Spatial-Temporal Data Modelling and Processing for Personalised Decision Support

Muhaini Othman

A thesis submitted to Auckland University of Technology
in fulfilment of the requirement for
the degree of Doctor of Philosophy

Supervisors:
Prof. Nikola Kasabov
Assoc. Prof. Russel Pears
Assoc. Prof. Dave Parry

Consultants:
Dr. Rita Krishnamurthi
Prof. Valery Feigin
Assoc. Prof. Sue Worner

prepared at The Knowledge Engineering and Discovery Research Institute, KEDRI

submitted on September, 2015

School of Computer and Mathematical Sciences

Abstract

Introduction

Capturing the nature of spatio/spectro-temporal data (SSTD) is not an easy task nor is understanding the relationships between the different data dimensions such as between temporal and spatial, temporal and static, and between temporal variables themselves. In the past it has been normal to separate the SSTD dimensions and only take one dimension of the data and convert it into a static representation and model from there. While other dimensions are either ignored or modelled separately. Although this practice has had significant outcomes, the relationships between data dimensions and the meaning of that relationship defined be the data is lost and can result in inaccurate solutions. Any relationship between the static and dynamic or temporal data has been under analysed, if analysed at all, dependent upon the field of study.

Purpose of the research

The purpose of this research is to undertake the modelling of dynamic data without losing any of the temporal relationships, and to be able to predict likelihood of outcome as far in advance of actual occurrence as possible. To this end a novel computational architecture for personalised (individualised) modelling of SSTD based on spiking neural network methods (PMeSNNr), with a three dimensional visualisation of relationships between variables is proposed. The main architecture consists of a spike time encoding module; a recurrent or evolving 3D spiking neural network reservoir (eSNNr); an output module for either classification or prediction based around another evolving spiking neural network; and a parameter optimisation module. In brief, the architecture is able to transfer spatio-temporal data patterns from a multidimensional input stream into internal patterns in the eSNNr. These patterns are then analysed to produce a personalised model for either classification or prediction dependent on the specific needs of the situation.

Method

The architecture described above was constructed using MatLab in several individual modules linked together to form NeuCube (M1). This is the first iteration of the NeuCube architecture and as such remains relatively basic in its operations. The value of results obtained have also been analysed against the backdrop of the limitations of existing global and personalised methods with respect to SSTD. The following list briefly outlines the constituent components of the current version of NeuCube (M1) that was developed by our team.

- An encoding method employing Address Event Representation (AER) algorithm.
- A recurrent 3D SNN reservoir based on the Liquid-State Machine (LSM) concept and implementation of Spike Time Dependent Plasticity (STDP) as a learning rule.
- Innovative input variables mapping techniques utilizing Factor Graph Matching (FGM) algorithm.
- A predictive personalised modelling method for early event prediction.
- Various selections of evolving spiking neural network classifiers including a novel extended dynamic evolving spiking neural network method for multi-NN classification and regression problems called deSNNs_wkNN.
- A grid-search optimisation module and visualisation of the spiking network activities specifically on a group and personalised level.

This methodology has been applied to two real world case studies. Firstly, it has been applied to data for the prediction of stroke occurrences on an individual basis. This data consists of static variables (personal and geographic), and dynamic variables (climate, pollution and geomagnetic daily readings). Secondly, it has been applied to ecological data on aphid pest abundance prediction. The aphid data consists of only dynamic climate and geomagnetic variables. Two main objectives for this research when judging outcomes of the modelling are accurate prediction and to have this at the earliest possible time point. These two objectives are applied

to both case studies. Decisions of accuracy and dependability of the prediction are dependent upon the data available and the desired precision of the prediction.

Product

This study has found a number of interesting results.

- Firstly that using spiking neural networks for personalised modelling is more suitable for analysing and modelling SSTD dynamically compared with conventional machine learning methods that use global modelling, thus verifying the validity of this approach and that this methodology has also achieved a better results in terms of prediction accuracy.

- Secondly, using this approach early event prediction is possible where the time length of the training data (samples, collected in the past) and the test data (samples used for prediction) can be differentiate. Early event prediction is very crucial when solving important ecological and social tasks and disease risk prediction described by temporal-and/or spatial-temporal data, such as stroke risk prediction, pest population burst prevention, natural disaster warning, and financial crisis prediction.

- Thirdly, that these methods take all features without the need to filter noise and still produce good results.

- Fourthly, the innovative input variables mapping techniques enable dynamics mapping of SSTD variables and assist in revealing unknown spatio-temporal patterns and its associations.

- Lastly, the visualisation of spiking network activities enables deep network learning of the spiking patterns. This assists us in understanding the spiking neurons connection and relationships. Furthermore this visualisation reveals new knowledge about the SSTD that deserves to be investigated further.

Conclusions

The implications of these findings are not insignificant in terms of health care management and environmental control. As the case studies utilised here represent

vastly different application fields, it reveals more of the potential and usefulness of NeuCube for modelling data in an integrated manner. This in turn can identify previously unknown (or less understood) interactions thus both increasing the level of reliance that can be placed on the model created, and enhancing our human understanding of the complexities of the world around us without the need for over simplification. The visualisation of the cube inside NeuCube enables the researcher to gain valuable insight into not just the connectedness of variables but how this change dynamically as new data is presented. A simulation of what the real situation is more likely to be like in its construction, connection and the nature of the interaction between variables, i.e. does the current neuron promote the next neuron or inhibit it. The findings were published in five (5) papers and two (2) more have been recently submitted.

Attestation of Authorship

"I hereby declare that the submission is my own 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 extend has been submitted for the award of any other degree or diploma of a university or other institution of higher learning."

Auckland, 2015

Signature

Muhaini Othman

Name

15/09/2015

Date

Acknowledgement

Alhamdullillah to Allah for giving me strength and perseverance to finish the journey that started almost 4 years ago. There are so many people that have helped me and I am very thankful to have been blessed to meet such good people. First of all, I would like to express my deepest gratitude to my supervisor, Professor Nikola Kasabov, whose guidance and support from the very beginning to the final stage of my journey has helped me learn and deepen my understanding of the research area. Nikola, who has the attitude and the substance of a genius; has continually and convincingly conveyed a spirit of adventure in regard to research. He is a great supervisor with a pleasant personality, enthusiasm and wisdom.

I would also like to thank Dr. Yingjie Hu, who although supervised me briefly, advised and encouraged me and gave me strength to continue with my work. My thanks also go to Associate Professor Russel Pears for his insight and support in my research. I thank Associate Professor Dave Parry for giving me advice in the area of ontology.

I am heartily thankful to Dr. Rita Krishnamurthi and Professor Valery Feigin from The National Institute for Stroke and Applied Neurosciences (NISAN), for their advice, knowledge sharing and cooperation in my research. Without their case study, this research would be incomplete. My sincere thanks go to Associate Professor Sue Worner from Lincoln University for her advice.

During my study, I have had the opportunity to learn and work with a lot of people in the Knowledge Engineering and Discovery Research Institute (KEDRI). I would like to express my gratitude to the past and present members of KEDRI for their support and encouragement that had eased my PhD journey tremendously. Dr. Yixiong Chen and Dr. Jin Hu from Chinese Academy of Science, along with Dr. Enmei Tu from Shanghai Jiao Tong University deserves a special acknowledgement for their advice, guidance and for providing technical support, especially in system design and development during their brief stay at KEDRI. Without their genius mind, this research would be incomplete. Their kindness and friendship will forever be etched in my memory. I thank Vivienne Breen for her friendship and assistance in proof reading my writing and always being there for helpful discussions, especially

in understanding mathematical algorithms. My thanks goes to Reggio Hartono who helped in providing technical support whenever I needed it. My gratitude also goes to Dr. Yuxiao Li for her assistance in the case study of aphid pest abundance prediction. Thanks also to all the staff of KEDRI, Nathan Scott, Neelava Sengupta, Elisa Capecci, Maryam Gholami Doborjeh and Fahad Alvi. Not to forget the past staff of KEDRI, I thank Dr. Stefan Schliebs, Dr. Ammar Mohemmed, Dr. Haza Nuzly, Dr. Linda Liang, Dr. Kshitij Dhoble and Dr. Nuttapod Nuntalid for their support. I also like to thanks Nurdiana Nordin and Norhanifah Murli for their utmost sincere friendship and support.

To Joyce D'Mello, I am totally grateful for her assistance and support. She is always been there to advise and encourage me to pursue my dream and finish my study. She is the backbone of KEDRI. I am grateful for her friendship that will forever have a special place in my heart.

Thanks also to the Ministry of Education of Malaysia and University Tun Hussein Onn Malaysia (UTHM) for the financial support through the SLAB/SLAI Scholarship.

On a personal level, I would like to express my gratitude to my loving husband Kardi Sebli for his support, patience and encouragement. Without him by my side, I would not be able to finish this journey. To my sons Haziq, Syafiq and Taufiq, thanks for being such wonderful and supportive sons. You are my pillar of strength and I am doing this for you. My thanks also go to my beloved parents Othman Ismail and Sarah Awang who definitely cannot be thanked enough. Thank you for your support and prayers.

Lastly, I would like to offer my sincere appreciation to all of those who have helped me in any respect during my PhD journey.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

NO_2	Nitrogen Dioxide
O_3	Ozone
SO_2	Sulfur Dioxide
kNN	k-Nearest Neighbour
AER	Address Event Representation
Alate	Winged aphid
Anholocyclic	Parthenogenetics - reproduction without fertilization
ANN	Artificial Neural Network
Apterae	Non-winged aphid
ARCOS	Auckland Regional Community Stroke
BSA	Ben's Spike Algorithm
CNNs	Convolutional Neural Networks
CNS	Central Nervous System
CSVM	Cluster Support Vector Machine
DBNs	Deep Belief Networks
DENFIS	Dynamic Evolving Neural-Fuzzy Inference System
deSNN	Dynamic Evolving SNN
deSNNs_wkNN	A novel algorithm for classification and regression
ECF	Evolving Classification Function
ECOS	Evolving Connectionist System

EEG	Electroencephalography - a test that measures and records the electrical activity of the brain
eeSNN	Extended Evolving SNN
EFuNN	Evolving Fuzzy Neural Network
eSNN	Evolving Spiking Neural Network
FGM	Factor Graph Matching
FMRI	Functional magnetic resonance imaging - is a functional neuroimaging procedure using MRI technology that measures brain activity by detecting associated changes in blood flow, page 58
GRN	Gene Regulatory Network
HMM	Hidden Markov Models
Holocycly	Combination of sexual and asexual reproduction
HSA	Hough Spiker Algorithm
IMPM	Integrated Method for Personalised Modelling
LibSVM	A library for SVM
LIFM	Leaky Integrate-and-Fire Models
LOOCV	Leave-One-Out Cross Validation
LSM	Liquid-State Machine
LSSVM	Least Square Support Vector Machine
PMeSNNr	An evolving personalised modelling and spiking neural network framework and system
POC	Population Rank Order Coding
PSP	Post-synaptic Potential

QEA	Genetic Algorithm and Quantum-Inspired Evolutionary Algorithm
reSNN	Recurrent Evolving SNN
RO	Rank Order
ROC	Rank Order Coding
SNN	Spiking Neural Networks
SPAN	Spike Pattern Association Neuron
SRM	Spike Response Models
SSTD	Spectro, spatio-temporal data
SSVM	Smooth Support Vector Machine
STDP	Spike-Time Dependent Plasticity
STOS	Spatio-Temporal Ontology-based System
SVM	Support Vector Machine
TIA	Transient Ischaemic Attack
TWNFI	Transductive Neural Fuzzy Inference System with Weighted Data Normalization
TWRBF	Transductive RBF Neural Network with Weighted Data Normalization
wkNN	Weighted k-Nearest Neighbour
ww$k$NN	Weighted-weighted k-Nearest Neighbour
YDV	Yellow Dwarf Viruses

Chapter 1

INTRODUCTION

"All our knowledge has its origins in our perceptions."

- Leonardo da Vinci

1.1 Background

Spectro, spatio-temporal data (SSTD) is collected daily in many domains and is challenging to analyze because there are spatial and temporal connections amongst the data that need to be addressed accordingly. In them reside hidden patterns and new undiscovered knowledge that may solve numerous problems. Processing SSTD increases the data mining task complexity because it includes both temporal and spatial dimensions [Andrienko 2006].

In the domain area of bioinformatics, the concerns of manipulating SSTD to represent knowledge is crucial because it could lead to the notion of improving and saving lives either for humans, animals or the environment. In health related problems such as predicting stroke and heart attack occurrences, the analysis of SSTD will help in predicting the risk of these diseases by learning the temporal relations in the data for prevention purposes.

Analyzing SSTD related to ecological problems could help in restoration of the ecological balance that is sometimes disturbed or changed due to environmental factors. In the geological domain, SSTD pattern learning could assist in disaster management and may save lives.

1.2 Motivation

The development of personalised decision support systems has the potential to be the tool for better understanding health related problems like chronic disease including stroke, cardiovascular disease, cancer and countless unsolved medical problems. For instance, health related problems like chronic diseases are the major cause of death in almost all countries and it is projected that 41 million people will die of a chronic disease by 2015 unless urgent action is taken [Organization 2005]. Various initiatives have been taken to control the progression of symptoms in chronic disease patients such as clinical prevention using combination of drug therapy and calculation of a person's risk by referring to an existing risk chart which takes into account several risk factors. Additional initiatives involve the use of statistical methods to generate a survival model and to investigate several risk factors associated with chronic disease, such as the Cox Proportional Hazards Model [Lumley 2002], [Wolf 1991], [Yusuf 1998]. There are also several machine learning applications that used global models for prediction of a person's risk or the outcome of a certain diseases [Khosla 2010], [Das 2003], [Anderson 2006], [Levey 1999]. According to [Shabo 2007] there is evidence that prediction and treatment based on global models are only effective for some patients (about 70% average) leaving the remaining 30% of patients without proper treatment which could worsen their condition and possibly lead to their death. A global model is derived from all available data for the target and then applied to any new patient anywhere at any time. While it may give 70% to 80% average accuracy over the whole population, it still may not be suitable for many individuals [Kasabov 2010b]. Hence, using global models for prediction of a person‘s risk is inadequate, based on the assumption that every person or individual has their own unique characteristics.

Personal human health is defined by many factors such as the food they eat, their lifestyle, life stage, ethnic origin, previous growth and development, gender, environment influences, genetic differences, allergies, diseases and many other important factors [Lange 2007], including information regarding space (such as region and distance) and temporal constraint (for a period of time before the event) and relations between them. An example of stroke related studies, a simplified framework of the causal relations between climate-related factors and stroke was developed to

clarify the relations between environmental factors, lifestyle and a clinical risk factor with stroke occurrences.

Consequently, the emerging approach utilized to solve the problem is personalised modelling, where a model is created for every single new input vector of the problem space based on its nearest neighbours using a transductive reasoning approach [Kasabov 2007a]. However, there are very few efficient methods for the analysis of such complex data and discovery of complex spatio-temporal patterns, especially for on-line and real time applications.

1.3 Research Objectives, Research Questions and Hypothesis

Global modelling applied in most conventional machine learning methods has proven its effectiveness in the past, however it has a limited capability in producing models that fit each person or each case in the problem space since global modelling takes all available data in a problem space and produce a single general function [Kasabov 2007b]. The produced model is applied to a new individual regardless of their unique personal features. Common global modelling algorithms include Support Vector Machine (SVM) [Vapnik 1963] and Multilayer Perceptron (MLP) [Hornik 1989]. Therefore, in the case of stroke or any medical condition, personalized modelling methods are preferred for the reason that they can produce a model for each individual based on their personal features.

In numerous incidents, unforeseen events occur when triggered by the cascading effect of specific spatio, spectro temporal pattern interaction amongst multiple features over a period of time such as in the case of stroke [Feigin 1997], [Low 2006], ecological problems [Lankin 2001], geological disaster, financial crisis and many more. Such events can be avoided or the aftermath minimized if the risk is predicted early enough. However classical personalized modelling methods such as k-Nearest Neighbour (kNN) [Fix 1951] and weighted k-NN (wkNN) [Dudani 1976] are only suitable when classifying static vector based data, not SSTD.

The concept of spiking neural networks (SNN) has been considered as an emerging computational technique for the analysis of spatio-temporal datasets. This is

because SNN has the potential to represent and integrate different aspects of information dimensionality such as time and space; and has the ability to deal with large volumes of data using trains of spikes [Kasabov 2013]. SNN models such as Spike Response Models (SRM) [Gerstner 1995], Leaky Integrate-and-Fire Models (LIFM) [Gerstner 2002], Evolving Spiking Neural Network (eSNN) [Wysoski 2006] and Izhikevich models [Izhikevich 2004] have been successfully utilized in several classification tasks. They process input data streams as a sequence of static data vectors, ignoring the potential of SNN to simultaneously consider space and time dimensions in the input patterns. It can be viewed that SNN has more potential and is more suitable for SSTD pattern recognition utilizing emerging new methods such as reservoir computing [Maass 2002], Probabilistic Spiking Neuron Model [Kasabov 2010a], Extended Evolving SNN (eeSNN) [Hamed 2011], Recurrent Evolving SNN (reSNN) [Schliebs 2010], Spike Pattern Association Neuron (SPAN) [Mohemmed 2011] and Dynamic Evolving SNN (deSNN) [Dhoble 2012].

The main goal of this research is to develop a novel framework of an information method and system to analyse SSTD for personalised knowledge interpretation and prognosis. The main objective is to develop a generic modelling environment to analyse SSTD (medical, brain, financial, geological or ecological data, etc.) using personalised modelling and spiking neural network methods. Accordingly, the personalised modelling method called the Integrated Method for Personalised Modelling (IMPM) introduced by [Kasabov 2010b] will be incorporated into the system. The proposed framework will be applied to case studies related to stroke occurrences and ecological problems.

1.3.1 Research Objectives

Based from the above considerations, the research will achieve the following objectives:

1. To review the literature concerning how personalised modelling based on spiking neural networks method can best predict possible outcomes for a new person/event using historical SSTD.

2. To design a framework that can analyse and learn from SSTD and produce a

model that facilitates new knowledge discovery and provides better decision support.

3. To develop software systems that analyse, learn and visualise the pattern residing in SSTD.

4. To verify the proposed method and system for personalised decision support utilising case studies related to a chronic disease and an ecological problem.

1.3.2 Research Questions

The main research question here is:

Can personalised modelling based on spiking neural networks methods be developed to learn SSTD and produce a better personalised knowledge representation and risk prognosis for a person/event?

More specifically, several sub questions can be derived from this:

1. How to select an optimal set of features, neighbourhood, model and parameters for SSTD using spiking neural network methods?

2. How to encode the real value continuous SSTD into a train of spikes?

3. How to develop a recurrent 3D spiking neural networks reservoir for learning the continuous train of spikes?

4. How to utilise spiking neural networks modeling for improved classification accuracy without filtering any noise?

5. How to visualise complex SSTD feature correlation and interaction patterns for better interpretation of knowledge?

6. How to obtain the earliest time point for best prediction of the risk of an event occurring in the future for an individual?

7. How to improve the spiking neural networks method for regression problems?

1.3.3 Hypothesis

We hypothesise that the new method for a given complex problem,

1. utilising an individualised (personalised) modelling approach, where an individual model is created for every new individual, will be more accurate than a global modelling approach, where a single model is derived from all existing data to predict at earliest time a future event can be accurately predicted for any individual regardless of their specific static variable values.

2. that analysing all data collectively without data pre-processing or filtering proves that NeuCube is robust to noise.

3. the visualisation of interaction patterns amongst the features will assist in the learning process. The network of connections created during the learning process can be visualised and the relationship between features can be comprehended through the understanding of changes in the connection weights of neurons.

1.4 Thesis Structure

- Part 1 - Literature Review
  - Chapter 2 outlines the fundamentals of data modelling and pattern recognition approaches, including comparison between inductive modeling and transductive modeling approaches. This is followed by a more detailed discussion of global, local and personalized modeling approaches including conventional methods related to these approaches.
  - Chapter 3 introduces the Spiking Neural Networks as the new paradigm to process SSTD. Similarity between biological neurons and artificial neurons is reviewed. This chapter also outlines a brief history of SNN and its components including neuron models, data encoding, learning algorithms, working memories, reservoir computing and is followed by a review of several types of new SNN model and applications for spatio-temporal pattern recognition such as eSNN, eeSNN, reSNN and deSNN. This chapter also reviews a new paradigm of integrated system for brain data modelling.

- Part 2 - Proposed Novel PMeSNNr for SSTD and Applications
  - Chapter 4 discusses the motivation behind the development of this novel evolving personalised modelling and spiking neural network framework and system (PMeSNNr). Each component of the framework will be outlined; the encoding module, the unsupervised learning module, the supervised learning module and optimization module. New method that combines deSNNs with the wkNN method for Multi-NN classification and regression are proposed in this chapter.
  - Chapter 5 discusses the implementation of the PMeSNNr framework called NeuCube M1 and demonstrates the system's capability for predictive modelling; and added functionality to assist in deep learning and knowledge discoveries.
  - Chapter 6 reviews on the stroke disease including modifiable factors and external factors that influence the stroke occurrences. This chapter will also review previous studies regarding the influence of environmental factors that may cause brain stroke in humans. For application purposes, the New Zealand stroke occurrences case study will be used to evaluate the feasibility of the PMeSNNr in analysing and modelling real-value SSTD. This proposed method is used to do predictive personalised modelling for stroke risk prediction using temporal environmental data. The experimental study aims to produce an individual model for each subject and obtain the earliest time point to best predict the risk of a stroke event occurring in the future for an individual. Several groups of individuals are chosen according to season and personal information. Comparative experiments with conventional machine learning methods are also carried out. Discovery on new personalised knowledge will be further discussed based on visualisation generated during the modelling process.
  - Chapter 7 reviews the ecological problem relating to aphids pest infestation in certain areas of New Zealand. The case study will used for classification application using NeuCube. Comparative experiments with conventional machine learning methods are also carried out.

