behaviour and Performance of Chaos Driven PSO Algorithm with Inertia Weight. *Journal of Computers and Mathematics with Applications*, 66, 122-134.
- Prucnal, M., & Polak, A. G. (2017). Effect of Feature Extraction on Automatic Sleep Stage Classification by Artificial Neural Network. *Metrology and Measurement Systems*, 24(2), 229-240.
- Ramirez, A., & Arbuckle, M. R. (2016). Synaptic Plasticity: The Role of Learning and Unlearning in Addiction and Beyond. *Biological psychiatry*, 80(9), e73–e75. <https://doi.org/10.1016/j.biopsych.2016.09.002>.
- Ramírez, J., Górriza, J. M., Ortizc, A., Martínez-Murcia, F. J., Segovia, F., Salas-Gonzaleza, D., Castillo-Barnesa, D., Illána, I. A., & Puntonetd, C. G. (2018). Ensemble of Random Forests One vs. Rest Classifiers for MCI and AD Prediction using ANOVA Cortical and Subcortical Feature Selection and Partial Least Squares. *Journal of Neuroscience Methods*, 302, 47–57.
- Reid, D., Hussain, A. J., & Tawfik, H. (2013). A Spiking Neural Network for Financial Prediction. *IEEE Proceedings of International Joint Conference on Neural Networks*, (pp. 3111-3118), Dallas, Texas, USA.
- Reid, D., Hussain, A. J., & Tawfik, H. (2014). Financial Time Series Prediction Using Spiking Neural Networks. *PLOS ONE*, 9, 1-13.
- Reid, D., Hussain, A. J., Tawfik, H., & Ghazali, R. (2014). Prediction of Physical Time Series Using Spiking Neural Networks. *Springer 10th International Conference ICIC 2014, LNAI 8589*, (pp. 816-824).
- Roesler, C., & Mobley, C. (2015). Physics of Absorption. Retrieved from http://www.oceanopticsbook.info/view/absorption/physics_of_absorption. *Ocean Optics web book*.

Rovatti, R., Setti, G., & Mazzini, G. (1998). Chaotic Complex Spreading Sequences for Asynchronous DS-CDMA-Part II: Some Theoretical Performance Bounds. *Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 45(4), 496-506.

Salgado Pereira, Cristiane & Dias, Douglas & Pacheco, Marco & Vellasco, Marley & Abs da Cruz, Andre & Hollmann, Estefane. (2020). Quantum-Inspired Genetic Programming Algorithm for the Crude Oil Scheduling of a Real-World Refinery. *IEEE Systems Journal*. PP. 1-12. 10.1109/JSYST.2020.2968039.

Santos Coelho, L. D. (2009). Reliability-Redundancy Optimisation by Means of a Chaotic Differential Evolution Approach. *Journal of Chaos, Solitons and Fractals*, 41, 594-602.

Santos Coelho, L. D., & Mariani, V. C. (2009). A Novel Chaotic Particle Swarm Optimisation Approach Using Henon Map and Implicit Filtering Local Search for Economic Load Dispatch. *Journal of Chaos, Solitons and Fractals*, 39, 510-518.

Santos Coelho L. D., & Rodrigues Coelho, A. A. (2009). Model-Free Adaptive Control Optimisation Using a Chaotic Particle Swarm Approach. *Journal of Chaos, Solitons and Fractals*, 41, 2001-2009.

Santos Coelho, L. D., Sauer, J. G., & Rudek, M. (2009). Differential Evolution Optimisation Combined with Chaotic Sequences for Image Contrast Enhancement. *Journal of Chaos, Solitons and Fractals*, 42, 522-529.

Schliebs, S. Michael Defoin Platel, Susan Worner and Nikola Kasabov, Integrated Feature and Parameter Optimization for Evolving Spiking Neural Networks: Exploring Heterogeneous Probabilistic Models, *Neural Networks*, 22, 623-632, 2009.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Journal of Neural Networks*, 61, 85-117.