- Part 3 - Conclusion and future direction
  - Chapter 8 summarizes the findings and contributions of this research proposed further future developments. For example; combining ontology-based systems for more organized and systematic modelling of SSTD, to enhance NeuCube M1's optimisation strategies, dealing with variability in data and multiple type of data.

1.5 Thesis Contribution

This is the first comprehensive study of utilising personalised modelling based on spiking neural network methods resulting in several contributions to the areas of both information science and bioinformatics.

During the course of this study, several novel contributions have been applied including analysing the problems related to global modelling and conventional personalised modelling for SSTD and their respective potential solutions; development of a prototype system based on the PMeSNNr framework called NeuCube M1 which comprises an encoding method employing Address Event Representation (AER) algorithm; a recurrent 3D SNN reservoir based on the Liquid-State Machine (LSM) concept and implementation of Spike Time Dependent Plasticity (STDP) as a learning rules; an innovative input variables mapping techniques utilizing Factor Graph Matching (FGM) algorithm; a predictive personalised modelling method for early event prediction; various selections of evolving spiking neural network classifiers including a novel extended dynamic evolving spiking neural network method called deSNNs_wkNN for multi-NN classification and regression problems; a grid-search optimisation module and visualisation of the spiking network activities specifically on a group and personalised level. All these contributions are described and applied in Chapters 4, 5, 6 and 7. The methods have been applied to two real world case studies which are stroke occurrences prediction and aphid pest population prediction.

This study has found a number of interesting results. Firstly is that using spiking neural networks for personalised modelling is more suitable for analysing and modelling SSTD dynamically compared with conventional machine learning meth-

ods that use global modelling, thus verifying the validity of this approach and that this methodology has also achieved a better results in terms of prediction accuracy. Secondly, using this approach, early event prediction is possible where the time length of the training data (samples, collected in the past) and the test data (samples used for prediction) can be differentiate. Early event prediction is very crucial when solving important ecological and social tasks and disease risk prediction described by temporal-and/or spatio-temporal data, such as stroke risk prediction, pest population burst prevention, natural disaster warning, financial crisis prediction. Thirdly, that these methods take all features without the need to filter noise and still produce good results. Fourthly, the innovative input variables mapping techniques enable dynamics mapping of SSTD variables and assist in revealing unknown spatio-temporal patterns and its associations. Lastly, the visualisation of spiking network activities enables deep network learning of the spiking patterns. This assists us in understanding the spiking neurons connection and relationships. Furthermore this visualisation reveals new knowledge about the SSTD that deserves to be investigated further.

NeuCube revealed hidden associations amongst environmental features in stroke prediction case study where the associations of environmental factors suggest there is influence on stroke occurrences. We also discovered that there is a cascading effect, unique to each individual depending on their exposure to certain environmental factors within a specific time window. This study has also successfully and accurately predicted the risk of stroke occurrences at an earlier time point then produces models and demonstrates that analysing all the features collectively can accurately predict stroke risk. The second case study on ecological data in aphid pest abundance prediction, verified NeuCube's capability in modelling any type of SSTD. The result has been an earlier prediction of aphid pest abundance to assist in timely agricultural management.

This study gives light to future research directions for personalised modelling based on SNN with the improvements in the NeuCube architecture for SSTD processing and personalised profiling. The main results of this study emphasise the new discoveries that have been published as conference papers and will further published as journal papers.

1.6 Publication List

- **Journal**

1. Kasabov, N., Feigin, V., Hou, Z.G., Chen Y., Liang, L., Krishnamurthi, R., Othman, M., Parmar, P. (2014). Evolving spiking neural network method and systems for fast spatio-temporal pattern learning and classification and for early event prediction with a case study on stroke. Neurocomputing, Volume 134, 25 June 2014, Pages 269-279.
2. Nikola Kasabov, Nathan Scott, Enmei Tu, Stefan Marks, Neelava Sengupta, Elisa Capecci, Muhaini Othman, Maryam Doborjeh, Norhanifah Murli, Reggio Hartono, Josafath Israel Espinosa-Ramos, Lei Zhoua,Fahad Alvi, Grace Wang, Denise Taylor, Valery Feigin, Sergei Gulyaeh, Mahmoud Mahmoud, Zeng-Guang Hou,Jie Yang. Evolving Spatio-Temporal Data Machines Based on the NeuCube Neuromorphic Framework: Design Methodology and Selected Applications, Neural Networks, Preliminary Accepted 2015.
3. Kasabov, N., Othman, M., Tu, E., Krishnamurthi, R., Feigin, V. Personalised Predictive Modelling with Spiking Neural Networks: Predicting Stroke Risk, Nature Reviews Neurosciences, submitted 2015.

- **Conference**

1. Othman, M., Kasabov, N., Hu, Y. (2012), Spatial-Temporal Data Representation in Ontology System for Personalized Decision Support, Talent Management Symposium 2012, Northern Melbourne Institute of TAFE, Australia, UTHM Publisher.
2. Othman M, Kasabov N, Tu E, Feigin V, Krishnamurthi R, Hou Z, et al (2014). Improved predictive personalized modelling with the use of Spiking Neural Network system and a case study on stroke occurrences data. Neural Networks (IJCNN), 2014 International Joint Conference on; 2014: IEEE; 2014. p. 3197-3204.
3. Tu E, Kasabov N, Othman M, Li Y, Worner S, Yang J, et al. (2014). NeuCube (ST) for spatio-temporal data predictive modelling with a case

study on ecological data. Neural Networks (IJCNN), 2014 International Joint Conference on; 2014: IEEE; 2014. p. 638-645.

4. Keynote Speaker on behalf of Prof Nikola Kasabov at New Zealand Applied Neuroscience Conference (NZANC), Auckland University of Technology, New Zealand on 19th September 2014 on a paper titled Personalised Predictive Data Modelling Methods and Case Study Applications.

5. Othman, M., Kasabov, N., Tu, E., Feigin, V., Krishnamurthi, R.(2015), Using NeuCube, 13th International Conference on Neuro-Computing and Evolving Intelligence 2015, Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand.

- **Abstract**

1. Othman, M., Breen, V., Kasabov, N. (2014), Personalised Predictive Data Modelling Methods and Case Study Applications, New Zealand Applied Neuroscience Conference (NZANC), Auckland University of Technology, New Zealand.

2. Othman, M., Kasabov, N., Tu, E., Feigin, V., Krishnamurthi, R.(2015), Using NeuCube, 13th International Conference on Neuro-Computing and Evolving Intelligence 2015, Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, New Zealand.

3. Othman, M., Kasabov, N. (2015). Extended Dynamic Evolving Spiking Neural Network for Spectro-Spatio Temporal Pattern Multi-NN Classification, Evolving System.

- **Poster**

1. Othman, M., Kasabov, N., Hu, Y. (2012), Spatial-Temporal Data Representation in Ontology System for Personalized Decision Support, 12th International Conference of Neuro-Computing and Evolving Intelligence 2012 (NCEI'12), Knowledge Engineering and Discovery Institute, Auckland University of Technology, New Zealand.
2. Othman, M., and Kasabov, N. (2013), Spatial-Temporal Data Representation and Processing in Ontology-based System for Personalized Decision Support, 26th Australasian Joint Conference on Artificial Intelligence 2013, University of Otago, Dunedin, New Zealand.
3. Othman, M., Kasabov, N., Tu, E., Feigin, V., Krishnamurthi, R. (2015), Evolving Spiking Neural Networks for Predictive Data Modelling, 13th International Conference of Neuro-Computing and Evolving Intelligence 2015 (NCEI'15), Knowledge Engineering and Discovery Institute, Auckland University of Technology, New Zealand.

Chapter 2

PERSONALISED MODELLING: A REVIEW

"The measure of intelligence is the ability to change."

- Albert Einstein

2.1 Introduction

This chapter reviews the concept of the personalised modelling method. However before the personalised modelling method can be discussed in detail, the basis of data modelling and pattern recognition approaches need to be addressed briefly. Inductive and transductive inference approaches are two of the most basic theories for data modelling and the main idea behind global, local and personalised modelling methods.

2.2 Inductive and Transductive Inference Approaches

Inductive and transductive inference approaches are commonly used to build models and systems for data analysis and pattern recognition [Kasabov 2009b]. Inductive inference approaches will create a single function (a model) based on historical data to predict a future event [Levey 1999]. In the inductive inference approach the model is created based on the analysis of the entire problem space (global space) without taking into account the information related to the new data vector. Neglecting information from the new data vector raises an issue about the relevance of global

modelling to produce an accurate model or solution to a specific problem. Figure 2.1 illustrates the inductive inference approach. The engine will train on historical data and create a global function to model incoming new data. Popular inductive inference approaches are Support Vector Machine (SVM) [Cortes 1995], Multi-Layer Perceptron [Hornik 1989] and Linear Regression.

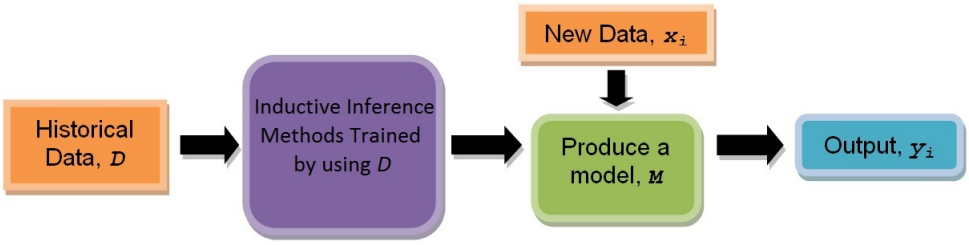

Figure 2.1: Inductive inference approach.

The transductive inference was introduced by [Vapnik 1998] as a solution to solve the issue raised by the inductive inference engine. This approach creates a model based on observations of a specific group of data vectors and only focuses on one point in the space (local space). Transductive inference takes into account the additional information of the new data vector to find relevant information for analysis purposes. This in the end will create many different specific models (functions), to test every new data vector. Figure 2.2, illustrates a basic process of transductive inference.

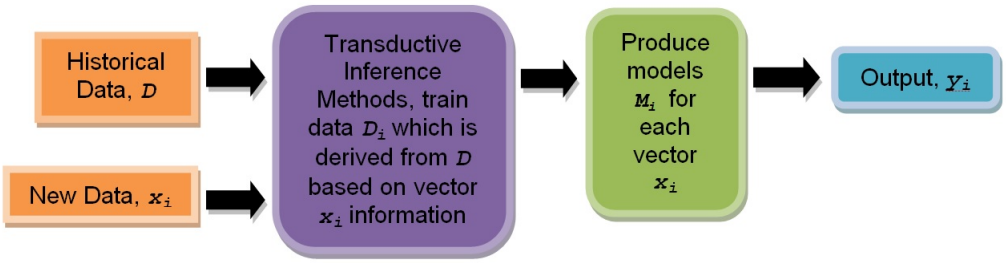

Figure 2.2: Transductive inference approach.

Several types of advanced transductive inference model have been build such as Transductive RBF Neural Network with Weighted Data Normalization - TWRBF [Song 2004] and Transductive Neural Fuzzy Inference System with Weighted Data

Normalization - TWNFI [Song 2006] and successfully applied for medical decision support and time series prediction. As a result the transductive inference approach is considered the most suitable approach toward building a learning model for the application of personalised decision support, especially in medical application or event prediction. Since individual personal features of a patient or event are important to consider for future prediction or treatment decision.

2.3 Global, Local and Personalised Modelling

2.3.1 Introduction

In computational intelligence modelling and learning, the main techniques are global, local or personalised modelling which are derived from inductive and transductive inference approaches. Global modelling produces a model from the data for the whole problem space. The model represents the data by a single function whereas local modelling creates a set of models from data where each model represents a cluster of the whole problem space. These models can be a set of functions or set of rules. Personalised modelling on the other hand utilises transductive reasoning to create a specific model for each data point (a patient, an event) within a localised problem space.

2.3.2 Global Modelling

Support vector machine (SVM) also called support vector networks is one of the most popular algorithm used for global modelling. It is very efficient in classifying static and vector-based data using few training samples. However, SVM is not suitable to analyse high-dimensional dataset like SSTD.

2.3.2.1 Support Vector Machine

Support vector machine is widely used for classification and regression problems. Originally the SVM algorithm was created by Vladimir Vapnik in 1963 [Vapnik 1963] then new SVM with 'soft margin' approach was introduced by Vladimir Vapnik and colleagues in 1995 [Cortes 1995]. After that, several other extended versions has been

developed such as Least Square SVM (LSSVM) [Suykens 1999], Linear Proximal SVM [Fung 2001], Wavelet SVM [Zhang 2004], Smooth SVM (SSVM) [Lee 2001] and the robustness of SVM still inspired researchers to extend the algorithm, current examples like SVM-Wavelet Transform [Mohammadi 2015], Cluster SVM (CSVM) [Harris 2015] and many more. Since the active development of the SVM algorithms, a group of researcher developed a library for SVM called LibSVM [Chang 2011] to support users in implementing their application using SVM.

Fundamentally SVM is based on the concept of decision planes that define decision boundaries. The decision planes (hyperplanes) are like clear gaps that separate a set of objects that belong to different classes, the distance from the hyperplane to the data is maximized (also known as the maximum margin hyperplane). For example for a linear SVM (illustrated in Figure 2.3), the set of objects either belong to class RED or BLUE. The line represents the linear decision surface that separates between RED and BLUE class. When a new object (black circle) is added to the problem space, it will be mapped to the features space of these two planes either in RED or BLUE. Depending on where it is mapped, it will be classified as RED when it falls in the left plane and BLUE if it falls in the right plane.

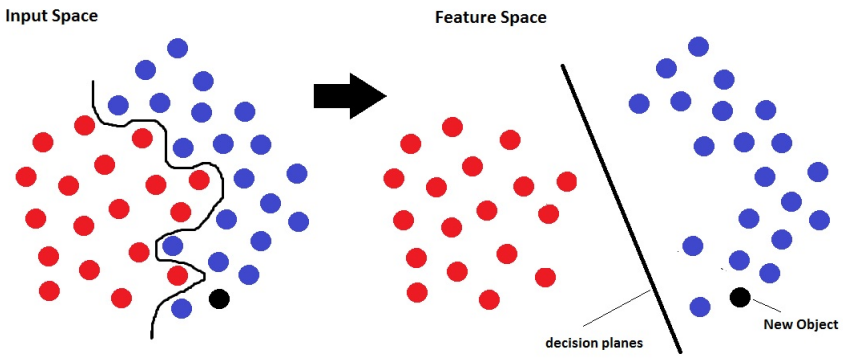

Figure 2.3: Overview of simple SVM transformation (mapping).

In mathematical terms, linear SVM can be defined as follows. Given a set of data that can be linearly separated:

$$D = \{x_i, y_i \mid x \in R^p, y \in \{-1, 1\}\}_{i=1}^{n} = size \tag{2.1}$$

where D is the training data, x_i is a p-dimensional vector, n is a set of data points,

and y_i is either -1 or 1, indicating which class x_i belongs to.

Maximum margin hyperplane is found using Equation 2.2, to separate the two classes.

$$w.x - b = 0 \tag{2.2}$$

where w the normal vector to the hyperplane, b is a scalar and . denotes the dot product.

Two hyperplanes can be selected to separate the data, where there no data points lies between them and try to maximize their distance. The region bounded by the hyperplanes is called the margin and is described by the following equations.

$$w.x - b = 1 \tag{2.3}$$

and

$$w.x - b = -1 \tag{2.4}$$

Constraints must be added to keep the data point from falling inside the margin and to classify each sample into a specific class. The constraints are:

$$w.x_i - b <= -1 \tag{2.5}$$

where x_i belong to first class, and

$$w.x_i - b >= 1 \tag{2.6}$$

where x_i belong to second class.

Figure 2.4 shows the overview of linear SVM.

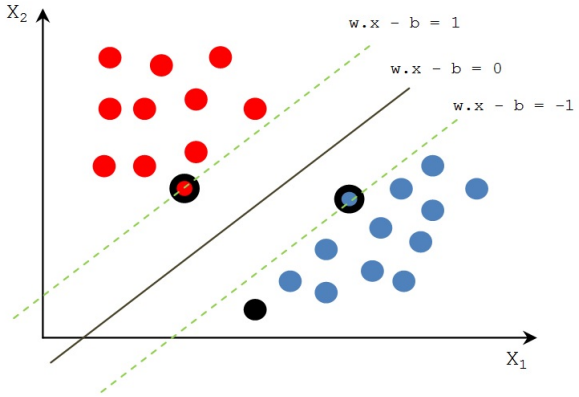

Figure 2.4: Overview of simple linear SVM. The samples on the margin are called support vectors.

To overcome the issue of inseparable data where some data cannot be linearly separated, nonlinear SVM is introduced by applying kernel approach to find maximum margin hyperplanes. The data is initially transformed into high dimensional space a using nonlinear kernel function, then the standard algorithm is used to find the maximum margin hyperplanes [Boser 1992]. Several types of kernel can be utilized in SVM which include linear, polynomial, radial basis function (RBF) and sigmoid.

- Linear: $K(x_i, x_j) = x_i.x_j$
- Polynomial: $K(x_i, x_j) = (\gamma x_i.x_j + C)^d$
- Radial Basis Function:$K(x_i, x_j) = exp(-\gamma \mid x_i.x_j \mid^2)$
- Sigmoid: $K(x_i, x_j) = tanh(\gamma x_i.x_j + C)$

where $K(x_i, x_j) = \delta(x_i).\delta(x_j)$. The kernel function represents a dot product of input data points mapped into the higher dimensional feature space by transformation δ. Gamma (γ) is an adjustable parameter of certain kernel function.

One of the disadvantages of SVM is that it has a high computational burden because of the quadratic programming, making it slow in the training phase [Horváth 2003]. Another drawback is the choice of kernels and kernel parameter determination suitable for the data under investigation. Kernel models are sensitive to over-fitting the model selection criterion [Cawley 2010]. Domain knowledge is also hard to incorporate in SVM, especially new information about the new sample.

2.3.3 Local Modelling

The local modelling approach was created to overcome the drawbacks of global modelling where it is more adaptable to the new data vector, and to create a model to represent the cluster within which the new data vector resides. This has made local modelling methods more suitable to analyse individual samples than global modelling. Evolving Classification Function (ECF) is one example of local modelling methods and is built based on the concept of Evolving Connectionist System (ECOS) [Kasabov 2002].

2.3.3.1 Evolving Classification Function (ECF)

ECOS are systems that evolve in time through interaction with the environment; it is adaptable to changes in the system through new incoming information [Kasabov 1998b]. Evolving Classification Function (ECF) was developed based on ECOS principles has four layers of neurons (nodes) which represent input variables, fuzzy memberships functions, a set of data centers in input spaces and classes [Kasabov 2002]. ECF methods exhibit fast incremental on-line and off-line learning and have dynamic environments that allocate rule nodes to help users understand and verify the model's functionality. Figure 2.5 illustrates clusters of nodes in the ECF environment, based on the information of new input vector (n_i) ECF will produce clusters of rule nodes that are identified by its center (o_j), radius (r_j) and class (C).

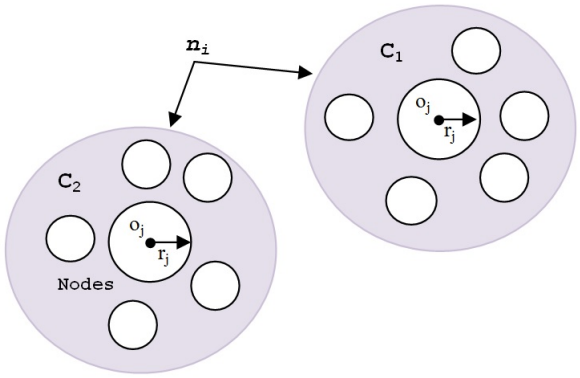

Figure 2.5: An example of evolving clusters in ECF.

2.3.4 Personalised Modelling

Personalised modelling is different from global modelling because it will create a specified model for each new data vector based on the samples that are closest to the new data vector in the dataset. Other than advance transductive methods listed above, methods that can be categorised as personalised modelling are k-Nearest Neighbour (kNN), weighted k-Nearest Neighbour (wkNN) and weighted-weighted k-Nearest Neighbour (ww$k$NN).

2.3.4.1 *k*-Nearest Neighbour(*k*NN)

The k-Nearest Neighbour (kNN) method is a supervised learning algorithm that has been successfully used for classifying sets of samples based on nearest training samples in a multi-dimensional feature space, and was originally proposed by [Fix 1951]. The basic idea behind the kNN algorithm is depicted in Figure 2.6:

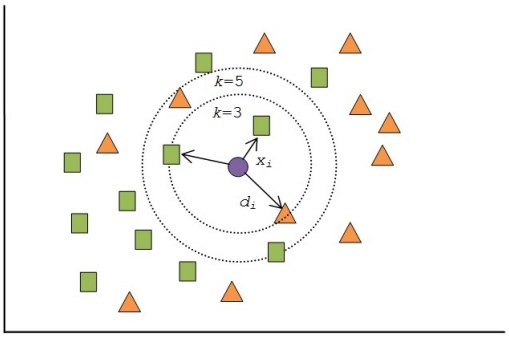

Figure 2.6: kNN modelling.

The kNN modelling:

- Firstly, a set of pairs features (e.g. (x_1, y_1),...(x_n, y_n)) are defined to specify each data point, and each of those data points are identified by the class labels $C = c_1, ...c_n$.
- Secondly, a distance measure (d_i) is chosen (e.g. Euclidean distance, or Manhattan distance) to measure the similarity of those data points based on all their features.
- Finally, the k-nearest neighbours are found for a target data point by analyzing similarity and using the majority voting rule to determine which class the target data point belongs to.