Schrauwen B., & Campenhout, J. V. (2003). BSA, a Fast and Accurate Spike Train Encoding Scheme. *IEEE Proceedings of the International Joint Conference on Neural Networks*, (pp. 2825-2830), doi: 10.1109/IJCNN.2003.1224019.

Schrauwen, B., & Campenhout J. V. (2004). Improving spikeprop: Enhancements to an error-backpropagation rule for spiking neural networks. *Proceedings of the 15th ProRISC Workshop*, (pp. 301-05).

Sherwin, C. W. (1959). Introduction to Quantum Mechanics, *Holt, Rinehart and Winston, Inc. New York*, pages 199.

Silveira, L.R. & Tanscheit, Ricardo & Vellasco, Marley. (2016). Quantum Inspired Evolutionary Algorithm for Ordering Problems. *Expert Systems with Applications*. 67. 10.1016/j.eswa.2016.08.067.

Skowron A., & Rauszer C. (1992). The Discernibility Matrices and Functions in Information Systems. In: *Słowiński R. (eds) Intelligent Decision Support. Theory and Decision Library (Series D: System Theory, Knowledge Engineering and Problem Solving)*, 11, 311-362.

- Smith S. S. (2013). $\alpha 4\beta\delta$ GABAA receptors and tonic inhibitory current during adolescence: effects on mood and synaptic plasticity. *Frontiers in neural circuits*, 7, 135. <https://doi.org/10.3389/fncir.2013.00135>
- Sobenin, N. P., Melekhin, V. N., Karev, A. I., Shvedunov, V. I., & Trower, W. P. (1996). Rectangular Microtron Accelerating Structure. *IEEE Particle Accelerator Conference*, (pp.1827-1829).
- Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian Learning Through Spike-Timing-Dependent Synaptic Plasticity. *Journal of Nature Neuroscience*, 3(9), 919-926.
- Squire, L. R., (2004), Memory systems of the brain: a brief history and current perspective. *Neurobiol Learn Mem*, 82, 171–177.
- Squire, L. R., (2009), Memory and Brain Systems: 1969 –2009, *The Journal of Neuroscience*, 29(41), 12711–12716.
- Stern, M., & Shea-Brown, E. (2020). Network Dynamics Governed by Lyapunov Functions: From Memory to Classification. *Trends in Neurosciences*, 43(7), 453–455. <https://doi-org.ezproxy.aut.ac.nz/10.1016/j.tins.2020.04.002>.
- Stephan, A. H., Barres, B. A., & Stevens, B. (2012). The Complement System: An Unexpected Role in Synaptic Pruning During Development and Disease. *Annual Review of Neuroscience*, 35:1, 369-389.
- Stojanovski, T., & Kocarev, L. (2001). Chaos-Based Random Number Generator-Part I: Analysis. *Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 48(3), 281-288.
- Strehl, U. (2014). What learning theories can teach us in designing neurofeedback treatments, *Frontiers in Human Neuroscience*, 8:894. doi: 10.3389/fnhum.2014.00894.
- Strevens, M. (2006). Chaos Theory. In D. M. Borchert (ed.), *Encyclopedia of Philosophy*, second edition. Macmillan Reference USA, Detroit, 1-7.
- Strogatz, S. H. (2014). Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity). Westview Press, 2end Edition, ISBN-13: 978-0813349107.
- Sun, Y., Zeng, Y., & Zhang, T. (2020). Quantum Superposition Spiking Neural Network, arXiv:2010.12197v2 [cs.NE].
- Suzuki, W. A. (2005). Associative Learning and the Hippocampus. *Psychological Science Agenda*, American Psychological Association.
- Talairach, J., & Tournoux, P. (1988). Co-planar stereotaxic atlas of the human brain:3-dimensional proportional system—an approach to cerebral imaging. New York: Thieme Medical Publishers.

- Tang, K. T., Toennies, J. P., & Yiu, C. L. (1998). The generalized heitler-london theory for interatomic interaction and surface integral method for exchange energy. *International Reviews in Physical Chemistry*, 17(3), 363–406. <https://doi-org.ezproxy.aut.ac.nz/10.1080/014423598230090>.
- Tang, W., & Mao, K. Z. (2007). Feature Selection Algorithm for mixed data with both nominal and continuous features. *Pattern Recognition Letters*, 28, 563-571.
- Tang, X., Zhuang, L., Cai, J., & Li, C. (2010). Multi-Fault Classification Based on Support Vector Machine Trained by Chaos Particle Swarm Optimisation. *Journal of Knowledge-Based Systems*, 23, 486-490.
- Taylor, D., Scott, N., Kasabov, N., Capecciy, E., Tuz, E., Saywell, N., Chenx, Y., Hux, J., & Houx, Z. G. (2014). Feasibility of NeuCube SNN architecture for detecting motor execution and motor intention for use in BCI applications. *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, 3221-3225, doi: 10.1109/IJCNN.2014.6889936.
- Techapet, N. J-P., & Nana E. S. C. (2012). Concise Quantum Associative Memories with Nonlinear Search Algorithm. *Fortschritte der Physik, Progress of Physics*, 64(2-3), 250-268.
- Techapet, N. J-P., Nana, E. S. C., & Wofo, P. (2015). Quantum-inspired associative memory with Improved Distributed Queries. *International Journal of Theoretical Physics*, 52, 1787-1801.
- Techapet, N. J-P., & Nana, E. S. C. (2018). Quantum-inspired associative memory with Linear and non-Linear Algorithms for the Diagnosis of some Tropical Diseases. *Neural Networks*, 97, 1-10.
- The University of Queensland, Queensland Brain Institute. Retrieved from <https://qbi.uq.edu.au/brain-basics/memory/where-are-memories-stored>, 2020, The University of Queensland, Australia, Updated: 23 Jul 2018.
- Tu, E., Kasabov, N., Othman, M., Li Y., Worner, S., Yang, J., & Jia, Z. (2014). NeuCube for Spatio-temporal Data Predictive Modelling with a Case Study on Ecological Data. *International Joint Conference on Neural Networks, IJCNN 2014*, (pp. 638-645).
- Ventura, D., & Martinez, T. (2000). Quantum-inspired associative memory. *Information Sciences*, 124, 273-296.
- Vlachogiannis, J. G., & Lee, K. Y. (2008). Quantum-Inspired Evolutionary Algorithm for Real and Reactive Power Dispatch. *IEEE Transactions on Power Systems*, 23(4), 1627-1635.
- Watson, J. F., Ho, H., & Greger, I. H. (2017). Synaptic transmission and plasticity require AMPA receptor anchoring via its N-terminal domain. *eLife*, 6, e23024. <https://doi.org/10.7554/eLife.23024>
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature* 393(1), 440–442.

Werndl, C. (2010). The Formulation and Justification of Mathematical Definition Illustrated by Deterministic Chaos. *EPSA Philosophical Issues in the Sciences: Launch of the European Philosophy of Science Association*, Springer, ISBN: 978-90-481-3252-2, 2010, 1-13.

Whittington J. C. R. & Bogacz, R. (2019). Theories of Error Back-Propagation in the Brain. *Trends in Cognitive Sciences*, 23(3), 235-250, doi: <https://doi.org/10.1016/j.tics.2018.12.005>.

Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2018). Spatio-temporal Backpropagation for Training High-Performance Spiking Neural Networks. *Frontiers in Neuroscience*, 12, 737-744, doi: 10.3389/fnins.2018.00331.

Xiong, Q., Zhang, X., Wang, W. F., & Gu, Y. (2020). A Parallel Algorithm Framework for Feature Extraction of EEG Signals on MPI. *Computational and Mathematical Methods in Medicine*, <https://doi.org/10.1155/2020/9812019>.

Yan, H., Zhao, L., Hu, L., Wang, X., Wang, E., & Wang, J. (2013). Nonequilibrium landscape theory of neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 110(45), 18035-18035.

Yang, L., & Zhongjian, T. (2011). Prediction of Grain Yield Based on Spiking Neural Networks Model. *IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, (pp. 171-174).