2.3.4.2 Weighted *k*-Nearest Neighbour (w*k*NN)

The weighted k-Nearest Neighbour (wkNN) is designed based on the transductive reasoning approach, which has been widely used to evaluate the output of a model focusing solely on an individual point of a problem space using information related to the individual [Vapnik 1998]. In the wkNN algorithm, each single vector requires a

local model that is able to best fit each new input vector rather than a global model, thus each those new input vectors can be matched to an individual model without taking into account any specific information about existing vectors. In contrast to the kNN algorithm, the output values of a new input vector (y_i), is not only dependent upon its output values of k-nearest neighbour vectors (y_j), but also upon the weight (w_j) that is decided by the distance between existing vectors and the new input vector. This is the basic idea behind the wkNN algorithm. Mathematically wkNN can be described as:

$$y_i = \sum_{j=1}^{k} \frac{w_j y_j}{w_j} \tag{2.7}$$

where weight (w_j) is calculated based on the distance of k-nearest neighbour vectors to new vector using the following equation:

$$w_j = [max(d) - (d_j - min(d))]/max(d) \tag{2.8}$$

The vector $d = [d_1, d_2, ...d_{N_i}]$ is defined as the distances between input vector (x_i) and the k nearest neighbour (x_1, y_1) for $j = 1$ to k. The Euclidean distance measured between new vector (x_i) and neighbouring vector (x_j) is calculated based on:

$$d_j = sqrt[\sum_{l=1}^{V}(x_{i,l} - x_{j,l})^2] \tag{2.9}$$

where V is the number of the input variables, $x_{i,l}$ and $x_{j,l}$ are the values of the variables in vector x_i and x_j, respectively. An example of wkNN implementation in a classification problem that consists of two classes, represented by 0 (class 1) and 1 (class 2) as output class labels. If the new vector (x_1) belongs to class 2, this means it has "personalised probability". To classify the new vector (x_1) into classes, there has to be probability threshold selected P_{thr}, so if the output value $y_i \geq P_{thr}$ then the new vector (x_1) will be classified into class 2. For example the probability threshold value is set to 0.5 and if the output value is 0.75 which is more than the probablity threshold, the new vector will be classified into class 2 not class 1 where the output value should fall within the range of $0 \leq y_i \leq 0.5$.

2.3.4.3 Weighted-Weighted k-Nearest Neighbour (ww$k$NN)

The weighted-weighted k-Nearest Neighbour (ww$k$NN) is a novel personalised modelling algorithm which was proposed by [Kasabov 2007b]. The basic idea behind this algorithm is the output of each new input vector is measured dependent upon its k-nearest neighbours and also upon the distance between the existing vectors and the new input vectors, and the power of each vector which is weighted according to its importance within the sub-space (local space) to which the new input vector belongs. The new Euclidean distance measure is calculated using this equation:

$$d_j = sqrt[\sum_{l=1}^{V}(c_{i,l}(x_{i,l} - x_{j,l}))^2] \tag{2.10}$$

where $c_{i,l}$ is the coefficient weighing variables x_l in the neighbourhood of x_i. The coefficient value is calculated using the Signal-to-Noise Ratio (SNR) procedure that ranks each variables across all vectors in the neighbourhood set D_i of N_i vectors.

$$C_i = (c_{i,1}, c_{i,2}, ..., c_{i,V}) \tag{2.11}$$

$$c_{i,l} = S_l/sum(S_l) \qquad \text{for} \quad l = 1, 2, \cdots, V \quad \text{where} \tag{2.12}$$

$$S_l = abs(M_l^{(class1)} - M_l^{(class2)}/(Std_l^{(class1)} + Std_l^{(class2)}) \tag{2.13}$$

$M_l^{(class1)}$ and $Std_l^{(class1)}$ is the mean value and standard deviation of variable x_l for all vectors in D_i that belong to class 1. The new distance measurement that assigned weight to all variables according to its importance is the new feature in ww$k$NN that differentiates it from wkNN. Weighting variables in personalised models is also used in TWNFI models [Kasabov 2007b], [Song 2006].

2.4 Integrated Method for Personalised Modelling

Personalised modelling framework for gene data analysis and biomedical applications was proposed by [Kasabov 2010b]. The framework is called Integrated Method for

Personalised Modelling (IMPM) (refer to Figure 2.7). The methodology of IMPM is described in Table 2.1 below:

Table 2.1: The IMPM Methodology

1:	Collect, filter and store data D .
2:	Compile new input vector x of a new person.
3:	Select a subset of relevant variables, V_x of the the new input vector x from a global variables set V.
4:	Select k-nearest neighbour vectors K_x from the global data set D and forming a neighbourhoodD_x of similar samples to x using the variables from V_x to define the similarity.
5:	Rank the V_x variables within the local neighbourhood D_x in order of importance to the outcome, obtaining a weight vector W_x.
6:	Train and optimise a local prognostic/ classification model M_x, that has a set of model parameters P_x, a set of variables V_x and local train/test data set D_x.
7:	Generate a functional profile F_x for the person x using the selected set V_x of variables, along with the average profiles of the samples from D_x that belong to different outcome classes, e.g., F_i and F_j.

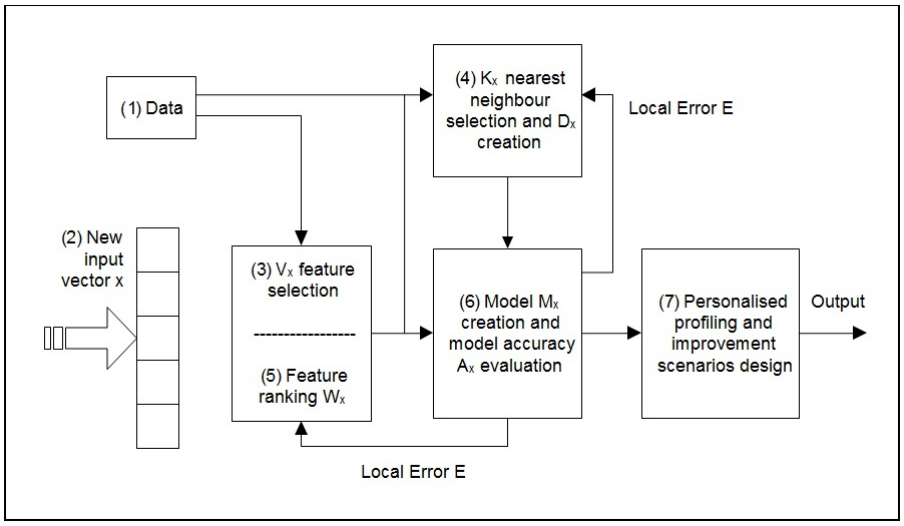

Figure 2.7: Functional block diagram of IMPM [Kasabov 2010b].

2.5 Chapter Summary

Global modelling applied in most conventional machine learning methods has proven its effectiveness in the past, however it has a limited capability in producing mod-

els that fit each person or each case in the problem space since global modelling takes all available data in a problem space and produces a single general function [Kasabov 2007a]. The produced model is applied to a new individual regardless of their unique personal features. Therefore, in the case of specific medical condition e.g. stroke, heart attack and environmental events e.g. earthquake, volcano eruption; personalised modelling methods are preferred for the reason that they can produce a model for each individual/event based on their personal features.

Chapter 3

SPIKING NEURAL NETWORKS: A REVIEW

"Change is the end result of all true learning."

- Leo Buscaglia

3.1 Introduction

This chapter reviews the various approaches to SSTD and SNN principles which are the basis of our methodology. The expected benefits and limitations of past and present research will be addressed. A brief introduction to SSTD modelling techniques will be given together with the fundamental components of SNN including data encoding methods, neuron models and learning algorithms. This chapter will highlight the feasibility of SNN methods for analyzing SSTD.

3.2 Spatio, Spectro Temporal Data Modelling

SSTD incorporates multi-dimensional information like space and time, discrete and continuous representation of the data, making it difficult to model especially using statistical and artificial neural network machine learning methods. Conventional methods either analyse SSTD spatial and temporal elements independently or collectively while ignoring the association present in SSTD. Some emerging methods analysed the space and time elements together, on a frame-by-frame basis which is unreasonable for prediction of future events. Such models are based on the deep learning machine approach, a combination of Deep Belief Networks (DBNs-Generative Model) and Convolutional Neural Networks (CNNs-Discriminative Model) [Arel 2010].

Since future events happen based on influence of past events which occur over time, conventional machine learning methods fail to consider the connections residing in SSTD. One of the popular statistical approaches for processing temporal information is Hidden Markov Models (HMM) which is the simplest form of dynamic Bayesian Network. HMM is a tool for representing probability distributions over a sequence of observations [Ghahramani 2001]. The temporal evolution is modeled as a sequence of probabilistic jumps from one discrete state to the other [Turaga 2008]. HMM has been successfully applied to problems such as computational biology [Krogh 1994], speech recognition [Rabiner 1989], signature verification [Yang 1995], gene expression [Schliep 2003] and many more. However HMM is restricted in defining only a single independent variable and stationary temporal patterns, either time or one-dimensional position [Turaga 2008], which limit its ability to learn more complex SSTD patterns.

The complexity of analyzing SSTD has motivated many researchers to find solutions. Brain inspired spiking neural networks has the ability to learn spatio-temporal patterns by using a train of spikes [Gerstner 2002] because it is a complex and biologically plausible connectionist system. The implementation of the learning rule Spike-Time-Dependent-Plasticity (STDP) [Legenstein 2005] makes it a dynamic and evolving learning system. The reservoir computing that maps input signal into the higher dimension of 3D structure can capture the multi-dimensional nature of SSTD patterns over time through spike transmission, resulting in forming a SSTD memory [Maassa 2002].

As a result, many researchers have strived to develop new SNN models and applications including our group research that have developed several SNN models [Kasabov 2010a], [Schliebs 2010], [Soltic 2010], [Soltic 2011] to process SSTD, however the SSTD is processed as a sequence of static feature vectors extracted from a segment of data overlooking SNN capability in processing the whole SSTD. To address the drawback of previous SNN models, our research group continues the effort to fully utilise the SNN capability by developing new SNN models incorporating reservoir computing techniques [Hamed 2011], [Mohemmed 2011], [Schliebs 2011] and [Dhoble 2012]. The SNN proves to be an evolving and dynamic system; therefore we aim to exploit its capability by further developing new SNN models. Before

going further, a brief introduction to the history of SNN will be addressed.

3.3 History of Spiking Neural Networks

The human brain always intrigues researchers because of its complex cognitive ability. The information processing capabilities of the brain have inspired numerous mathematical abstractions of biological neurons. Through the advancement of electronics, engineering and computer science, researchers have been able to relatively emulate the human brains cognitive ability through artificial neural networking (ANN) models. Initial implementation of ANN was a simple electrical circuit designed and built in by [McCulloch 1943]. In 1952, a scientific model of a spiking neuron was developed to describe the ionic mechanisms underlying the initiation and propagation of action potentials in the giant squid axon and this model is called Hodgkin-Huxley model [Hodgkin 1952c]. Since then many ANN models have been developed either in software or hardware applications such as the Hopfield network [Hopfield 1982]; the Elman net [Elman 1991]; back propagation network [Russell 1995], [Rumelhart 1985] and von Neumann machine [Boahen 2007]. Many of these ANN models have been successfully applied across many disciplines, including classification, time series prediction, and pattern recognition [Kasabov 2010a]. However when applied to complex stochastic system such as brain disease processing, biological and environmental situation most ANNs fail to deliver a satisfactory result. This resulted in the emergence of spiking neural networks (SNN) as the new generation of neural network models that are vital to computational functionality.

SNN is the third generation of neural network (NN) models. Their mechanisms are more realistic in terms of spiking processes to the biological neurons because it utilises trains of spikes to represent and process pulse coded information [Maass 2001]. The biologically realistic information processing ability of SNN models can produce solutions to numerous real world problems. SNNs represent biological neural networks where all neurons are connected through synapses and electrical signals pass information from one neuron to another. SNN allows the representation of multi-dimensional information (e.g. space, time, frequency) in communication and computation, just like a real neuron. In recent years, SNN

has become a powerful computational tool that is widely adopted for diagnosis and prognosis of a disease, as evidenced by over 500 published papers each year featuring neural network applications in medicine [Gant 2001]. Since the introduction of SNN, many advancements has been made in developing novel biological spiking neuron models such as Hodgkin-Huxley's model [Hodgkin 1952b], Spike Response Models (SRM) [Gerstner 1995]; [Gerstner 2002], Integrate-and-Fire Models [Maass 2001] and Izhikevich models [Izhikevich 2004]; [Izhikevich 2006]; [Izhikevich 2008]. Advantages of SNN over traditional ANN are that SNN needs fewer neurons to accomplish the same task as ANN [Maass 2001];[Gerstner 2002], they are more biological plausible [Maass 2001]; and have the capability of estimating functions. Moreover SNN require less computing time and memory because SNN neurons communicate through a spike signal instead of analog values and can be multiplexed as binary codes. Additionally SNN have more domain range capabilities such as spatio-temporal domain, complex domains requiring massive scale networks (several thousand neurons) and domains requiring biologically derived models.

3.4 Neuron Models

The SNNs are model from one of the basic component of the central nervous system (CNS) which is neuron. Therefore to understand SNNs functionality, a brief overview of biological neuron will be addressed first followed by artificial neuron models.

The central nervous system (CNS) is composed entirely of two kinds of specialized cells: neurons and glia [Stufflebeam 2008]. A neuron will process and transmit information through electrical and chemical signals. Basically, a neuron will receive information from other neurons and then send output information to other neurons. The neurons are connected via synapses in which the information flows from one neuron to another. Neurons are connected to form neural networks.

A typical neuron possesses four parts that are depicted in Figure 3.1. The first part is the cell body or soma, is the processing unit of a neuron. The second and third parts are called the dendrites and axon where dendrites will receive the incoming signals from other neurons and outgoing signals will go through the axon.

A neuron may have thousands of dendrites but only one axon. The fourth part is the axon terminal that lies at the end of the axon where it contains neurotransmitters, a chemical medium in which a signal will flow through to next neuron. Synapses connect an axon to another axon and a dendrite to another dendrite.

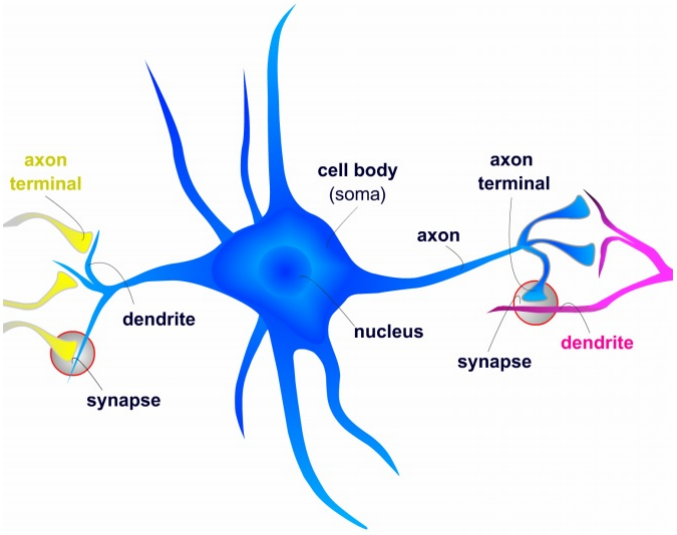

Figure 3.1: Biological neuron model [Stufflebeam 2008].

3.4.1 Biological Neurons

Action Potential

Action potentials (electrical signals) are transmitted along axons through conduction. Initially an action potential is generated near the soma of the axon when ions move across the neuronal membrane. The inflow and outflow of ions during neurotransmission will make the inside of the target neuron more positive and when it reaches a threshold level a large electrical signal is generated. This signal is transmitted along the axon until it reaches the axon terminal where neurotransmission begins again. The output from electrical synapses will be electrical signals but when it reaches chemical synapses the output will be a neurotransmitter.

Neurotransmission

Neurotransmission is a communication process between neurons through the movement of chemical or electrical signals across a synapse. Figure 3.2 illustrates the difference between chemical synapses and electrical synapses. In chemical neuro-

transmission, the pre-synaptic neurons and the post-synaptic neuron are separated by a small gap called the synaptic cleft that is filled with extracellular fluid. The synaptic cleft acts as a chemical messenger that links the action potential of a neuron with a synaptic potential of another neuron.

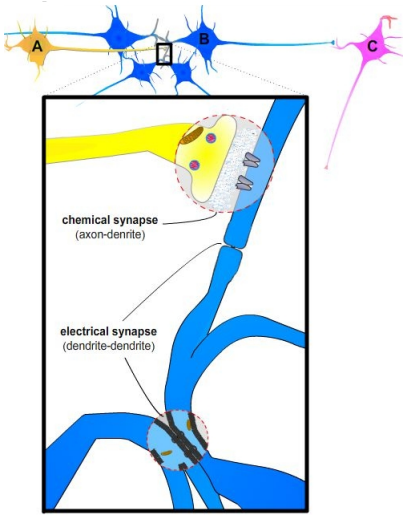

Figure 3.2: Example of chemical synapse and electrical synapse [Stufflebeam 2008].

3.4.2 Artificial Neuron

The artificial neurons are designed to mimic biological neurons. Figure 3.3 depicts a general form of an artificial neuron. The information process in artificial neurons is simply a weighted sum of inputs compared to a threshold value. If the sum value is higher than the threshold the output will either be set to 1 or -1.

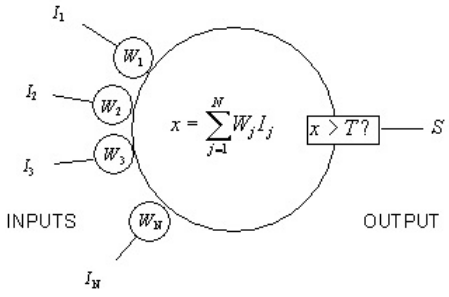

Figure 3.3: A general form of artificial neuron.

The similarity between biological neuron to artificial neurons can be seen in the dendrites, soma and axon parts. Dendrites allow a neuron to receive input signals from a large number of neighboring neurons. These dendrites act as input

vector receptors and multiply the input by that dendrites weight value. In biological neurons, the soma acts as a summation function as each input signal arrives from the dendrites to create a solution or output. The axon will receive its signal from the summation behavior of the soma. When the soma reaches a certain threshold (action potential), the axon will propagate an electrical signal pulse down its length. Biological neurons fire a discrete pulses which be translated into a continuous value.

A SNN neuron simulates how the biological neuron processes the information with two major characteristics. First, the weights from incoming input vectors have different strengths and second, a spike is released when it reaches or exceeds a given threshold. There are many neuron models developed to simulate biological neuron process. The following section provides a short description on some of these neuron models.

3.4.2.1 Hodgkin-Huxley Model

In learning how a biological neural really works, Hodgkin and Huxley wrote a series of five papers describing the experiments they conducted on a giant squid axon. These experiments aimed to discover and simulate the movement of ions in a nerve cell during an action potential. In the initial four papers [Hodgkin 1952e], [Hodgkin 1952c], [Hodgkin 1952a], [Hodgkin 1952d], the experiments aimed to examine the function of the neuron membrane, the effect of changes in sodium concentration on action potential, the effect of sudden potential changes on action potential as well as ionic conductance. The fourth paper outlined how the inactivation process reduces sodium absorptivity. The fifth paper compiled all the findings of their previous experiments and transformed it into a mathematical model [Hodgkin 1952b]. Figure 3.4 illustrates the circuit model of an axon.

In the nerve cell, Hodgkin and Huxley discovered three important components which are sodium, potassium and a Cl-ions leak current. Capacitors are used to model the charge storage capacity of the cell membrane, while resistors are used to model the numerous type of ion channels embedded in the membrane, and batteries are used to represent the electrochemical potentials established by differing intra- and extracellular ion concentrations [Nelson 2004].

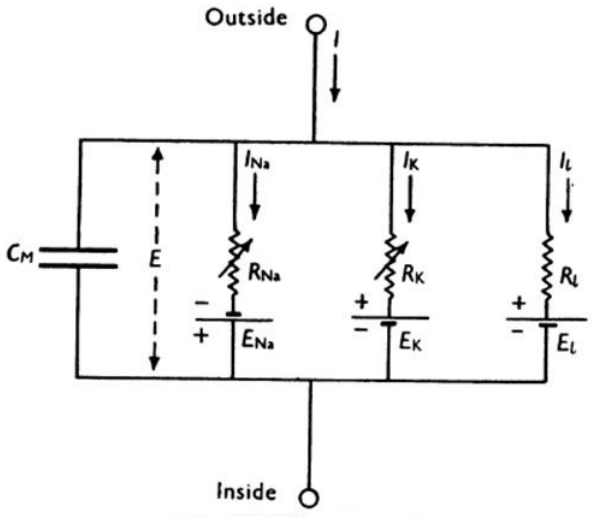

Figure 3.4: Circuit model of an axon[Hodgkin 1952b].