Yang T. & Chua, L. O. (1997). Impulsive Stabilization for Control and Synchronization of Chaotic Systems: Theory and Application to Secure Communication. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 44(10), 976-988.

Yao, Y. Y. (1998). A Comparative Study of Fuzzy Sets and Rough Sets. *Information Science*, 109(1-4), 227-242.

Yerramalla, S., Fuller, E., Mladenovski, M., & Cukic, B. (2003). Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System. In: Huang ST., Herman T. (eds) *Self-Stabilizing Systems. SSS 2003. Lecture Notes in Computer Science*, 2704. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45032-7_6.

Yu, F. & Xu, X. (2014). A Short-Term Load Forecasting Model of Natural Gas Based on Optimised Genetic Algorithm and Improved BP Neural Network. *Journal of Applied Energy*, 134, 102-113.

Zhou, R., Wang, H., Wu, Q., & Shi, Y. (2012). Quantum Associative Neural Network with Nonlinear Search Algorithm. *International Journal of Theoretical Physics*, 51(3), 705-723.

Zhang, G. X. (2010). Quantum-Inspired Evolutionary Algorithms: A Survey and Empirical Study. *Journal of Heuristics*, 17, 303-351.

Zhang, G. X., Gheorghe, M., & Wu, C. Z. (2008). A Quantum-Inspired Evolutionary Algorithm Based on P Systems for Knapsack Problem. *IOS press, Fundamenta Informaticae*, 87, 93-116.

- Zhang, G., Li, N., Jin, W., & Hu, L. (2006). Novel Quantum Genetic Algorithm and Its Applications. *Frontiers of Electronic Engineering*, 31-36.
- Zhang, G. X., Liu, C. X., & Rong, H. N. (2010). Analysing Radar Emitter Signals with Membrane Algorithm. *Mathematical and Computer Modelling*, 52, 1997-2010.
- Zhang, G. X., & Rong, H. N. (2007). Real-Observation Quantum-Inspired Evolutionary Algorithm for a Class of Numerical Optimisation problems. In *Lecture Notes in Computer Science*, 4490, 989-996.
- Zhang, M., Wang, J., Zhang, Z., Belatreche, A., Wu, J., Chua, Y., Qu, H., & Li, H. (2020). Spike-Timing-Dependent Back Propagation in Deep Spiking Neural Networks. *arXiv:2003.11837 [cs.NE]*.
- Zhang, X., Mei, C., Chen, D., & Li, J. (2016). Feature Selection in Mixed Data: A Method using a Novel Fuzzy Rough Set-Based Information Entropy. *Journal of Pattern Recognition*, 56, 1-15.
- Zhou, T., & Wachs, J. P. (2017). Early Turn-Taking Prediction with Spiking Neural Networks for Human Robot Collaboration. *arXiv:1709.09276v1 [cs.RO]*.

Appendix A

Fuzzy Rough Set Feature Selection

The performance of the fuzzy rough set feature selection method proposed in chapter 7 is evaluated by applying it to a real-world dataset of Dementia patients which consist of 84 nominal and continuous features for 67 patients. These patients belong to three different dementia classes: Alzheimer Disease (8 patients), Vascular Dementia (14 patients) and, Mixed Dementia (45 patients). Although the main purpose of this research is to perform feature selection on the Dementia dataset, four additional datasets from the UCI Repository of machine learning databases are also used to provide a fair comparison. The description of these datasets is summarized in table 1. Then the results of the fuzzy rough set feature selection algorithm are compared with three other feature selection algorithms.

Table 8-1: The characteristics of the Datasets used for experiments

Dataset	Number of Attributes	Number of Instances	Missing Values
Dementia	85	67	Yes
Insurance Company (COIL2000)	86	4000	No
Heart Disease	75	303	Yes
Wine	13	176	No
Breast Cancer Wisconsin (Diagnostic)	32	569	No

K-nearest Neighbors for Missing Data Imputation

Dementia and Heart disease datasets have some missing values. K-nearest neighbours are used as a data imputation technique to replace these missing values by defining a set of K-nearest neighbours for each sample or individual and replacing the missing data for a given variable by averaging (non-missing) values of its neighbours.