Mathematically the model can be described as:

$$I_{Inward} = C_m \frac{dV}{dt} + I_{Na} + I_k + I_l \tag{3.1}$$

where I_{Na} represent sodium, I_k potassium, and I_l leakage ions. Each current can be determined by a driving force which is represented by a voltage difference and a permeability coefficient. Conductance is the inverse of resistance and equations are derived using Ohm's law (V=IR)

$$I_{Na} = g_{Na}(E - E_{Na}) \tag{3.2}$$

$$I_k = g_k(E - E_k) \tag{3.3}$$

$$I_l = g_l(E - E_l) \tag{3.4}$$

where g_{Na} and g_k are both functions of time and membrane potential. E_{Na}, E_k, E_l, C_m and g_l are all constants that are determined via experimentation. The model can further be expanded by adding the following relationships:

$$I_{Na} = g_{Na}(V - V_{Na}) = g_{Na}(E_{Na} - E_R) \tag{3.5}$$

$$I_k = g_{Na}(V - V_k) = g_k(E_k - E_R) \tag{3.6}$$

$$I_l = g_{Na}(V - V_{Na}) = g_{Na}(E_l - E_R) \tag{3.7}$$

$$V = E - E_R \tag{3.8}$$

where E_R is the resting potential. The simplification made in the Hodgkin-Huxley model is that the Ionic currents can be modeled with accuracy by first order differential equations. The equations that govern n, m, and h are described below:

$$\frac{dn}{dt} = \alpha_n(1-n) - \beta_n n \tag{3.9}$$

$$\frac{dm}{dt} = \alpha_m(1-n) - \beta_m m \tag{3.10}$$

$$\frac{dh}{dt} = \alpha_h(1-n) - \beta_h h \tag{3.11}$$

where n is a dimensionless variable that varies from 0 to 1, m is the proportion of activating carrier molecules (ion channels) and h is the proportion of inactivation carrier molecules (ion channels). While α and β are rate constants that are similar to the rate constants for the potassium conductance.

It can be seen that the Hodgkin-Huxley model can reproduce electrophysiological measurements very accurately and its parameter can be easily evaluated from the experiments. However, due to the models complexity, it is computationally expensive making it inappropriate for large networks of spiking neurons. It also has some disadvantages linked to the approximations required [Saïghi 2008].

3.4.2.2 Leaky Integrate and Fire Model (LIFM)

The state of a neuron in the Leaky Integrated and Fire model (LIFM) is characterized by its membrane potential where it will receive excitatory or inhibitory contributions by synaptic inputs that arrive from other neurons [Burkitt 2006].

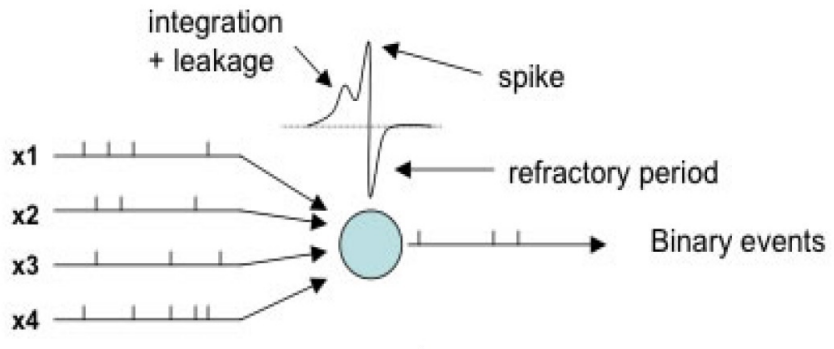

Figure 3.5: Leaky Integrated and Fire Model (LIFM) [Kasabov 2012a].

Figures 3.5 and 3.6 depict the structure and functionality of LIFM where the membrane potential increases with every input spike at a time t multiplied by the synaptic efficacy (strength) until it reaches a threshold and emits an output spike, then the membrane potential is reset to an initial states (e.g. 0) [Kasabov 2012a].

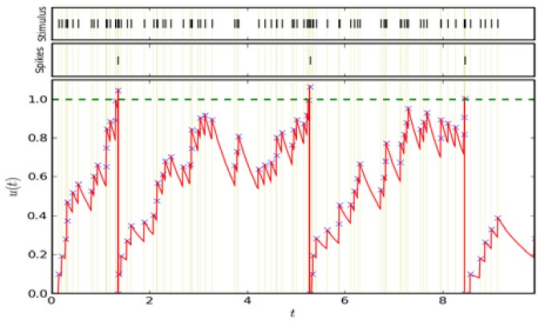

Figure 3.6: Functionality of Leaky Integrated and Fire Model [Kasabov 2012a]

The membrane potential will leak in between spikes and is defined by a set of parameters. The neuron is leaky as the summed contributions to the membrane potential decay with a characteristic time constant [Burkitt 2006]. The mathematical equation of this model is:

$$I_t - \frac{V_m(t)}{R_m} = C_m \frac{dV_m(t)}{dt} \tag{3.12}$$

where R_m is the membrane resistance. This forces the input current to exceed some threshold $I_{th} = V_{th}/R_m$ in order to cause the cell to fire, else it will simply leak out any change in potential. The firing frequency will be set to:

$$f(I) = \begin{cases} 0 & , \quad I \leq I_{th} \\ \left[t_{ref} - R_m C_m log\left(1 - \frac{V_t h}{I R_m}\right) \right]^{-1} & , \quad I \geq I_{th} \end{cases} \tag{3.13}$$

This converges for large input currents to the previous leak-free model with refractory period. In a network the neurons can also be stimulated by a presynaptic neuron j. The synaptic input of neuron i is the weighted sum over all the current generated by the presynaptic neurons and can be represented as :

$$I_i(t) = \sum_j W_{ij} \sum_f \alpha(t - t_j^{(f)}) \tag{3.14}$$

where, weight W_{ij} reflects the strength of the synapses from neuron j to i. And α represents the time course of the postsynaptic current. The simplicity of the LIF model makes its suitable for use in large scale networks allowing for efficient simulation.

3.4.2.3 Spike Response Model (SRM)

A Spike Response Model (SRM) is the generalisation of a LIFM and gives a simple description of action potential generation in neurons. In a SRM, the state of the neuron i is defined by a single parameter $u_i(t)$ (membrane potential). The u_i continues to be at resting value $u_{rest} = 0$, until it receives a spike. On reaching the membrane threshold ϑ, the neuron fires and the membrane potential resets to its resting potential. Assuming that the neuron i has fired its last spike at time $\hat{t}_i$, then the evolution of u_i after firing can be expressed as:

$$u_i(t) = \eta(t - t_i) + \sum_j w_{ij} \sum_f \varepsilon_{ij}(t - \hat{t}_i, t - \hat{t}_j^{(f)}) + \int_0^\infty \kappa(t - \hat{t}_i, s) I_{ext}(t - s) ds \quad (3.15)$$

where the spike time of presynaptic neuron j is denoted by $\hat{t}_j^{(f)}$, η is a function that describes the form of action potential and after potential, t denotes time course of postsynaptic potential and w_{ij} represents synaptic efficacy. The linear response of the membrane for external input current I_{ext} is represented by the kernel function κ. Compared to the LIF model, the membrane threshold of SRM is not fixed but may depend on $t - \hat{t}_i$, hence

$$\vartheta \rightarrow \vartheta(t - \hat{t}_i) \quad (3.16)$$

The SRM is ideal for simulating large numbers of neurons in a network due to its simplicity. Compared to LIF neuron, SRM also allows covering refractoriness. According to [Gerstner 2002] refractoriness may be characterized experimentally by the observation that immediately after a first action potential, it is impossible (absolute refractoriness) or more difficult (relative refractoriness) to excite a second spike [Fuortes 1962]. By setting the dynamic threshold during a time Δ^{abs} to an extremely high value that cannot be attained an absolute refactoriness can be accomplished.

There are many ways to simulated relative refractoriness, (see Figure 3.7). First, after the membrane potential spike, and hence η, passes through a regime of hyperpolarization (after-potential) where the voltage is below the resting potential. During this phase, more stimulation than usual is needed to drive the membrane potential above threshold. This is equivalent to a transient increase of the firing threshold. Second, ε and κ contribute to relative refractoriness because, immediately after an action potential, the response to incoming spikes is shorter and, possibly, of reduced amplitude [Fuortes 1962]. Thus more input spikes are needed to evoke the same depolarization of the membrane potential as in an 'unprimed' neuron. The first argument of the ε function (or κ function) allows us to incorporate this effect.

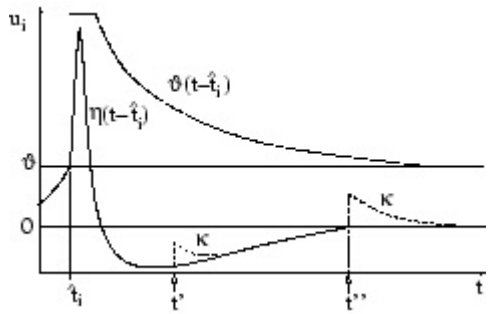

Figure 3.7: Schematic interpretation of SRM [Gerstner 2002].

3.4.2.4 Probabilistic Spiking Neuron Model

Illustrated in Figure 3.8 is a Probabilistic Spiking Neuron Model which has a dynamic synaptic connection [Kasabov 2010a]. As defined in [Kasabov 2012a], the state of a spiking neuron n_i is described by the sum $PSP_i(t)$ of the inputs received from all m synapses. When the $PSP_i(t)$ reaches a firing threshold $\vartheta_i(t)$, neuron n_i fires, i.e. it emits a spike. Connection weights $(w_{j,i}, j = 1, 2, ..., m)$ associated with the synapses are determined during the learning phase using a learning rule. In addition to the connection weights $w_{j,i}(t)$, the probabilistic spiking neuron model has the following three probabilistic parameters:

- A probability $p_{cj,i}(t)$ that a spike emitted by neuron n_j will reach neuron n_i at a time t through the connection between n_j and n_i. If $p_{cj,i}(t) = 0$, no connection and no spike propagation exist between neurons n_j and n_i. If $p_{cj,i}(t) = 1$ the probability for propagation of spikes is 100%.

- A probability $p_{sj,i}(t)$ for the synapse $s_{j,i}$ to contribute to the $PSP_i(t)$ after it has received a spike from neuron n_j.
- A probability $p_i(t)$ for the neuron n_i to emit an output spike at time t once the total $PSP_i(t)$ has reached a value above the PSP threshold (a noisy threshold).

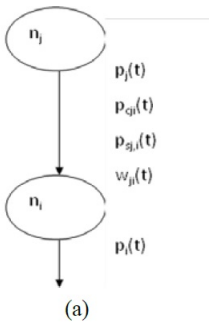

Figure 3.8: Probalistic Spiking Neuron Model [Kasabov 2010a].

The total $PSP_i(t)$ of the probabilistic spiking neuron n_i is calculated using the following formula:

$$PSP_i(t) = \sum_{p=t_0,.,t} \left(\sum_{j=1,\ldots,m} e_j f_1(p_{cj,i}(t-p)) f_2(p_{sj,i}(t-p)) w_{j,i}(t) + \eta(t-t_0) \right) \tag{3.17}$$

where:

e_j is 1, if a spike has been emitted from neuron n_j, and 0 otherwise;
$f_1(p_{cj,i}(t))$ is 1 with a probability $p_{cj,i}(t)$, and 0 otherwise;
$f_2(p_{sj,i}(t))$ is 1 with a probability $p_{sj,i}(t)$, and 0 otherwise;
t_0 is the time of the last spike emitted by n_i;
$\eta(t-t_0) = -e^{(t-t_0)^m_n}$ is an additional term representing decay in the PSP_i.

As a special case, when all or some of the probability parameters are fixed to "1", the above probabilistic model will be simplified and will resemble the well-known Integrated and Fire Model. A similar formula is used when a LIFM is used as the fundamental model and a time decay parameter is introduced.

3.5 Data Encoding

How exactly the biological neuron encodes and decodes information into conceptual elements is still controversial or according to some authors, unknown [Bishop 1999], [Gerstner 2002]. As the information in SNN is represented in spikes, the information must be encoded in the spike pulse. The theoretical numbers of possible encoding schemes are numerous. Two well-known encoding techniques are the Rank Order Coding (ROC) and the Population Order Coding (POC).

3.5.1 Rank Order Coding (ROC)

ROC encodes information by using the order of the neuron firing times [Thorpe 1998]. For example, for four neurons, where N3 $<$ N1 $<$ N4 $<$ N2, the rank assigned to each neuron is therefore N2=0, N4 = 1, N1 = 2 and N3 =3 as illustrated in Figure 3.9 below. ROC is very effective in modelling audio and visual systems and has been tested in several applications such as visual recognition [Wysoski 2006], audio recognition [Wysoski 2007] and speech recognition [Loiselle 2005].

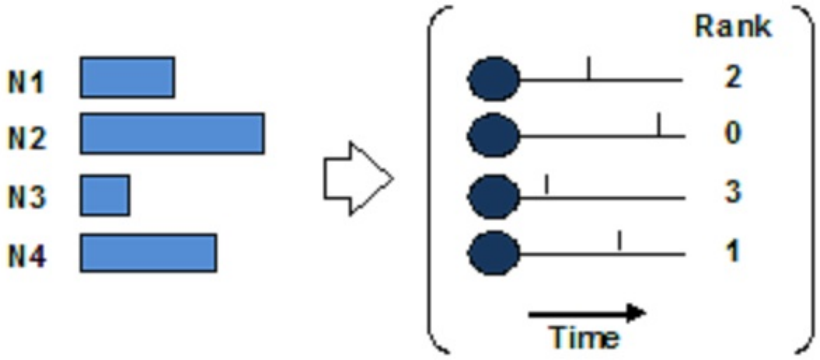

Figure 3.9: Rank Order Coding (ROC) [Thorpe 1998].

3.5.2 Population Rank Order Coding (POC)

Population Rank Order Coding (POC) distributes a single input value to multiple input neurons [Bohte 2002]. The firing time of an input neuron i is calculated using the intersection of a Gaussian function and represents the combination of all relevant input spike firing times. Figure 3.10 depicts the POC encoding process.

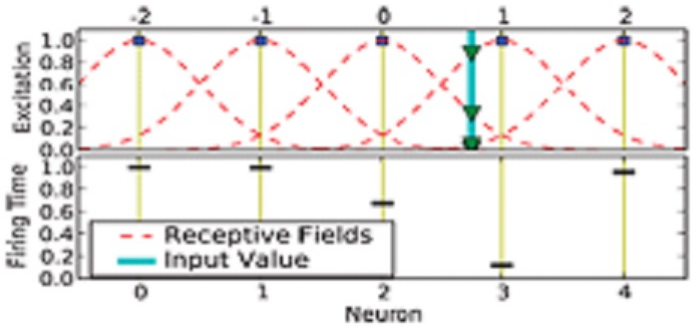

Figure 3.10: Population Order Coding (POC) [Schliebs 2009a].

The center is calculated using Equation 3.18 and the width is computed using Equation 3.19 with the variable interval of $[I_{min}, I_{max}]$. The parameter β controls the width of each Gaussian receptive field.

$$\mu = I_{min} + (2i - 3)/2(I_{max} - I_{min})/(M - 2) \tag{3.18}$$

$$\sigma = 1/\beta(I_{max} - I_{min})/(M - 2) \qquad \text{where} \quad 1 \leq \beta \leq 2 \tag{3.19}$$

3.6 Learning Algorithms

Learning in SNN is a complex process since information is represented in spikes, which is time dependent. There are several learning algorithms designed for SNN.

3.6.1 Spike-Time Dependent Plasticity (STDP)

Spike-time dependent plasticity (STDP) is a form of Hebbian Learning where spike time and transmission are used in order to calculate the output of a neuron. If a pre-synaptic spike arrives at the synapse before the post-synaptic action potential, the synapse is potentiated; if the timing is reversed, the synapse is depressed [Markram 1997], [Bi 2001]. As illustrated in Figure 3.11, if the difference in the spike time between the pre-synaptic and post-synaptic neurons is negative (pre-synaptic neuron spikes first) than the connection weight between the two neurons increases, otherwise it will decrease [Kasabov 2012a]. Through the STDP learning algorithm,

the temporal associations between data are preserved because the connected neuron learns consecutive temporal associations from the data. Long-term potentiation (LTP) can be induced by pre-synaptic activity that precedes post-synaptic activity while reversing this will cause long-term depression (LTD).

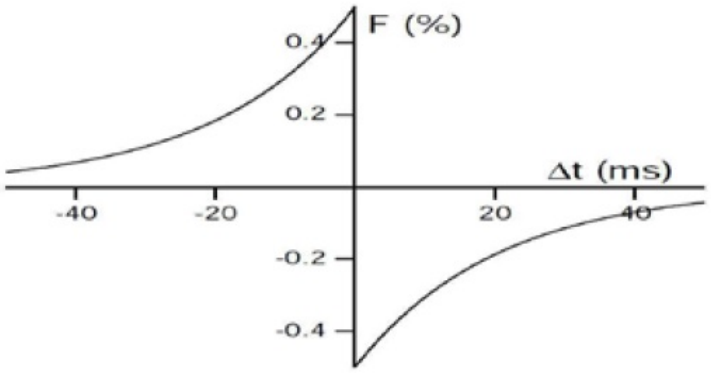

Figure 3.11: Spike-time dependent plasticity (STDP) [Kasabov 2012a].

The mathematical representation of STDP is explained as; the weight change Δw_j of a synapse from a pre-synaptic neuron j is dependent on the relative timing between pre-synaptic spike arrivals and post-synaptic spikes which matches the total weight change is described as [Kempter 1999], [Gerstner 1996]:

$$\Delta w_j = \sum_{f=1}^{N} \sum_{n=1}^{N} W(t_i^n - t_j^f) \tag{3.20}$$

$W(x)$ denotes one of the STDP functions illustrated in Figure 3.12. The pre-synaptic spike arrival times t_j^f at synapse j where $f = 1, 2, 3, ...$ counts the pre-synaptic spikes. The post-synaptic spike arrival times t_i^n at synapse i with $n = 1, 2, 3, ...$ counts the post-synaptic spikes.

An example of STDP function is:

$$W(x) = \begin{cases} A_+ exp(-x/\tau_+) & , \quad \text{for } x > 0 \\ -A_- exp(x/\tau_-) & , \quad \text{for } x < 0 \end{cases} \tag{3.21}$$

The parameters A_+ and A_- may depend on the current value of the synaptic weight w_j. The time constants are on the order of 10 ms for both τ_+ and τ_-.

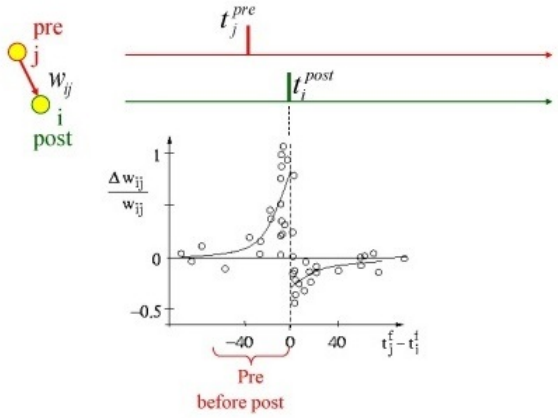

Figure 3.12: The STDP function shows the change of synaptic connections as a function of the relative timing of pre- and post-synaptic spikes after 60 spike pairings [Bi 1998].

3.6.2 Spike-Driven Synaptic Plasticity (SDSP)

Spike-driven synaptic plasticity (SDSP) is a modification of STDP where an unsupervised learning method directs the change of the synaptic plasticity ΔV_{w0} of a synapse w_0 depending on the time of spiking of the pre-synaptic neuron and the post-synaptic neuron. ΔV_{w0} increases or decreases, depending on the relative timing of the pre- and post-synaptic spikes [Kasabov 2012a].

In SDSP, the synaptic efficacy will increase (potentiation), if a pre-synaptic spike arrives at the synaptic terminal before a post-synaptic spike within a critical time window. If the post-synaptic spike is emitted just before the pre-synaptic spike, synaptic efficacy is decreased (depressed). Mathematically the synaptic change can be represented as:

$$\Delta V_{w0} = \begin{cases} \frac{I_{pot}(t_{post})}{C_p}\Delta t_{spk} & , \quad \text{if } t_{pre} < t_{post} \\ \frac{I_{dep}(t_{post})}{C_d}\Delta t_{spk} & , \quad \text{if } t_{post} < t_{pre} \end{cases} \tag{3.22}$$

where Δt_{spk} is the pre- and post-synaptic spike time window.

The SDSP rule can also be implemented as a supervised learning algorithm by entering the desired output spiking sequence along with the training spike pattern as a teacher signal and without changing the weights of the teacher input.

3.6.3 Others types of Learning Algorithm

Other types of learning algorithms are Reinforcement Learning where the learning process is organized by the interaction with the environment, which normally uses in robotic application. ReSuME, following the Hebbian concept is a new supervised learning method for SNN [Ponulak 2005] that was applied in control of movement for the physically disabled. There are also some attempts to copy the Backpropagation learning algorithm which is well-known from the Multilayer Perceptron (MLP). This algorithm, called Spiking Backpropagation Algorithm or SpikeProp [Bohte 2002] is suitable for supervised learning in SNN. The One-Pass Algorithm which is proposed by [Sèguier 2002] is the learning algorithm for eSNN and has been tested by many researchers [Wysoski 2006], [Wysoski 2007], [Soltic 2008], [Schliebs 2009a], [Schliebs 2009b]. Applications of this learning will be discussed further in the next section.

3.7 Working Memory

In cognitive psychology, memory is defined as an organism's ability to store, retain, and recall information. Humans have demonstrated the ability to retrieve information from applied associated stimuli. How the human brain efficiently organizes and stores such vast amounts of information and is able to recall it given partial or incomplete stimuli has brought forth much interest. Both traditional and spatio-temporal connectionist networks represent memory in the form of connection weights. The memories that are represented by the connection weights are often updated after each training cycle and referred to as long term memory as they remain fixed while the network is operating. Other than connection weights, many networks used other network parameters to represent the long term memory. These may be a network connectivity scheme recurrent network, types of transmission delays associated with the connections, or the internal processing elements initial activation values.

Working memory in a network of spiking neurons is presented in terms of firing patterns that are created dependent upon the temporal order of the neurons and previous spike state. Two terms that are used to describe working memory are 'synfire chain' and 'polychronisation'.

3.7.1 Synfire Chain

Synfire chain is a feed-forward network of neurons with many layers (or pools) where each neuron in one pool feeds many excitatory connections to neurons in the next pool. Each neuron in the receiving pool is excited by many neurons in the previous pool and this activity produces a cascade of pools that propagates a synchronous stream of spikes from pool to pool [Abeles 2009]. Figure 3.13 shows the layout of a synfire chain. The order of activation in time is shown by the arrows and the same neuron can be part of more than one pool. Repeated participation of the same neurons can occur up to the limit of the synfire chain's memory capacity.

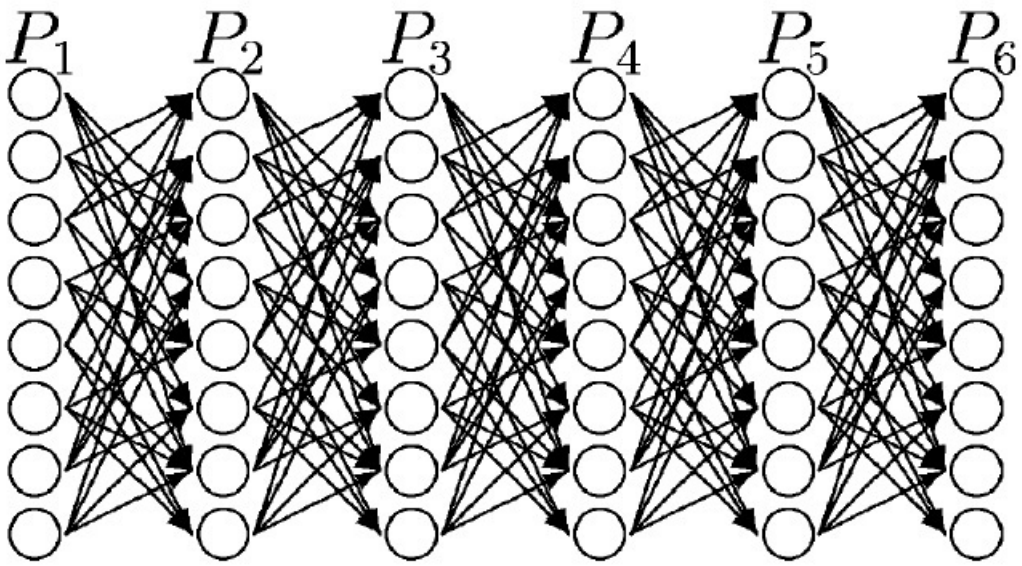

Figure 3.13: Schematic view of a synfire chain: Every neuron in pool i projects to m neurons in pool $i+1$. The width of the chain is the number of neurons in a pool (eight in this example), and the multiplicity (m) of a chain is the average number of cells in pool $Pi+1$ to which a cell in pool Pi is connected (four in this example) [Abeles 2004].

The synfire chains retain the connections of input patterns that result in long term memory. Synfire chains with STDP learning can enhance the memory capacity to process spatio-temporal patterns as demonstrated by [Humble 2012] where multiple connections are created spanning several temporal stages in the chain with regards to the temporal order of the neurons when all pre-synaptic and post-synaptic

connections are considered. Previously learnt individual chains can be combined by repeatedly presenting them in a fixed order.

The synfire chain models have several outstanding properties such as stability over the time span of a few hundreds of milli-seconds, reproducibility, learnability by a self-organisation process, a large storage capacity and ability to build complex representations out of simple parts and to reconfigure the same part in many different ways [Abeles 2004].

3.7.2 Polychronisation

Polychronisation refers to a group of neurons participating in a consistent spatio-temporal firing pattern [Izhikevich 2006]. Synchronisation can be considered as a special case of polychronisation where a group of neurons tend to fire at the same time. Izhikevich in his paper on polychronisation, defined it as reproducible time-locked but not synchronous firing patterns with millisecond precision. Neurons are spontaneously organised into polychronous groups by the STDP changes that select conduction delays to allow the groups to form. As each neuron participates in many groups, with firing one group at one time, the number of coexisting polychronous groups could be far greater than the number of neurons in the network and even greater than the number of synapses. [Szatmáry 2010] has proposed a model where memories are represented by numerous overlapping neuronal groups (polychronous groups) that have time-locked firing patterns called polychronous patterns, depicted in Figure 3.14. Apart from allowing large memory content with actual memories, polychronous neuronal groups (PNG) also allow us to see the internal state/behaviour of a network. Thus removing the'black-box' title that has often represented artificial neural network systems.

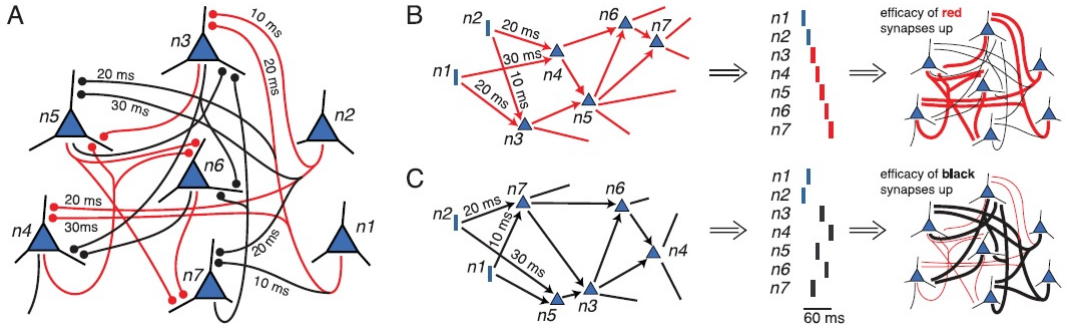

Figure 3.14: Illustration of polychronous neuronal groups and associative short-term plasticity. (A) Synaptic connections between neurons n1, n2, ..., n7 have different axonal conduction delays arranged such that the network forms two functional subnetworks, red and black, corresponding to two distinct PNGs, consisting of the same neurons. Firing of neurons n1 and n2 can trigger the whole red or black PNG. (B) If neuron n1 fires followed by neuron n2 10 ms later, then the spiking activity will start propagating along the red subnetwork, resulting in the precisely timed, i.e., polychronous, firing sequence of neurons n3, n4, n5, n6, n7, and in the short-term potentiation of the red synapses. (C) If neurons n2 and n1 fire in reverse order with the appropriate timings, activity will propagate along the black subnetwork making the same set of neurons fire but in a different order: n7, n5, n3, n6, n4, which temporarily strengthens the black synapses. Readout: post-synaptic neurons that receive weak connections from neurons n3, n4, and n5 with long delays and from neurons n6 and n7 with shorter delays (or, alternatively, briefly excited by the activity of the former and slowly inhibited by the latter) will fire selectively when the red polychronous pattern is activated, and hence could serve as an appropriate readout of the red subnetwork [Szatmáry 2010].

3.8 Reservoir Computing

Mapping SSTD is possible with reservoir computing where input signals are mapped into a higher dimensional space of a dynamic system. The pattern of the input signals can be recognised through the firing of hundreds of thousands of connected neurons. The reservoir concept was first proposed by [Maass 2002] which composed of LIF neurons. When the inputs signals are transferred into high-dimensional

space, they become easily separated. Then the readout function is used to read the states/dynamics of the reservoir for imposing an input-output mapping of the desired class label. The reservoir comprises of a group of recurrently connected neurons. The connectivity of neurons is generally random and the units are non-linear. The activity in the reservoir is stimulated by the input and influenced by past activity. The reservoir approach allows for real-time computation on continuous input streams in parallel. Each neuron is stimulated by time varying inputs from external sources as well as from other neurons. The recurrent connectivity turns the time varying input into a spatio-temporal pattern of activations in the network nodes [Maass 2002]. Based on this fundamental idea, numerous approaches were introduced namely, Liquid-State Machines, Echo State Networks, Backpropagation-Decorrelation and Temporal Recurrent Networks.

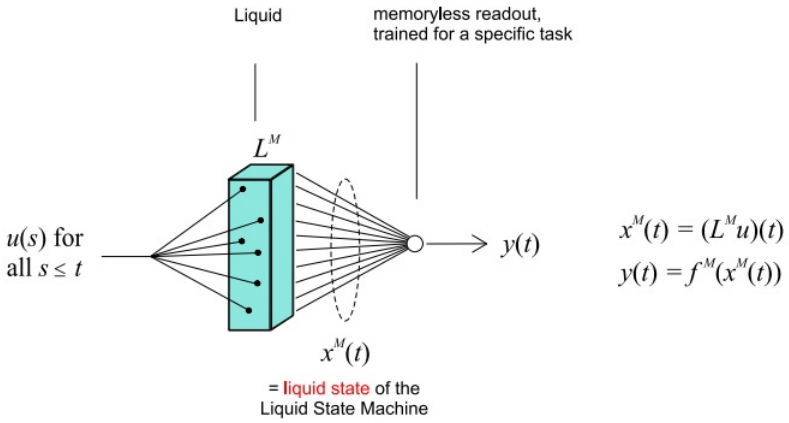

Figure 3.15: Simple liquid state machine structure [Maass 2010].

Figure 3.15 depicts the structure of a Liquid State Machine (LSM), which transforms input streams $u(s)$ into output streams $y(t)$. L^M denotes a Liquid (e.g., some dynamic system), and the "liquid state" $x^M(t) \in R^k$ is the input to the readout at time t. More generally, $x^M(t)$ is that part of the current internal state of the Liquid that is "visible" to an observer. Only one input and output channel are shown for simplicity in Figure 3.15. Advantages of LSM are learning for a single output neuron is fast and cannot get stuck in local minima, and the simplicity of this learning device allows an optimal generalization capability of learned computational operations to new inputs streams [Maass 2010].

3.9 Tools and Applications of Spiking Neural Networks

SNN has been increasingly applied in the field of science and engineering as in other disciplines to solve complicated prediction and classification problems, such as learning rules[Bohte 2002], speech recognition [Verstraeten 2005], [Yau 2007], financial forecasting [Schneider 2008], audio and video analysis [Tsapatsoulis 2007], [Fyfe 2008], associative memory [Knoblauch 2005] and other applications involving biologically realistic controllers for autonomous robots [Floreano 2006], [Wang 2008], robot controller using EEG signals [Pfurtscheller 2006] and many more.

In recent years, SNN is becoming a powerful computational tool for diagnosing and monitoring the prognosis of disease, such as modelling the progression or response to treatment of neurodegenerative diseases, for example Alzheimer's Disease [Schliebs 2005], [Kasabov 2011], prediction of stroke risk [Liang 2014], breast cancer diagnosis [Kiyan 2011], and predicting the risk of death for small-cell lung cancer patients [Bartfay 2006].

Since the introduction of the spiking neuron, there have been several enhancements and variants of the Spiking Neural Network (SNN) such as Evolving Spiking Neural Network (eSNN) [Wysoski 2006], Extended Evolving Spiking Neural Network (eeSNN) [Hamed 2011], Recurrent Spiking Neural Network (reSNN) [Schliebs 2011] and Dynamic Evolving Spiking Neural Network (deSNN) that was developed based on the concept of Evolving Connectionist System (ECOS).

3.9.1 Evolving Connectionist System (ECOS)

Evolving Connectionist System (ECOS) was inspired by the human brains unique learning and abstraction ability [Kasabov 1998a]. ECOS can be described as intelligence, incremental learning and knowledge representation systems that are adaptive and evolving based on continuous incoming data [Kasabov 2009a].

The main principles of ECOS are as follows [Kasabov 1998a]:

1. Represent all information dimensions such as space and time, according to their scales;

2. Evolving structure, growing in open space where the dimension of space can change;

3. Fast learning from a large amount of data (e.g. one pass training);

4. Incremental learning through real time and on-line modes;

5. Facilitated data learning and knowledge representation in a comprehensive and flexible way (e.g. supervised learning, unsupervised learning);

6. Active interaction with other ECOSs and with the environment in a multi-modal fashion;

7. System's self-evaluation in terms of behaviour, global error and success and related knowledge representation.

While the classical ECOS such as EFuNN and DENFIS [Kasabov 2007a] use a simple McCulloch and Pitts model of a neuron, where data is represented as scalars, the further developed evolving spiking neural network (eSNN) architectures use a spiking neuron model, while applying the same or similar ECOS principles. ESNN use data represented as temporal sequences of spikes in a similar mode as information is represented and processed in the brain.

3.9.2 Evolving Spiking Neural Network (eSNN)

Evolving SNN (eSNN) [Wysoski 2006], [Wysoski 2007], [Kasabov 2007a], [Wysoski 2008b], constitute a class of SNN that evolve their structure through incremental, fast one-pass learning utilizing time-to-first spike or similar algorithms [Thorpe 1997], [Bohte 2002], [Sèguier 2002]. The output neuron evolves based on input pattern and weight similarity. During the learning phase, for every new input pattern, the eSNN creates a new class output neuron and calculates its synaptic weights. It then compares its synaptic weights with the other neurons for the same output class. If the difference is lower than a threshold, the new neuron is merged with the closest one belonging to the same output class; otherwise the neuron stays as new neuron. ESNN evolve/develop their structure and functionality in an incremental way from incoming data based on the following principles [Kasabov 2009a]:

1. New spiking neurons are created to accommodate new data, e.g. new output classes, such as faces in a face recognition system.

2. Spiking neurons are merged if they represent the same concept (class) and have similar connection weights.

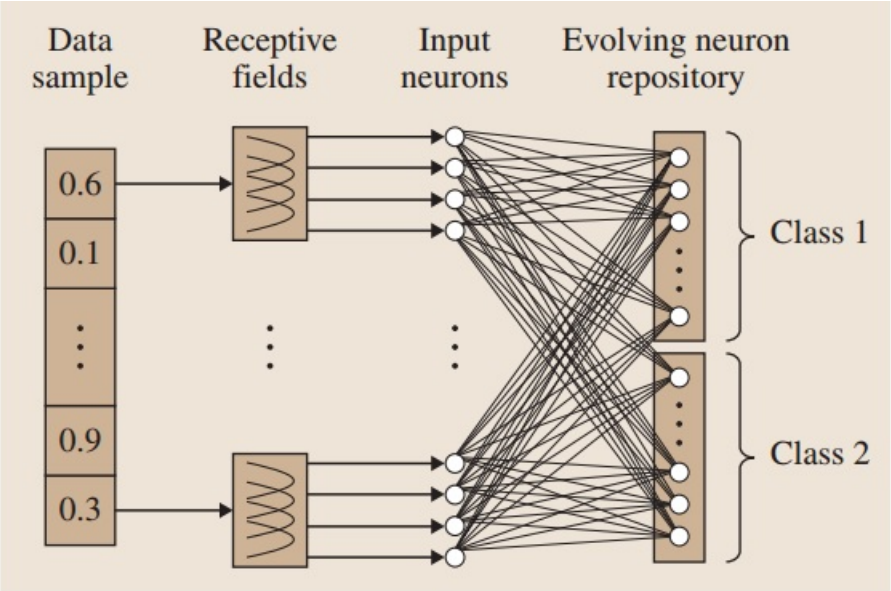

Figure 3.16: Schematic diagram of evolving SNN (eSNN) [Wysoski 2008a].

Figure 4.4 depicts the architecture of eSNN. Continuous-valued vector elements are mapped into the time domain using rank order population encoding based on Gaussian receptive fields. As a consequence of this transformation, input neurons emit spikes at pre-defined firing times invoking the one-pass learning algorithm of the eSNN. The learning iteratively creates repositories of output neurons, one repository for each class. Here a two-class problem is presented. Due to the evolving nature of the network, it is possible to accumulate knowledge as it becomes available, without the requirement of re-training with already learnt samples [Schliebs 2014].

The change in synaptic weight is achieved through a simple rule:

$$\Delta w_{j,i} = mod^{order(j)} \tag{3.23}$$

where $w_{j,i}$ is the weight between neuron j and neuron i, $mod \in (0, 1)$ is the modulation factor, $order(j)$ is the order of arrival to neuron i of a spike produced by neuron j.

For each training sample, we use the *winner-takes-all* approach, where only the neuron that has the highest PSP value has its weights updated.

$$PSP_{max} = \sum w_{j,i} \cdot mod_i^{order(j)} \tag{3.24}$$

The postsynaptic threshold (PSP_{Th}) of a neuron is calculated as a proportion $C \in [0, 1]$ of the maximum post-synaptic potential (PSP) generated with the propagation of the training sample into the updated weights, such that:

$$PSP_{Th} = C \cdot PSP_{max} \tag{3.25}$$

To merge a newly created neuron with an existing neuron, the weights W of the existing neuron n are updated calculating the average as:

$$W = \frac{W_{new} + N_{Frames} W}{I + N_{Frames}} \tag{3.26}$$

where N_{Frames} is the number of frames previously used to update the respective neuron. Similarly, the average is also computed to update the corresponding PSP_{Th}:

$$PSP_{Th} = \frac{PSP_{Thnew} + N_{Frames} PSP_{Th}}{I + N_{Frames}} \tag{3.27}$$

Creating and merging neurons based on localized incoming information and on system's performance are the main operations in the above architectures that make them continuously evolvable.

The pseudo code of the eSNN training algorithm is present below:

Algorithm 3.1 The eSNN training algorithm

Input: Spike trains; SET eSNN parameters (Mod, Sim, C).

Initialise neuron repository R

for *every input samples i belonging to the same output class.* **do**

Create (evolve) a new output neuron and compute the connection weights (refer 3.23).

Calculate PSP_{max} (using formula 3.24).

CalculatePSP_{th} (using formula 3.25).

Calculate the similarity between weight vectors of newly created neuron and existent neurons.

if *similarity* $> PSP_{th}$ **then**

Merge newly created neuron with the most similar neuron (using 3.26 and 3.27).

else

Add the neuron to the output neuron repository R.

end

end

3.9.3 Extended Evolving Spiking Neural Network (eeSNN)

Extended Evolving Spiking Neural Network (eeSNN) is the extended version of eSNN to deal with spatio-temporal problems (STP). This method consists of two layers which are a spatio-temporal reservoir and the eSNN method as classifier. The first layer is designed to capture the whole SSTD pattern that needs to be further classified. This layer utilizes standard eSNN encoding method of Population Coding (POC) (refer to 3.5.2) to turn the input SSTD pattern into a spiking spatio-temporal input pattern. The multi-dimensional input pattern of spikes, also called Pre-synaptic Neurons, is then fed to the second layer for classification using eSNN (refer to 3.9.2) [Hamed 2011]. A schematic illustration of eeSNN is shown in Figure 3.17.

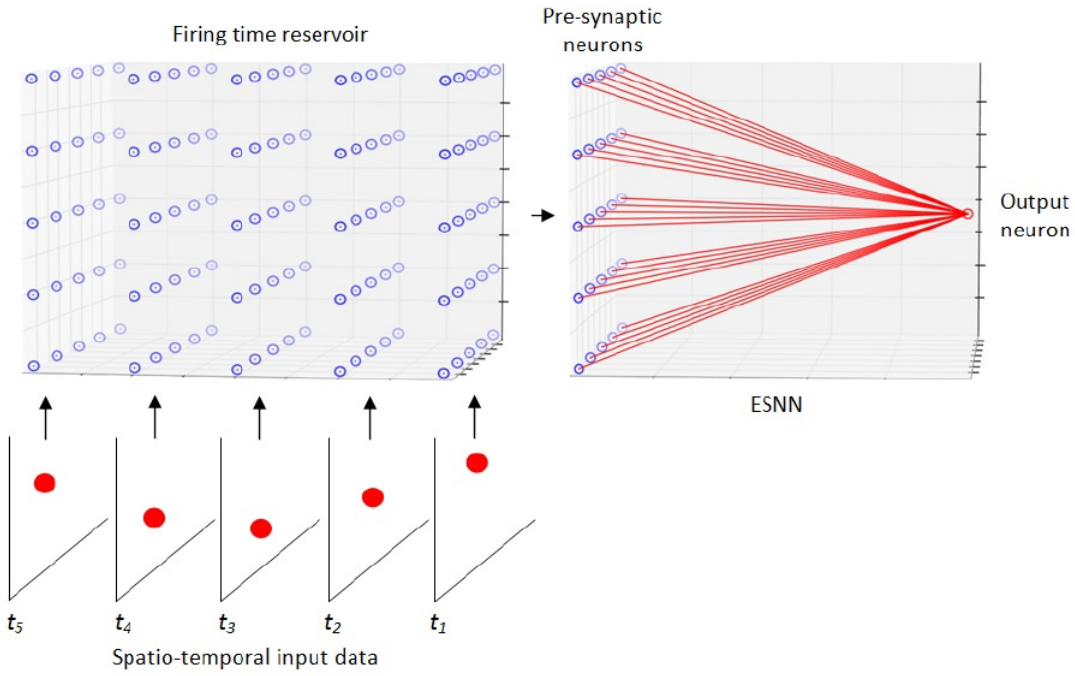

Figure 3.17: Schematic diagram of extended eSNN (eeSNN) [Hamed 2011].

The pseudo code of the eeSNN training algorithm is present below:

Algorithm 3.2 The eeSNN training algorithm

Input: Spike trains; SET eSNN parameters (Mod, Sim, C).

for *all samples in class c.* **do**

  for *all time points* **do**

    Encode every real-value spatial data vector into spike trains (refer 3.5.2).

  end

  Accumulate all spike trains for all time points in a reservoir.

end

Apply spike memory into standard eSNN for a classification task (refer 3.1).

3.9.4 Recurrent Evolving Spiking Neural Network (reSNN)

An extension of eeSNN was introduced by [Schliebs 2011] called Recurrent Evolving Spiking Neural Network (reSNN). This method is based on a simplified integrate-and-fire neural model that mimics the information processing of the human eye. The motivation for this development lies on the functionality of eSNN where it learns the mapping from a single data vector to a specified class label making it

suitable for classification of time-invariant data. Since any real time value will be continuously updated, adding an additional time dimension to the data sets, eSNN alone is insufficient to handle on-line updating.