SMOTE Oversampling Technique

Another issue with the dementia data set is that the class members distribution of the data is highly imbalanced. To handle this problem, the Synthetic Minority Over-Sampling Technique (SMOTE) is used to generate artificial data based on the similarities between the feature spaces of the existing minority class samples. The algorithm generates some random points in between the two specified vector points and thus more generalises the minority class decision region.

In these experiments, KNN, Decision Tree, Multilayer Perceptron, and Random Forest classifiers' accuracy are used to measure the quality of the feature subsets.

In table Table 8-2, a triangle fuzzy neighbourhood with neighbour radius N is shown. The neighbour radius in fuzzy rough set is related to granularity and determines the number of training samples in the classification boundary region. As it can be seen, the greater amount of neighbourhood radiuses results in a lower classification error.

Table 8-2: Dementia dataset feature selection results with different Neighborhood radius using fuzzy rough set-based information entropy

FRFS		Classifiers Error		
Neighbourhood radius	KNN	Decision Tree	Multilayer	Random
			Perceptron	Forest
			MLP	
N=0.25	0	0	0.4104	0.2278
N=0.5	0	0	0.3603	0.2258

N=0.75	0	0	0.3723	0.2242
N=1	0	0	0.3104	0.2036

Table 8-3: The classifiers' accuracy comparison for Dementia dataset using four feature selection algorithms

Methods/Classifiers	KNN	Decision Tree	Multilayer Perceptron (MLP)	Random Forest
Error				
GAFS	0.4708	0.0846	0.3612	0.3538
DEFS	0.4670	0.2149	0.3821	0.3112
PSOFS	0.5769	0.4876	0.5606	0.5182
FRFS	0	0	0.3104	0.2036

In Table 8-4, the results of the fuzzy rough set-based information entropy algorithm are compared with three feature selection algorithms which are wrapper methods based on genetic algorithm, deferential algorithm, and particle swarm optimization technique, respectively. The results proved the acceptable performance of the fuzzy rough set-based information entropy algorithm. However, Dementia small sample space here is a drawback and affects the results. As you can see, although the fuzzy rough set-based information entropy algorithm archived 100% accuracy for KNN and decision tree classifiers, for multilayer perceptron and random forest which are highly dependent on the training samples it couldn't provide a good result. Therefore, for more investigation, four additional datasets from the UCI Repository of machine learning databases are used to test the classification accuracy.

Table 8-4: Dementia dataset feature selection results using SMOTE data sampling method

FRFS	Classifiers				
	Parameter settings	KNN	Decision Tree	Multilayer Perceptron (MLP)	Random Forest
SMOTE	N=1	0	0	0.0537	0.1472
Imbalanced	N=1	0	0	0.3104	0.2036

Another issue with the Dementia dataset is its imbalanced class members. In table 4, the result of applying SMOTE technique on Dementia datasets is compared with imbalanced Dementia datasets. The results show improvement in classification accuracy by decreasing classifiers' errors.