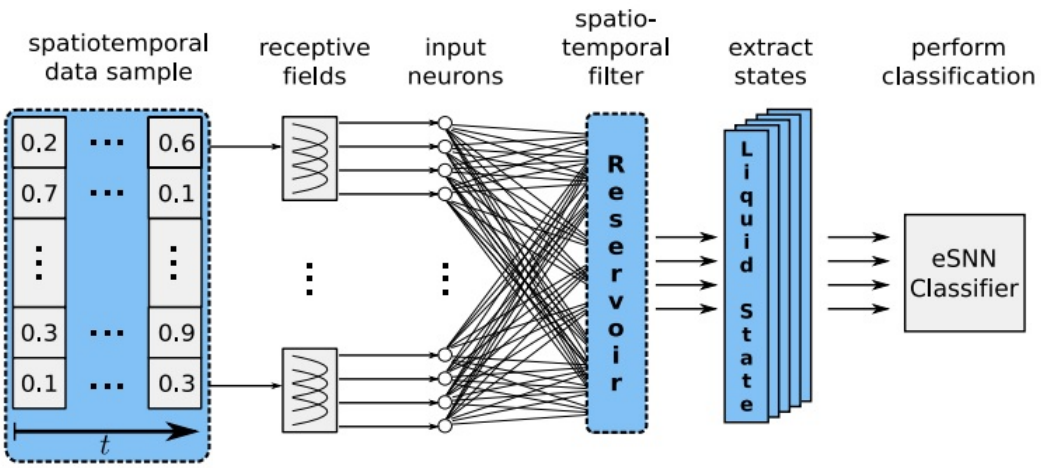

Figure 3.18: Schematic diagram of recurrent eSNN (reSNN) [Schliebs 2011].

reSNN applied the reservoir computing approach of the Liquid State Machine (LCM) [Maass 2002], [Maass 2010] for efficient processing of spatio-temporal data. Figure 3.18 illustrates a schematic diagram of the reSNN architecture. The additional layer exists between the reservoir and classifier represents the LSM. Initially similar to the eeSNN model, each real-value of a spatio-temporal data vector is encoded into trains of spike using the Population Rank Order Coding method. Then a series of spike trains is generated for all pre-synaptic neurons, each represent input neurons within the reservoir. Subsequently the input spike trains are fed continuously into the spatio-temporal filter/reservoir in temporal order. Once all the spike trains have been entered, the reservoir accumulates the temporal information of all input signals into a single high-dimensional intermediate liquid state. The topology and connection weight matrix is fixed during simulation. The reservoir output will be entered into a classifier for classification task where the eSNN acts as a readout function. The one-pass learning algorithm of eSNN is able to learn the mapping of the liquid state machine into a desired class label.

Before the reservoir output can be entered into a classifier, the reservoir responses are transformed into liquid states. Three major types of reservoir output are cluster, frequency and analog. The reSNN adopts the analog readout approach, in which

every spike is convolved (combine (one function or series) with another by forming their convolution) by an α-kernel function according to Equation 3.28:

$$\alpha(t) = e\tau_s^{-1} t e^{\frac{-t}{\tau_s}} \Theta(t) \tag{3.28}$$

where τ_s is the synaptic time constant, and $\Theta(t)$ is the Heaviside function defined as:

$$\Theta(t) = \left\{ \begin{array}{lll} 0 & , & \text{if } t < 0 \\ 1 & , & \text{if } t \geq 0 \end{array} \right\} \tag{3.29}$$

Thus, a convolved spike train $\tilde{s}(t)$ is computed as:

$$\tilde{s}(t) = \sum_{t^f} e\tau^{-1}(t - t^f) e^{-\frac{(t-t^f)}{\tau}} \Theta(t - t^f) \tag{3.30}$$

where the parameter t^f represents the firing time of a neuron.

The neural model in the reservoir is based on the idea of an electrical circuit containing a capacitor with capacitance C and a resistor with a resistance R, where both C and R are assumed to be constant. The dynamics of a neuron i are then described by the following differential equations:

$$\tau_s \frac{dI_i^{syn}}{dt} = -I_i^{syn}(t) \tag{3.31}$$

$$\tau_m \frac{du_i}{dt} = -u_i(t) + RI_i^{syn}(t) \tag{3.32}$$

The constant $\tau_m = RC$ is called the membrane time constant of the neuron. Whenever the membrane potential u_i crosses a threshold ϑ from below, the neuron fires a spike and its potential is reset to a reset potential u_r. The exponential synaptic current I_i^{syn} is used for a neuron i modeled by Equation 3.32 with τ_s being a synaptic time constant.

Below is the pseudo code of the reSNN algorithm.

Algorithm 3.3 The reSNN training algorithm

Input: Spike trains; SET eSNN parameters (Mod, Sim, C)

for *all samples in class c* **do**

  for *all time points* **do**

    Encode every real-value spatial data vector into spike trains (refer 3.5.2).

  end

  Accumulate all spike trains for all time points in a reservoir.

end

for *all spike trains* **do**

  Inject spike trains into the recurrent reservoir.

  Generate reservoir responses based on the neuron spikes.

  Produce the liquid states from reservoir responses.

end

Apply liquid states to classifier/readout function for a classication task (refer 3.1).

Difference between eeSNN and reSNN is the memory structure where reSNN has a more complex reservoir structure comprising an integrated recurrent network and LIF neurons thus requiring more computation time and resources. Even though eeSNN can process data much faster than reSNN, the liquid states output from the reSNN reservoir can be extracted at any time point and passed to the classifier making it more flexible for modelling temporal data.

3.9.5 Dynamic Evolving Spiking Neural Network (deSNN)

The eSNN algorithm recently extended by implementing the version of the STDP (spike-time dependent plasticity) learning rule called SDSP (spike-dependent synaptic plasticity) called dynamic eSNN (deSNN). The deSNN algorithm implements SDSP learning rule where a small drift of a synaptic weight is used to increase the weight if there is a spike, or decrease it if there is no spike, at each of time moment of simulation. The SDSP is an unsupervised learning method, a modification of the STDP (refer to figure 3.12), where the change of synaptic plasticity depends on the relative timing of pre and post synaptic spikes within a critical time window. The

synaptic efficacy can be determined using equation 3.22. Illustrated in Figure 3.19 is an example of using SDSP neurons.

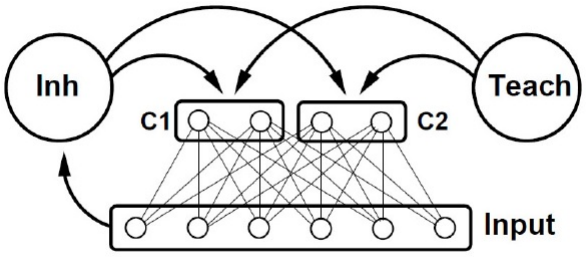

Figure 3.19: An example of using a SDSP neuron [Brader 2007].

Algorithm 3.4 The deSNN training algorithm

Input: Spike trains; SET deSNN parameters (including: Mod, C, Sim and the SDSP parameters)

for *every input spatio-temporal spiking pattern* Pj **do**

Create a new output neuron j for this pattern and calculate the initial values of connection weights $W_{j,i}$ using the RO learning formula 3.23.

Adjust the connection weights $W_{j,i}$ for consecutive spikes on the corresponding synapses using the SDSP learning rule (refer 3.22)

Calculate PSP_{max} using formula 3.24.

Calculate the spiking threshold of the ith neuron (PSP_{th}) using formula 3.25.

if *(the new neuron weight vector* $W_{j,i}$ *is similar in its initial* $W_{j,i}(0)$ **then**

Merge the two neurons (as a partial case only initial or final values of the connection weights can be considered or a weighted sum of them).

else

Add the new neuron to the output neurons repository.

end

end

The deSNN has two variations that differentiate during the recall procedure. These variations are called deSNNs and deSNNm [Dhoble 2012]. In deSNNs, the connection weight of the new input neuron is compared with the connection weight of existing neurons for which the output class is established during training. The closest neuron is the winner and defines the class of the new input pattern. This

means that it takes the whole input pattern of the new neuron and compares it with existing patterns. While in deSNNm, spikes of the new input neuron are propagated as they arrive to all trained neurons and the first one that spikes (its PSP is greater that its threshold using Eq:3.25) defines the class. It uses the parameter C that refers to an earlier percentage of input patterns to classify it. This algorithm considers the first incoming spike pattern as important for classification.The training algorithm of deSNN are presented in Table 3.4 [Kasabov 2013].

3.10 NeuCube for Spatio-temporal Modelling and Pattern Recognition of Brain Signals

The initial development of NeuCube M1 was to support the creation of multi-modular integrated systems, where different modules, consisting of different neuronal types and genetic parameters correspond in a way to different parts of the brain and different functions of interest (e.g.: vision; sensory information processing; sound recognition; motor-control). The whole system works in an integrated mode for brain signal pattern recognition [Kasabov 2012b].

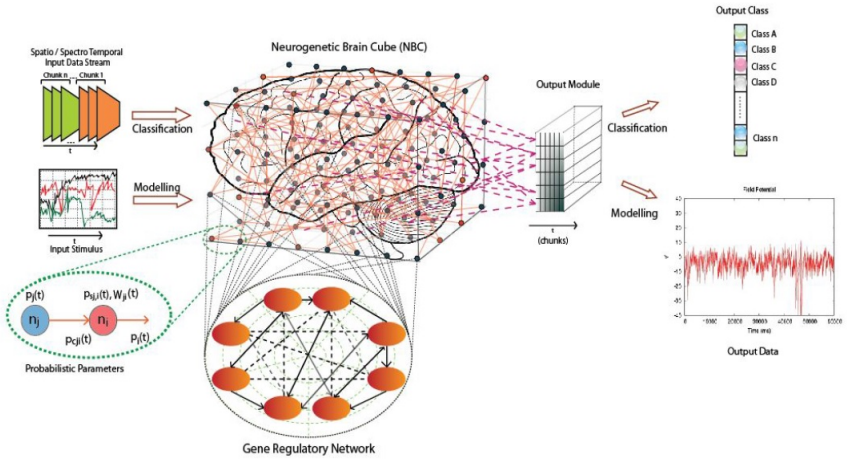

Figure 3.20: A schematic diagram of a NeuCube architecture for brain data modelling [Kasabov 2012b].

The NeuCube architecture as illustrated in Figure 3.20 comprises an input information encoding module; NeuCube network module (a neural network defined in a cubic structure); output module and a gene regulatory network (GRN) module.

This framework transfers brain data such as EEG, FMRI or sound/image data into trains of spikes directly into NeuCube network module.

The NeuCube network module is an approximate map of relevant regions of the brain along with relevant genetic information, represented as a 3D spiking neuronal structure. Figure 3.21 depicts the NeuCube reservoir structure similar to a brain structure after the initialisation process. For example, the original version of the NeuCube structure follows the Talairach coordinate system with only 14 input neurons according to the Emotiv EEG neuroheadset 14 EEG channels. The 14 channels originally follows International 10-20 locations which are AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 and then are mapped into Talairach coordinates of x, y, z and plotted spatially onto specific regions of the NeuCube Brain Architecture (refer to Figure 3.22 (a) and Figure 3.22(b)).

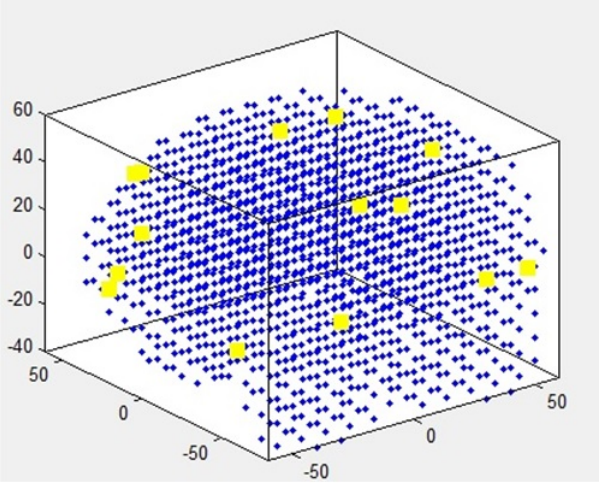

Figure 3.21: NeuCube reservoir after intialisation process.

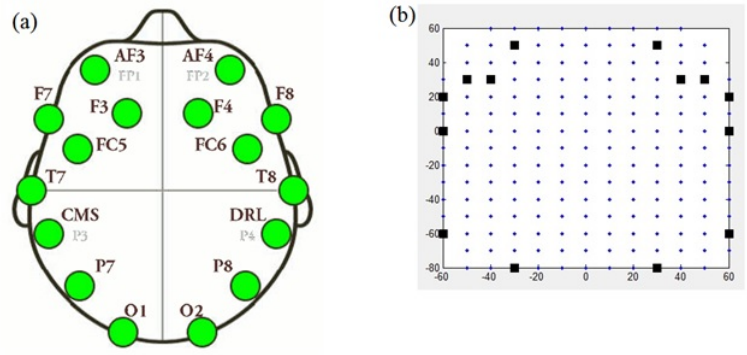

Figure 3.22: (a) Emotiv epoc electrode positions. (b) Neucube input neuron position.

Learning in the NeuCube Architecture is performed in two stages [Kasabov 2012b]:

- Unsupervised training: SSTD is entered into relevant areas of the cube over time and unsupervised learning is used to establish the connection weights. The NeuCube will learn to activate similar spiking trajectories when similar input stimuli are presented - a polychronisation effect [Izhikevich 2006].

- Supervised training of the output module: the same SSTD used for training is now propagated again through the trained NeuCube and output neurons are trained to classify the state of the cube into pre-defined labels (or output spike sequences). As a special case, all neurons from the NeuCube are connected to every output neuron. Feedback connections from output neurons to neurons in the NeuCube can be created for reinforcement learning.

Potential practical applications of this architecture would include: neuro-rehabilitation robots; neuro-prosthetics; control and navigation of wheelchair; cognitive and emotional systems; and serious games.

3.11 Chapter Summary

Earlier SNN models like Spike Response Models (SRM), Leaky Integrate-and-Fire (LIF) models Izhikevich Models, and eSNN are designed for processing static data, which is similar to conventional personalised methods like kNN, wkNN, ww$k$NN. These SNN models have not been successfully applied for a large scale and complex AI problems of classification that combine a string of continuous spatio-temporal patterns and associative memory [Kasabov 2010a]. Since there are some new techniques that have been developed to learn from SSTD pattern e.g. reservoir computing [Maass 2002], personalised modelling frameworks [Kasabov 2010b], Probabilistic Spiking Neuron Model [Kasabov 2010a], that allow the creation of new types of computational models such as Extended Evolving SNN [Hamed 2011], Recurrent ESNN (reSNN) [Schliebs 2011]; Spike Pattern Association Neuron (SPAN) [Mohemmed 2011]; Dynamic Evolving SNN (deSNN) [Dhoble 2012]. The NeuCube Brain architecture opens opportunities for the development of new generations of brain-like intelligent systems for pattern recognition, and for a better understanding

of the spatial-temporal connectivity of other types of SSTD not only brain data. The decision to utilise such models depends in its capability to model stochastic SSTD and the applicability of a number of different implemented algorithms.

The main goal of this study is to further develop a general NeuCube structure and apply a combination of SNN methods that can analyse other types of SSTD and solve real world problems by producing a personalised model for predicting a future event or risk prognosis in either medical or other areas. The applications of the proposed method will be discussed further in Chapter 6 and Chapter 7. While the next chapter we will outline the general framework for spectro, spatio-temporal pattern recognition, followed by the implementation of the framework in developing a practical system.

Chapter 4

PROPOSED NOVEL FRAMEWORK OF EVOLVING SPIKING NEURAL NETWORK METHODS FOR PERSONALISED MODELLING

"Every solution to every problem is simple. It's the distance between the two where the mystery lies."

- Derek Landy

4.1 Introduction

This chapter introduces a novel framework of evolving spiking neural network methods for predictive personalised modelling named PMeSNNr which has been published in the Neurocomputing Journal in 2014. In this chapter the basic component of PMeSNNr will be outlined and discussed in detail. The chapter also states the motivation behind the development of this novel personalised modelling and spiking neural network framework and system. The implementation of PMeSNNr is through the MatLab environment called NeuCube. The system not only exhibits superior diagnostic and prognostic performance, and personalised knowledge; but also discovery and understanding of new knowledge revealing hidden relationships among SSTD. This understanding can be achieved through visualisation of the neuron connections and interactions.

4.2 Motivation

As mentioned previously in Chapter 3, predictive modeling of spectro/spatio-temporal data (SSTD) is a challenging task because it is difficult to model together the time and space components of the data, with their close interaction and interrelationship. In them resides hidden patterns and new undiscovered knowledge that may solve numerous problems. In the health domain, the analysis of SSTD will help enhance the predictive accuracy of diseases such as stroke and heart attack, and aid prevention.

Classical machine learning methods (e.g. SVM, MLP) have limited success in analyzing complex problems with SSTD because their capabilities are limited. Outlined below are several reasons that render classical methods unsuccessful in analyzing SSTD:

1. Conventional machine learning methods are only suitable in classifying vector based and static type of data but not SSTD [Kasabov 2014a].

2. Most conventional machine learning methods apply global modeling techniques. In global modelling approaches, a model is derived from all available data that covers the whole problem space and is represented by a single function, which is then applied to a new individual anytime and anywhere regardless of their differing personal features leading to inaccurate decision. Even though personalised modeling techniques (e.g. kNN, wkNN) could overcome the drawbacks of global modelling, it is still currently only suitable for vector based classification and static data.

3. Conventional methods demand the training samples and testing samples have the same number of input features, which means in order to do early event prediction pre-processing of features needs to be applied in order to keep the features the same length. This type of pre-processing often cause either information loss or introduce residual false information [Tu 2014].

4. SSTD contains noise that may disrupt the analysis process and result in low accuracy so most classical machine learning methods have to remove the noise

by implementing filtering methods (e.g. Signal-to-Noise Ratio). This may result in information loss.

5. There are also conventional SNN models which are commonly used for dealing with temporal information (e.g. Hidden Markov Model [Rabiner 1989], ANN). However these models have limitations in integrating complex and long spatio-temporal components. Usually these models only process or analyses one dimension of the data which makes them incapable of optimally learning from SSTD patterns.

Based on the arguments given above we want to overcome these limitations by proposing the extension of conventional personalised modelling approaches based on a brain-inspired Spiking Neural Networks (SNN) methods, which we believed is capable of analysing personalised temporal data more successfully than classical machine learning methods. The brain-inspired SNN can employ reservoir computing as associative memory functions where trains of spike can be mapped spatially onto neurons and transmit signals via synapse connections. This means both temporal and spatial information can be encoded in an SNN as locations of neurons, and synapses and the time of their spiking activity. As learning is important before making a decision, a brain-inspired SNN can apply learning rules (e.g. STDP, SDSP) to learn spatio-temporal patterns.

4.3 Generic Methodology

Evolving SNN is used to learn from encoded temporal data and its relationships in continuous manner. Based on the pattern learned, the output will be either classified or predicted accordingly. The methodology of personalised modelling based on SNN that was published by our group in Neurocomputing 2014 [Kasabov 2014a] is as follow:

1. For a new individual vector x, described by its value for static variables $V_{s1}(x), V_{s2}(x), \cdots, V_{sm}(x)$, temporal variables $V_{t1}(x), V_{t2}(x), \cdots, V_{tm}(x)$ and unknown output event $E(x)$, select the closest N individuals based on static variables of x that have known outcome and known time of event using a selected distance measurement to create a cluster of neighbourhood Dx.

2. Create an evolving SNN personalised model to predict the event for x by training the model based on the dynamic temporal data of N individuals (Dx) that have been selected in (1). The dynamic temporal variables $V_t(x)$ of new individual x are used to recall the model and predict the outcome for x by calculating the risk and estimating the time when the event likely to happen in the future at the earliest possible time point.

3. Optimise the features, neighbourhood and parameters of the engine to gain optimal model and maximum accuracy at earliest possible time point before the occurrence of the event.

Figure 4.1 depicts the architecture of PMeSNNr. The basic components in the system consist of several functional sub-modules; a spike-time encoding module, a recurrent 3D SNN reservoir, an evolving SNN classifier and a parameter optimization module.

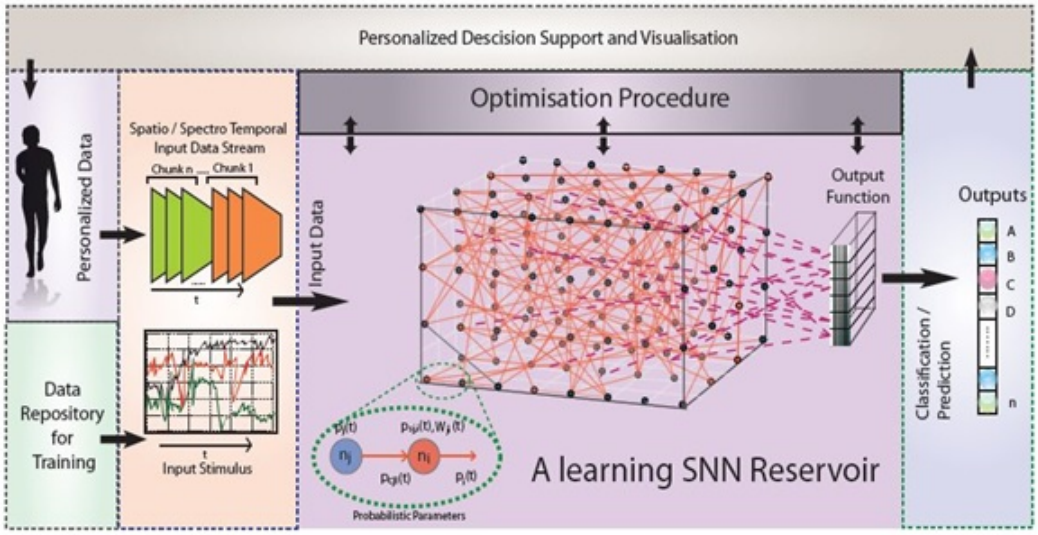

Figure 4.1: Schematic Diagram of the PMeSNNr Framework.

The methodology for creating the personalised model is based on the phases stated in Table 4.1 below:

Table 4.1: The PMeSNNr Methodology

1.	**Encode** the continuous value of data (SSTD) into train of spikes using spike-time encoding module.
2.	**Construct** and **train** a *recurrent 3D SNN reservoir*.
3.	**Construct** and **train** an *evolving SNN classifier* to learn and classify the various input pattern of SSTD from recurrent SNN reservoir trajectories that belong to particular classes.
4.	**Optimise** the model to find the optimum parameters that achieved maximum accuracy at earliest time of prediction through iterative application of step 1 to 3 above.
5.	**Save** the optimised model.
6.	**Recall** the model for new input data.

4.3.1 Input Data Encoding Module

The continuous values of the data (SSTD) are encoded into a train of spikes using the spike-time encoding module. Several types of encoding can be implemented such as the Address Event Representation Method [Culurciello 2001] where the encoding is based on calculating the difference between two consecutive values of the same input variable over time, hence it is suitable when the input data is a stream and only the changes in consecutive values are processed [Kasabov 2014b]. This encoding method was applied successfully to the artificial retina sensor data [Delbruck 2007]. This step is performing a bi-direction thresholding of the signal gradient with respect to time, d/dt. The threshold is self-adaptive and is determined in the following way.

For a signal $f(t)$, the mean m and standard variation s of the gradient df/dt is calculated, then the threshold θ is set to:

$$\theta = m + \alpha s \tag{4.1}$$

where α is a parameter controlling the spiking rate after encoding.

After this, a positive spike train which encodes the increasing place of the signal and a negative spike train which encodes the decreasing place of the signal are obtained. Figure 4.2 illustrated the AER encoding method.

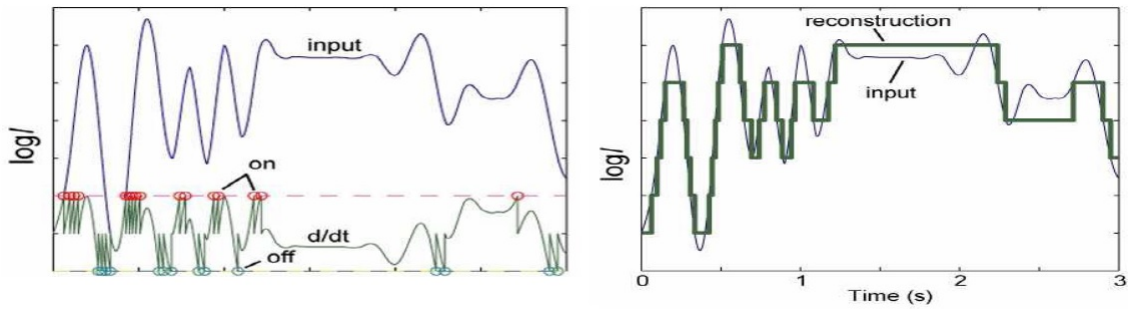

Figure 4.2: Address Event Representation (AER) encoding of continuous time series data into spike trains and consecutive recovery of the signal [Kasabov 2014a].

Other encoding methods are Ben's Spike Algorithm (BSA) [Schrauwen 2003] and Population Rank Order Coding [Thorpe 1998]. BSA encoding scheme has been used for encoding sound data and EEG signals [Nuntalid 2011]. Several advantages of BSA over HSA (Hough Spiker Algorithm) [Hough 1999] are that the frequency and amplitude features are smoother due to the smoother threshold optimization curve, it is less susceptible to changes in the filter and the threshold, and an improvement of 10dB-15dB [Schrauwen 2003],[Nuntalid 2011]. The stimulus is estimated from the spike train by:

$$s_{est} = (h \times x)(t) = \int_{+\infty}^{+\infty} x(t-\tau)h(\tau)d\tau = \sum_{k=1}^{N} h(t-t_k) \tag{4.2}$$

where, t_k represents the neurons firing time, $h(t)$ denotes the linear filters impulse response and $x(t)$ is the spike of the neuron that can be calculated as

$$x(t) = \sum_{k=1}^{N} \delta(t-t_k) \tag{4.3}$$

Figure 4.3 depicts an example of encoding EEG signal into spike train using BSA.

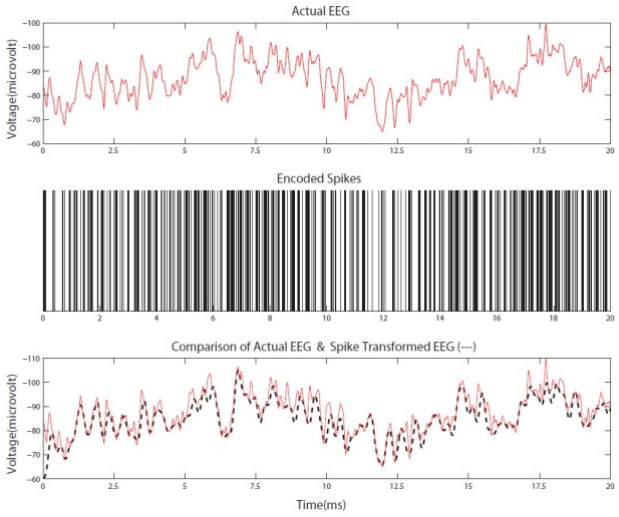

Figure 4.3: The top figure shows a single EEG signal for the duration of 20ms. The middle figure is the spike representation of the EEG signal obtained using BSA. The bottom figure shows the single EEG signal that has been superimposed with another signal (dashed lines) which represents the reconstructed EEG signal from the BSA encoded spikes [Nuntalid 2011].

4.3.2 A SNNr Module

A recurrent 3D SNN reservoir based on the Liquid-State Machine (LSM) concept is constructed and trained, connecting leaky-integrate and fire model (LIFM) spiking neurons with recurrent connections. The learning capability of the reservoir is through the implementation of a learning method called Spike-Time Dependent Plasticity (STDP), a form of Hebbian Learning where spike time and transmission are used in order to calculate the output of a neuron [Markram 1997]. Connected neurons, trained with STDP learning rules, learn consecutive temporal associations from data [Kasabov 2014b]. New connections can be generated based on activity of consecutively spiking neurons. STDP learning is considered a viable learning mechanism for unsupervised learning of SSTD patterns.

During the construction phase of the reservoir, the SNNr connectivity is initialised as small world connections, where nearby neurons are connected with a higher probability. Once spike trains are entered into the SNNr, it acts as a

larger dimensional space. The SNNr accumulates temporal information of all input spike trains and transforms them into high-dimensional intermediate ('liquid') states/trajectories that can be measured over time. The recurrent reservoir generates unique accumulated neuron spike time responses for different classes of input spike trains.

4.3.3 Evolving Output Classification Module

All the patterns created after SNNr training will be entered into an evolving SNN classifier and trained to be to be classified into particular classes. One aspect of originality of the proposed PMeSNNr method is that it utilises the ability of the eSNN to learn to recognise complex spatio-temporal patterns generated in the SNNr before the whole input data pattern is entered. Several evolving SNN classifiers can be implemented such as eSNN [Wysoski 2006], deSNNm or deSNNs where RO learning is used for initialisation of a synaptic weight based on the first incoming spike on this synapse, then this weight is modified based on following spikes using spike time dependent plasticity (STDP) [Dhoble 2012]; SPAN where as a reaction to a recognised input pattern a precise time sequence of spikes is generated at the neuronal output [Mohemmed 2011] and many more.

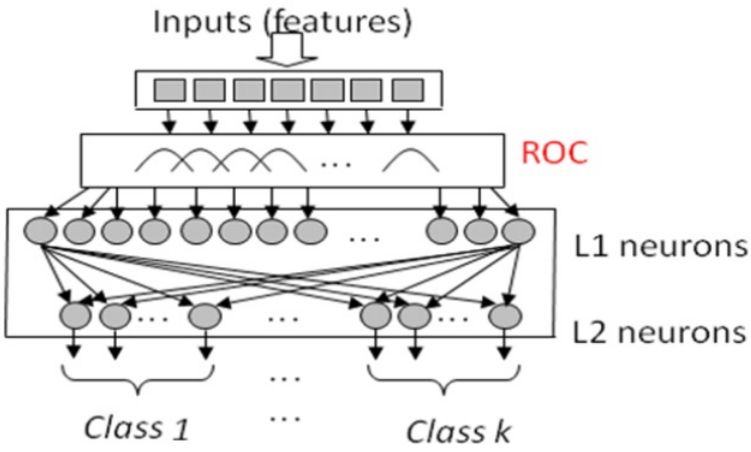

Figure 4.4: eSNN for classification using POC method [Kasabov 2007a].

The Rank Order (RO) learning rule implemented in the eSNN, uses important information from the input spike based on the rank of the incoming spike on each

synapse. It established the priority of inputs based on the order of the spike arrival for a particular pattern. Two advantages of using RO in SNN are fast learning (one pass learning of earlier incoming spike is often sufficient enough to recognise pattern) and asynchronous data-driven processing. Figure 4.4 is an example of integrate-and-fire neuron model implementation with RO learning rule.

ESNN adopts the concept of ECOS where their structure and functionality evolve in an on-line manner from incoming information. For every new input pattern, a new neuron is dynamically allocated and connected to the input neurons. The neuron connections are established using the RO rule for the neuron to recognise this pattern (or a similar one) as a positive example. The new neurons represent centers of clusters in the space of the synaptic weights. In some system implementations similar neurons are merged based on Euclidean distance between them [Wysoski 2010], [Kasabov 2007a]. That makes it possible to achieve very fast learning in an eSNN, both in a supervised and an unsupervised mode. The postsynaptic potential of a neuron i at a time t is calculated as:

$$PSP(i,t) = \sum mod^{order(j)} W_{j,i} \tag{4.4}$$

where:

mod is a modulation factor;

j is the index for the incoming spike at synapse j, i ;

$w_{j,i}$ is the corresponding synaptic weight;

$order(j)$ represent the order (rank) of the spike at the synapse j, i among all spikes arriving from all m synapses to the neuron i.

The $order(j)$ has the value 0 for the first spike and increases according to the input spike order. An output spike is generated by neuron i if the $PSP(i;t)$ becomes higher than a threshold $PSP_{th}(i)$. This is calculated dependent on the value of the parameter proporation factor (C), a fraction of $PSP(i,t)$. The values of parameter C lies between 0 - 1, where if the value is 0.7 the system should be able to flag a potential event before the exact event by learning 70% of the spike patterns.

The modulation factor reflects the connection weight based on the time-to-first spike algorithm; which means an earlier spike received by a neuron carries a stronger

weighted than later ones when the modulation factor is higher. The value should be between 0 - 1.

During the training process, for each training input pattern (sample, example) a new output neuron is created and the connection weights are calculated based on the order of the incoming spikes.

In the eSNN, the connection weights of online created connections between a neuron n_i, representing an input pattern of a known class, and an activated input (feature) neuron n_j, are established using the RO rule:

$$\Delta W_{j,i} = mod^{order(j,i(t))} \tag{4.5}$$

Once a synaptic weight $W_{j,i}$ is initialised based on the first spike at synapse j, the synapse becomes dynamic and adjusts its weight through the SDSP algorithm. It increases in value by a small positive value (positive drift parameter) at any time t when a new spike arrives at this synapse, and decreases its value (a negative drift parameter) if there is no spike at this time.

$$\Delta W_{j,i}(t) = e_j \cdot D \tag{4.6}$$

where $e_j(t) = 1$ if there is a consecutive spike at synapse j at time t during the presentation of the learned pattern by the output neuron i and -1 otherwise. In general, the drift parameter D can be different for 'up' and 'down' drifts. Drift is the negative influence on the connection weight when no spiking behavior appears within a certain time frame (constant leaking). The value is between 0 - 1, but it should be lower than the value of the modulation factor.

After the whole input pattern (example) is presented, the threshold of the neuron n_i is defined to make this neuron spike when this or a similar spatio-temporal pattern is presented again in recall mode. The threshold is calculated as a fraction (C) of the total PSP, calculated as:

$$PSP_{max} = \sum_{j=1}^{m} \sum_{t=1}^{T} (mod^{order(j,i(t))} W_{j,i(t)}) \tag{4.7}$$

$$PSP_{th} = C \cdot PSP_{max} \tag{4.8}$$

The calculation of threshold only involve several type of classifier which is eSNN, eeSNN and deSNNm while no threshold value is calculated during recall of deSNNs classifier because system will calculate the connection weight for the whole new input pattern and compare it to the existing pattern.

4.3.4 Parameter Optimisation Module

The model is optimised to find the parameter settings that achieved maximum accuracy at the earliest time of prediction. Optimisation methods that can be implemented include Grid Optimisation, Genetic Algorithm, Gravitational Search Algorithm, Particle Swarm Algorithm and many more. Currently, only the Grid Optimisation method has been implemented where the system iteratively performs an exhaustive search to find optimum parameters values.

Grid search is a straightforward and effective method to tune parameters. Suppose there are P parameters that have to be optimized simultaneously. For each parameter there are three hyperparameters to be specified manually: the minimal value m and the maximal value M of the searching interval, and the searching step size s. Given these three hyperparameters, we first create a P-dimensional matrix where each dimension corresponds to a optimizing parameter from m to M divided into $(M..m) = s$ entries. In this case, each entry of the matrix corresponds to a group of values of the optimising parameters. The drawbacks of this method are slow computation time and high resources utilisation.

4.4 Extended Dynamic Evolving Spiking Neural Networks

This section introduces a novel algorithm combining deSNNs and wkNN for Multi-NN classification and regression problems. During the recall procedure, deSNNs will directly compare a new input neuron connection weight with connection weight of a trained neuron and the closest neuron will define the class of the new input neuron (refer to Figure 4.5); this process is call 1-NN classification.

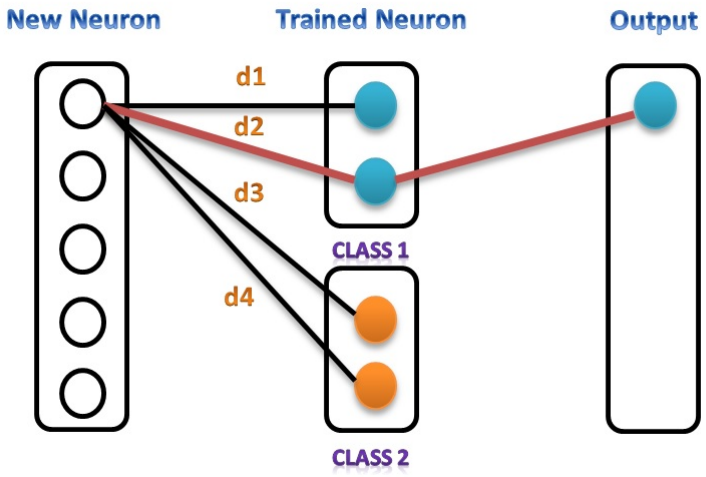

Figure 4.5: An example of 1-NN classification problem.

Within a new deSNNs_wkNN algorithm, during the recall procedure instead of directly comparing the closest distances to calculate the output value, another layer of filtering is added by selecting several closest trained neurons depending on the nominated k value. For example if $k = 3$, the closest 3 trained neurons will be selected (refer to Figure 4.6). A weight value is assigned to the neurons based on the distance of the new input neuron and existing neurons. The closest neurons will be weighted the strongest.

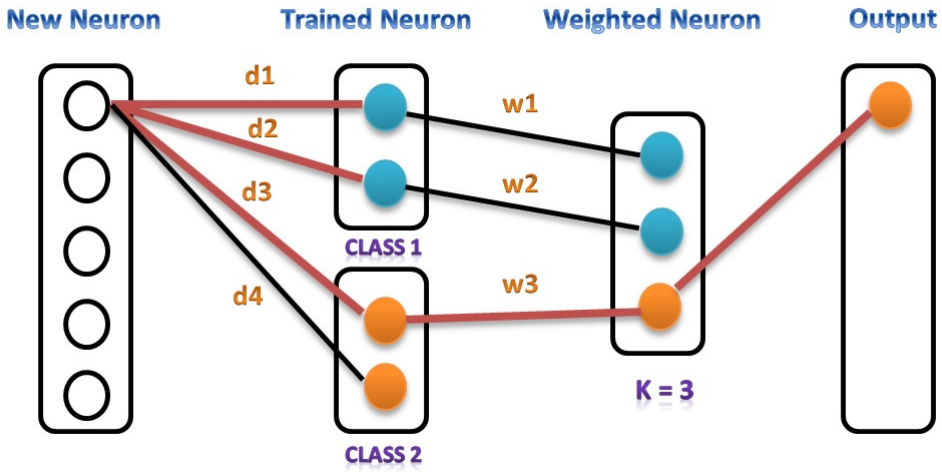

Figure 4.6: An example of Multi-NN classification problem.

Mathematical representation of wkNN is based on the following equation:

$$y_i = \sum_{j=1}^{N_i} \frac{w_j y_j}{w_j} \tag{4.9}$$

where (y_i) is the output value of a new input vector that depends on the output values of k-nearest neighbour vectors (y_j) and the distance between the existing vectors and the new input vector which is represented as a weight vector (w_j). Weight (w_j) is calculated based on the distance of k-nearest neighbour vectors to the new vector using equation:

$$w_j = [max(d) - (d_j - min(d))]/max(d) \tag{4.10}$$

The vector $d = [d_1, d_2, ...d_{N_i}]$ is defined as the distances between input vector (x_i) and (N_i) nearest neighbour (x_1, y_1) for $j = 1$ to N_i. The Euclidean distance measured between new vector (x_i) and neighbouring vector (x_j) is calculated based on:

$$d_j = sqrt[\sum_{l=1}^{V}(x_{i,l} - x_{j,l})^2] \tag{4.11}$$

where V is the number of the input variables, $x_{i,l}$ and $x_{j,l}$ are the values of the variables in vector x_i and x_j, respectively. An example of the wkNN implementation in a classification problem that consists of two classes, represented by 0 (class 1) and 1 (class 2) as output class labels is as follows. If the new vector (x_1) belongs to class 2, this means it has "personalised probability". To classify the new vector (x_1) into classes, there has to be a probability threshold selected P_{thr}, so if the output value $y_i \geq P_{thr}$ then the new vector (x_1) will be classified into class 2. For example the probability threshold value is set to 0.5 and if the output value is 0.75 which is more than the probablity threshold, the new vector will be classified into class 2 not class 1 where the output value should fall within the range of $0 \leq y_i \leq 0.5$. While for regression the output value (y_i) calculated will be assigned to the new input vector value. Algorithm 4.1 outlines the training procedure and Algorithm 4.2 the recall procedure of deSNNs_wkNN.

This proposed algorithm is implemented as a new functional module in the MatLab based system called NeuCube M1 for prototyping and testing case studies re-

lated to classification and regression problems that will be discussed further in next chapter.

Algorithm 4.1 The deSNNs_wkNN training algorithm

Input: Spike trains; SET deSNNs_wkNN parameters (mod, D, k, P_{thr})

for *every input STP* x_j *represent as output spike pattern for training* **do**

Create a new output neuron j for this pattern and calculate the initial values of connections weights using RO learning rule (refer 4.5).

Adjust the connection weights w_j for consecutives spikes on the corresponding synapses using SDSP learning rules (refer 4.6).

Calculate PSP_{max} using Formula 4.7.

Add new neuron to the output neurons repository.

end

Algorithm 4.2 The deSNNs_wkNN recall algorithm

Input: Spike trains; SET deSNNs_wkNN parameters (mod, drift (D), k, P_{thr}).

for *every input STP* x_i *represent as output spike pattern for validation* **do**

Create a new output neuron x_i for this pattern and calculate the initial values of connections weights using RO learning rule (refer 4.5).

Adjust the connection weights w_i for consecutives spikes on the corresponding synapses using SDSP learning rules (refer 4.6).

Calculate PSP_{max} using Formula 4.7.

Calculate distance d_j from connection weight of new neuron x_i to trained neuron x_j using Euclidean distance calculation 4.11.

Select the nearest neurons x_j based on k parameter.

Find d_{max} and d_{min}.

Calculate w_j weight to assign to the each distance, using Formula 4.10.

Calculate the output value y_i using Formula 4.9.

if *classification problem* **then**

classify the input vector x_i based on P_{thr}.

else

assign the output value y_i to predicted value for input vector x_i.

end

end

4.5 Chapter Summary

This chapter proposed a new type of evolving connectionist systems (ECOS) called PMeSNNr based on brain-like information processing principles that are designed to deal with large and fast spatio/spectro temporal data using Spiking Neural Network (SNN). The framework integrates novel machine learning and modelling techniques for a specific research problem such as predictive personalised modelling, SSTD pattern learning, classification, regression, early event prediction and much more.

The novelty of the proposed method are many which it is the first personalised modelling method developed to deal with SSTD without defining in advance the 'time lags' of the time series data, using the same paradigm of spiking information processing of the brain for spatio-temporal pattern recognition task and to process SSTD in the form of spiking neural network, the evolving nature of the system where unknown classes could be added incrementally as a result of new SSTD patterns being learned and recognised, which is also a principle of brain cognitive development and the ability to recognise and predict the outcome of a new SSTD pattern that is similar to previously learnt ones before the new pattern is fully presented to its inputs.

The next chapter will discussed the proposed NeuCube M1 architecture that is developed based on PMeSNNr framework.