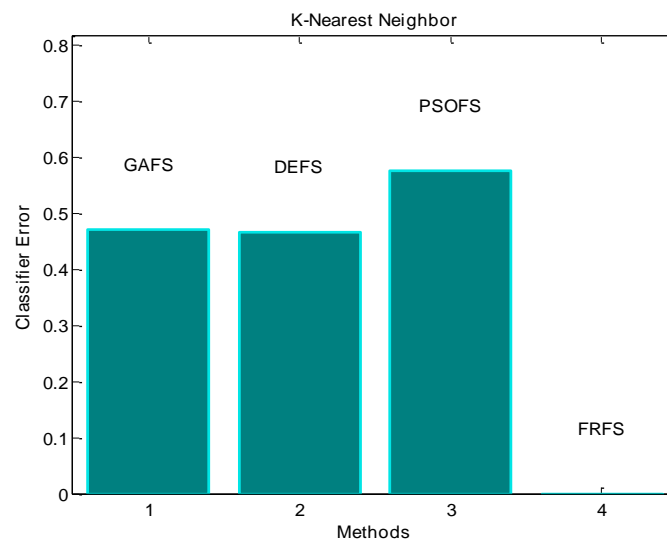


Figure 8-3: K-Nearest Neighbor classification errors for Dementia dataset

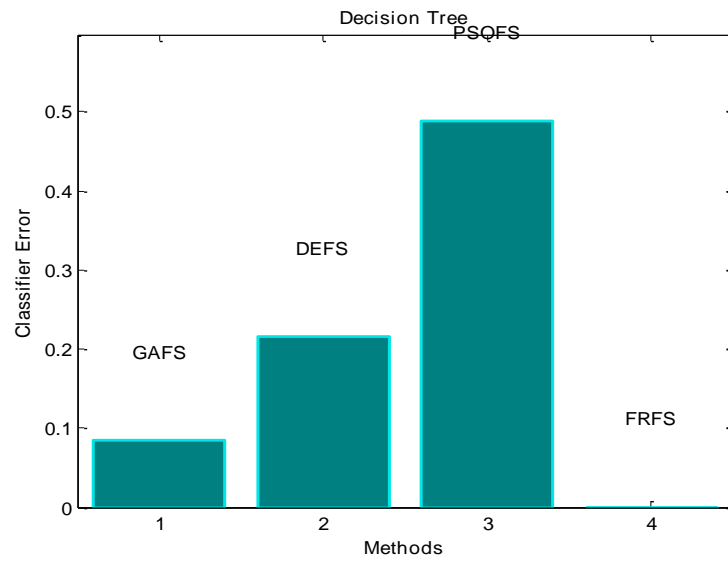


Figure 8-4: Decision Tree classification errors for Dementia dataset

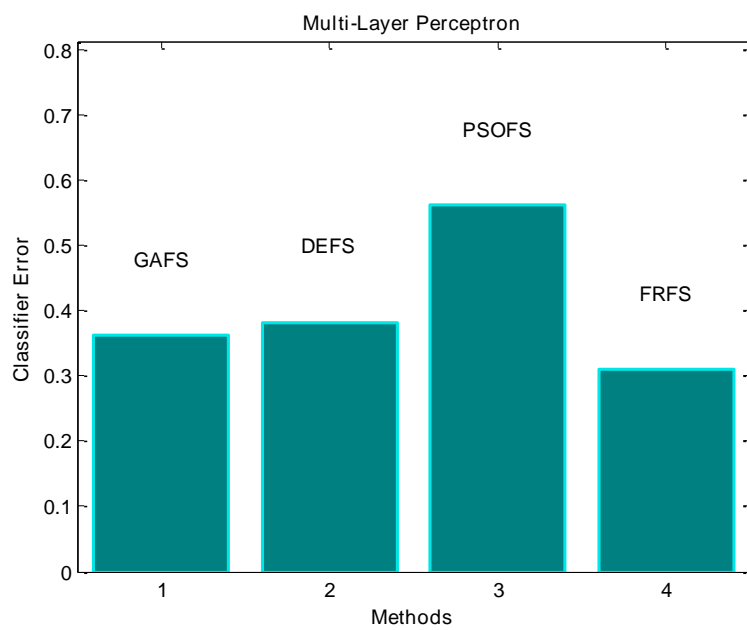


Figure 8-5: Multilayer perceptron classification errors for Dementia dataset

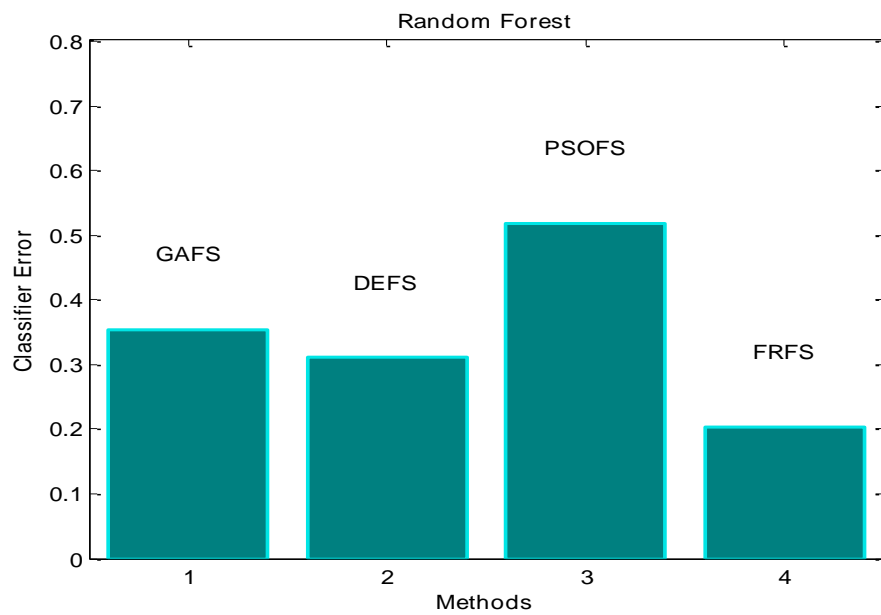


Figure 8-6: Random Forest classification errors for Dementia dataset

Table 8-5: Feature selection results for UCI datasets

Dataset	Methods/Classifiers Error	KNN	Decision Tree	Multilayer Perceptron (MLP)	Random Forest
Insurance Company (COIL2000)					
	DEFS	0.0375	0.0375	0.0696	0.0617
	PSOFS	0.1107	0.1107	0.0728	0.1107
	FRFS	0.0252	0.0252	0.0621	0.0584
	GAFS	0.058	0.058	0.0781	0.0734
Heart Disease					
	GAFS	0.0242	0.0242	1.018	0.2814
	DEFS	0	0	0.9539	0.2309
	PSOFS	0.0455	0.0455	0.5789	0.2545
	FRFS	0	0	0.4950	0.1142
Wine					
	GAFS	0	0	0.1198	0.029
	DEFS	0	0	0.1077	0.031
	PSOFS	0	0	0.0699	0.0363
	FRFS	0	0	0.0715	0.045
Breast Cancer Wisconsin (Diagnostic)					
	GAFS	0	0	0.1066	0.0325
	DEFS	0	0	0.1128	0.0256
	PSOFS	0	0	0.1616	0.051
	FRFS	0	0	0.1014	0.0271
	GAFS	0	0	0.1066	0.0325

In Table 8-5, four datasets from the UCI Repository of machine learning databases are used to test the performance of the selected algorithms. As it can be seen, the fuzzy rough set-based information entropy algorithm in three datasets performs better compared to the other three algorithms. However, in the Wine dataset, the wrapper method using the differential evolution algorithm is more successful. Although the performance of the fuzzy rough set-based information entropy algorithm is good in these experiments, the computation time is a critical issue, especially when the dataset dimension increases this algorithm performance is very poor compared to the other three wrapper methods. Among the selected methods, the differential evolution-based wrapper is the fastest one and as it can be seen the result produced by this algorithm are also acceptable. The disadvantage of evolutionary-based wrapper methods is parameter tuning and defining a fixed set of features for selection. In general, the combination of fuzzy rough set precision and evolutionary wrapper speed seems to be a promising approach to benefit the advantages of both techniques.

Conclusion

In this report, the impact of feature selection methods on the heterogeneous real-world datasets is studied. Among the proposed methods in the literature, the fuzzy rough set-based information entropy algorithm is chosen for performing feature selection because of its ability to handle mixed data and providing more precise results in terms of classification. The results of applying the algorithm on four real-world datasets were compared with three other feature selection algorithms from the literature. Although the selected filter-wrapper method outperforms three other feature selection methods in all the experiments, its main and important disadvantage is time computation cost, especially for a large data set. Moreover, in some cases the fuzzy lower space approximation exceeded the upper bounds which leads to producing incorrect feature subsets. In future work to overcome these issues, combination of this method with quantum

inspired evolutionary algorithm is recommended to both decrease computation time complexity and provide a better approximation in fuzzy lower space.