Chapter 5

A METHOD FOR PREDICTIVE DATA MODELLING IN NEUCUBE: HOW EARLY AND HOW ACCURATE

"Study the past if you would define the future."

- Confucius

5.1 Introduction

The NeuCube architecture was initially introduced by Nikola Kasabov [Kasabov 2014b] for mapping, learning and understanding spatio-temporal brain data. In order to capture the time and space characteristic of any type of SSTD, a more generalised NeuCube was developed following the framework of PMeSNNr. Since then NeuCube has evolved into an architecture called Neuro-computer for spatio and spectro-temporal data. Figure 5.1 depicts the NeuCube Architecture.

The first phase of NeuCube was the development of Module 1 for the purpose of prototyping and testing. The system was applied to the stroke occurrences case study [Othman 2014] and aphid case study [Tu 2014]. The NeuCube system architecture was also published in aphid case study paper [Tu 2014].

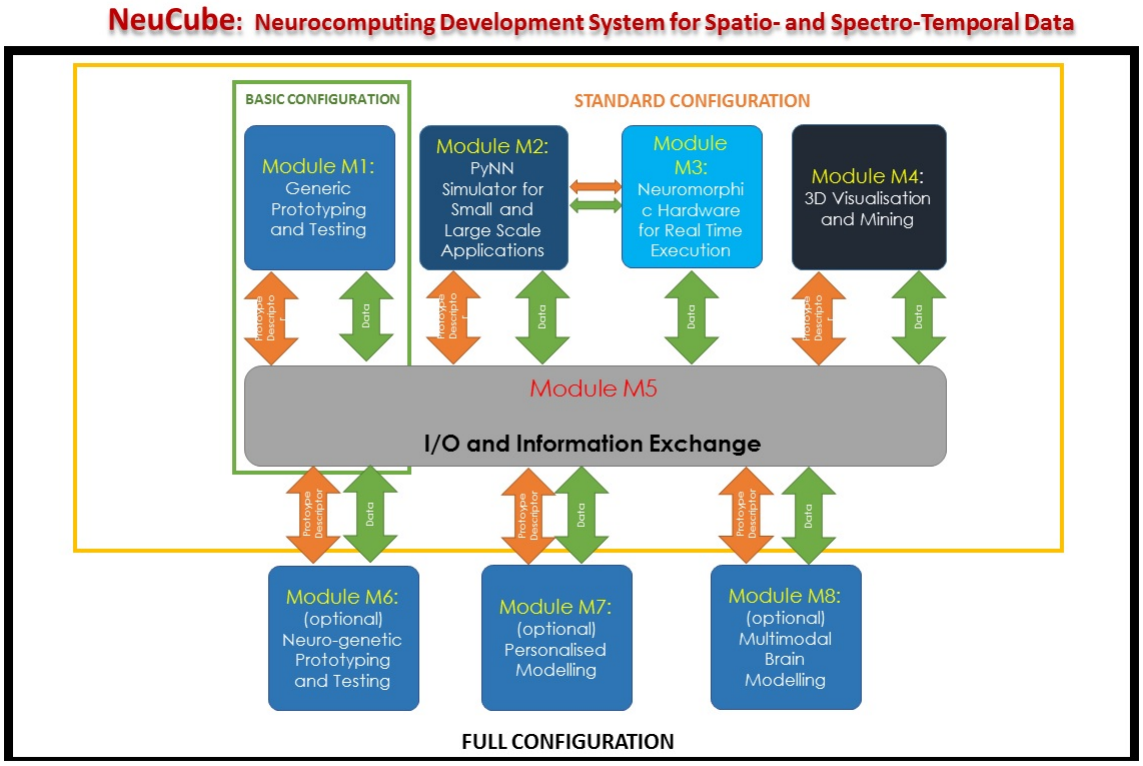

Figure 5.1: NeuCube Functional Diagram (http://www.kedri.aut.ac.nz/)

5.2 NeuCube M1 Architecture

The main parts of NeuCube are a three-dimensional spiking neural network reservoir (SNNr) and an evolving SNN classifier. A block diagram of the architecture is shown in Figure 5.2. It contains three parts: an input encoding module, a three-dimension SNNr and an output dynamic evolving spike neural network (deSNN) classifier.

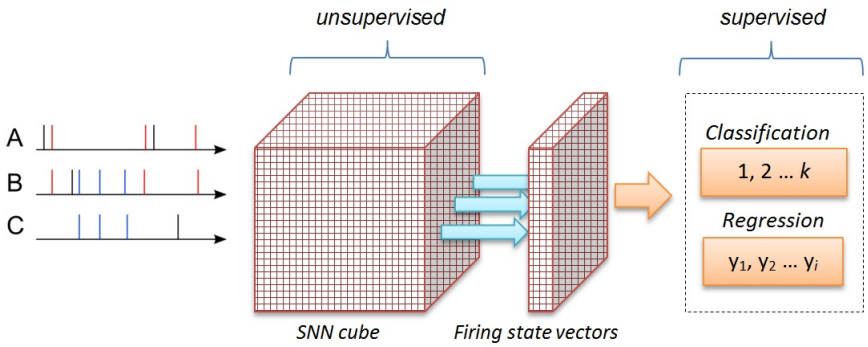

Figure 5.2: Simple NeuCube M1 Architecture.

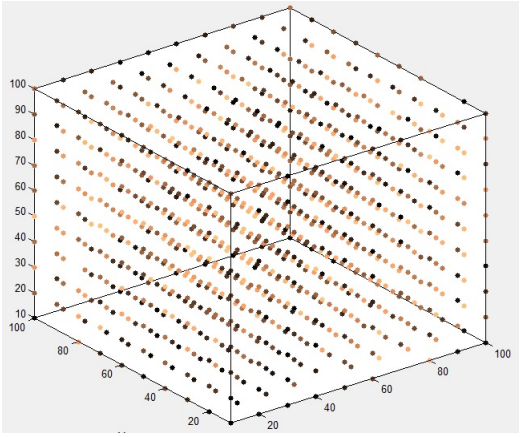

Figure 5.3: An example of 3D recurrent SNN reservoir with 1000 neurons.

The size of the SNNr is controlled by three parameters, n_x, n_y and n_z representing the neuron numbers along x, y and z axes. The total neuron number in the reservoir is $N = n_x \times n_y \times n_z$. Figure 5.3 shows an example of a $10 \times 10 \times 10$ reservoir resulting in 1000 neurons. The size of the SNNr can vary depending on the prediction task and the data. There are no specific gulidelines to determine the size of the cube, but theoretically the size can be determined through the understanding of the data and input neuron count representing each feature in the data. For example if the data has 10 features/input neuron, the size of the cube could be smaller compared to FMRI data which has more than 5000 input neurons which needs a bigger cube of neurons to learn both connections and relationships.

The NeuCube is trained in a two-stage training process. The first stage is unsupervised learning that makes the SNNr learn spatio-temporal relations from the input data by adjusting the connection weights in the SNNr. The second stage is supervised learning that aims at learning the class information associated with each training spatio-temporal sample. The modelling process contains five phases: data encoding, reservoir initialization and unsupervised training of the reservoir, supervised training of the classifier, and validation. The process reflects the PMeSNNr framework described in earlier sections.

- Data encoding - After uploading the data, it will be encoded into spike trains using AER encoding method (refer to section 4.3.1). These spikes trains are then feed into the NeuCube reservoir and these spike trains are mapped onto designated input neurons in the cube. The input mapping will be further

explained in the next section.

- Reservoir initialisation - the SNN reservoir is initialised following the small world connection rule: each neuron in the reservoir is connected to its nearby neurons which are within a distance d, where d equals to the longest distance between any pair of neurons in the reservoir multiplied by a parameter r. The initial weights of the connections are set to the product of a random number in $[-1, 1]$ and the inverse of the distance between them. The connections weights are randomly selected where 80% of the connection weights are set as positive and 20% as negative. For the input neurons, the connection weights between it and other neurons are doubled in order to emphasize its significance in the reservoir. This is currently the default setting for cube initialisation where 80% are excitatory connections and 20% are inhibitory connections to follow the assumption that more excitatory connections can enhance the relationship rather than deplete it.

- Unsupervised Training - the unsupervised learning stage is intended to encode 'hidden' spatio-temporal relationships from the input data into neuronal connection weights. According to the Hebbian learning rule (refer to section 4.3.2), if the interaction between two neurons is persistent, then the connection between them will be strengthened. Using the STDP learning rule ensures that the time difference in the input spiking trains, which encodes the temporal patterns in the original input signals, will be captured by the neuron firing state and the non-symmetrical connection weights in the reservoir.

- Supervised Training and Validation - once the NeuCube is trained, all connection weights in the reservoir and in the output classification layer are established, but they can change based on further training (adaptation) because of the evolvable characteristic of the architecture. For a given new sample without any class label information, the trained NeuCube can be used to predict its class label (refer to section 4.3.3 and section 4.4).

Figure 5.4(a) shows an example of the connectivity of a SNNr before training and Figure 5.4(b) shows the connectivity of a SNNr after training on environmental SSTD. It can be seen that as a result of training new connections have been created

that represent spatio-temporal interaction between input variables captured in the SNNr from the data. The connectivity can be dynamically visualised for every new pattern submitted.

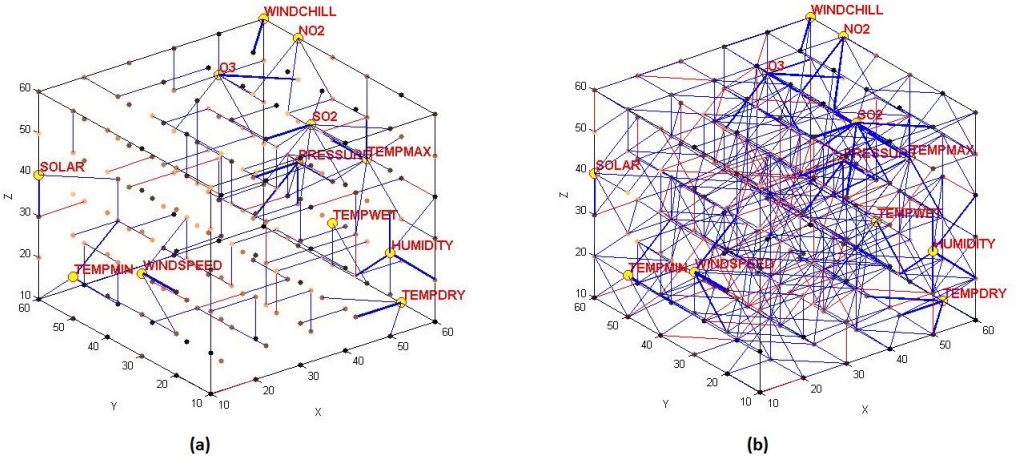

Figure 5.4: (a) SNNr connectivity during intialisation stage where blue is positive connections and red is negative connections (b) SNNr connectivity after training.

5.3 Predictive Modelling

Early event prediction is very crucial when solving important ecological and social tasks described by temporal-or/and spatio-temporal data; such as pest population burst prevention, natural disaster warning, and financial crisis prediction. The generic task is to predict early and accurately whether an event will occur in a future time based on already observed SSTD.

Two approaches can be implemented for predictive modelling, where the first approach is through 'synfire chain' connection path ways (refer to section 3.7) and the second approach is through appropriate optimisation of parameter C in eSNN (deSNN) classification procedure (refer to section 3.9.5) .

In NeuCube M1 similar connectivity and activation patterns (called 'polychronous waves') can be generated in the reservoir with recurrent connections to represent short term memory. The 'polychronous waves' are generated as the result of random excitation and a learning rule that depends on pre- and post-synaptic spike-times (STDP) [Izhikevich 2006]. Using STDP learning, connection weights change to form

Long-term Potentiation or Long-term Depression, which constitute long-term memory.

Results of the use of the NeuCube suggest that the NeuCube M1 architecture can be explored for learning long spatio-temporal patterns and to be used as associative memory. Once data is learned, the SNNr retains the connections as long-term memory. Since the SNNr learns functional pathways of spiking activities represented as 4 structural pathways of connections, when only a small initial part of the input data is entered the SNNr will 'synfire chain' learned connection pathways to reproduce learned functional pathways. Thus NeuCube M1 can be used as an associative memory and as a predictive system for event prediction when only some initial new data is presented.

To implement the 'synfire chain' approach the time length of the training data (samples, collected in the past) and the test data (samples used for prediction) can be different, as illustrated in Figure 5.5. NeuCube M1 can work on both even-feature-length data sets and uneven-feature-length data sets. To better demonstrate the 'synfire chain' approach preliminary experiment will be carry out in the next section. To further demonstrate the validity of the NeuCube M1 case studies on two real world data sets is presented in Chapter 6 and 7.

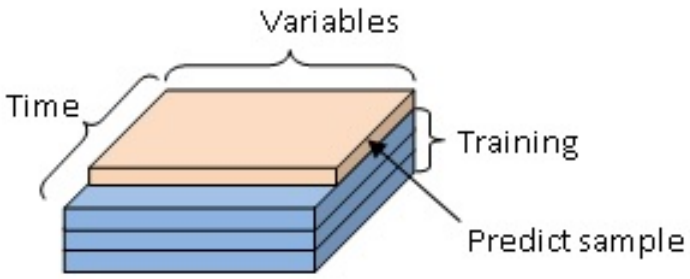

Figure 5.5: A spatio-temporal data model used for early event prediction.

The second approach is through parameter C in eSNN and deSNNm (refer to section 3.9.2). The parameter C, used to calculate the threshold of a neuron i, makes it possible for the neuron i to emit an output spike before the presentation of the whole learned pattern (lasting T time units) which the neuron was initially trained to respond to. This is an important property of eSNN that can be utilized to train an eSNN on whole temporal input patterns and to recall the eSNN on a

partial input pattern, e.g. only initial input data, so that the eSNN can predict an outcome earlier.

5.3.1 Preliminary Experiment

The capability of predictive modelling in NeuCube M1 is demonstrated through an experiment on synthetic data. To implement predictive modelling, the 'synfire chain' approach is applied to learn connection pathways and reproduce learned functional pathways. As described above, in the NeuCube reservoir 'polychronous waves' are generated with recurrent connections and a learning rule that depends on pre- and post-synaptic spike-times (STDP). The generation of 'polychronous waves' activate 'synfire chain' effect where the cube retains the connections of input patterns, resulting in a long term memory effect. Therefore it is possible for a NeuCube reservoir to be trained on the whole time length and recall (predict) only on a partial time length. When the connection weights and activation patterns of whole trained neurons are learned, then knowledge are stored as long term memory. During recall, this long term memory can reproduce learned functional pathways when just a small initial part of the validation data is entered into the spiking neural network reservoir.

Synthetic data is used in testing and creating many different types of systems. Here the synthetic data is used as a simulation of temporal data that has similar behavior of the authentic data that is used in the real world case studies in Chapter 6 and 7. Each feature is described by a different value of randomised components. The product R describes the relationship of different functions which reflect the complexity of the relationship and known outcome. The synthetic data comprised 21 samples described by six (6) variables in 31 time points. The samples are classified into two (2) classes, labelled Class 1 and Class 2. The variables are created in the following way:

- $Var1, Var2, Var3, Var4$ is a random number [1-200]
- $Var5$ is derived from $Var4$ where $Var5 = Var4/10$
- $Var6$ is a random number [1-100]

Mathematical representation of the data is as follows:

$$R = \frac{\sum_n V_n^1 \cdot \log(\sum_m V_m^2)}{\sum_o V_o^3 \cdot (\sin(\sum_p V_p^4) \cdot \sum_q V_q^5} + \sum_r V_r^6 \tag{5.1}$$

$$Class = \left\{ \begin{array}{lll} R > 0 & , & 1 \\ R \leq 0 & , & 2 \end{array} \right\}. \tag{5.2}$$

where $R > 0$, this pattern recognised as Class 1, otherwise it is Class 2 (refer 5.2) .

The experiment is designed to find the earliest time point to accurately predict future events. As depicted in Figure 5.6 the first experiment will take the full time length of the variables for both testing and training. The predictive ability of NeuCube is justified by training the full time length variables (100%) and validating/predicting 80% of the time length (7 time points earlier of the time length). This method preserved the temporal relationship residing between the input features where each time period describes all features related to each time point.

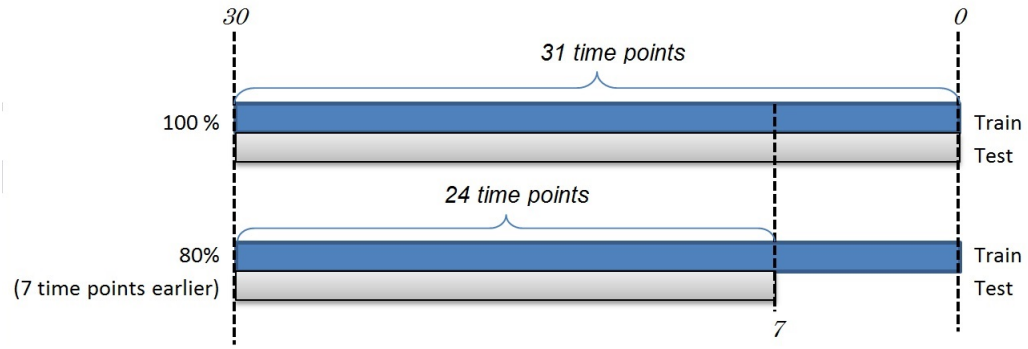

Figure 5.6: Experimental design for NeuCube M1 (synthetic data).

The recurrent 3D reservoir is comprised of 216 neurons (6x6x6). All data was presented for both training and testing through random sub-sampling validation technique where 50% of the data is selected for training and 50% for testing. The parameters for each case study were optimised through a grid search optimisation procedure in NeuCube. After attaining optimised parameter settings, the case studies are trained and recalled recurrently until NeuCube obtained the best model. The classifier selected for this experiment was deSNNs_wknn where the parameter for mod is 0.04, drift is 0.25, k is 3 and Probability threshold is 0.5.

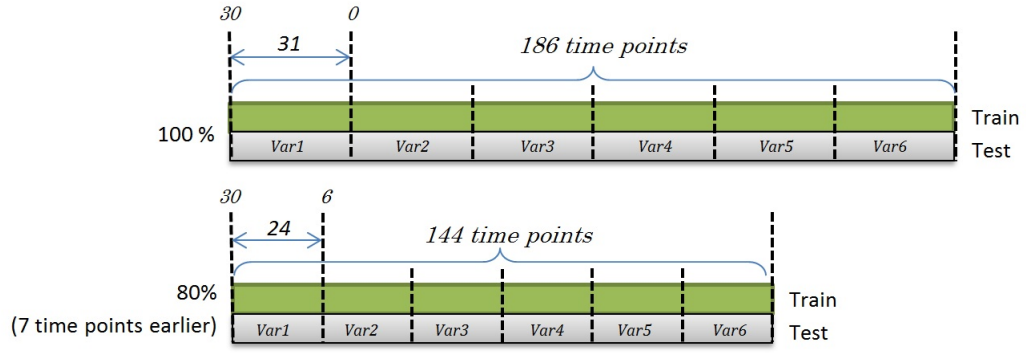

Figure 5.7: Experimental design for conventional machine learning algorithms (synthetic data).

As a comparative experiment, conventional machine learning methods such as Support Vector Machine and Multi-Layer Perceptron (MLP) are used as a baseline for performance and accuracy measures. All experiments will take the same time length percentage as the experiments with NeuCube where Experiment 1 will take the whole 100% time length while the second experiments will take 80% time length. However, it was observed that for these baseline algorithms, the time length of the training samples and validating samples needs to be equal and can not be differentiated as what is applied in NeuCube.

These methods cannot tolerate different length of feature vectors for training and recall which inhibits predictive modelling. Furthermore, in these methods the temporal relationship of a particular time point is marginalised. The data is arranged in one vector where the temporal variables for each sample were concatenated one after the other (refer to Figure 5.7). This means for Experiment 1, instead of taking 31 time points as full time length variables (100%), it takes 186 time points (31 days x 6 variables) for each sample. The second experiment takes 144 time points (80%). All experiments are executed through random sub-sampling without feature selection or normalization applied to the dataset to maintain the maximum similarity with NeuCube in relation to data representation. The parameters are optimised in a grid search method. For SVM method, the kernel is a linear. The MLP method used 20 hidden nodes and one output node, with a learning rate of 0.01 and 500 iterations.

For each experimental time length, NeuCube significantly outperformed both

SVM and MLP. Table 5.1 lists the overall accuracy from all experiments implemented.

Table 5.1: Experimental Result of Synthetic Data.

Time Length	**Accuracy (%) (Class 1, Class 2) (Mean ± Std. Deviation)**		
	NeuCube M1	**SVM**	**MLP**
100%	**(100.00, 100.00) (100.00 ± 0.00)**	(100.00, 40.00) (72.73 ± 46.70)	(54.55, 70.00) (61.90 ± 49.76)
80%	**(100.00, 80.00) (90.91 ± 30.15)**	(54.55, 70.00) (61.90 ± 49.76)	(45.45, 60.00) (52.38 ± 51.18)

This result clearly shows that NeuCube M1 is much more applicable when modelling such complex data without the need for noise filtering. This result proves that NeuCube is robust in relation to noise. Conventional methods are susceptible to noise, resulting in low accuracy if no feature selection method is applied. Furthermore; conventional methods are clearly not suitable when analysing complex problems that integrate different aspects of information because their capability is limited to learning static and vector-based data with no consideration of any spatial or temporal relationships.

Table 5.2: Experimental Result using different MLP parameter on 100% time length data

Nodes	**Iteration**	**Accuracy (Class 1, Class 2) (Mean ± Std. Deviation)**
5	300	(66.67, 60.00) (63.64 ± 50.45)
100	1000	(66.67, 60.00) (63.64 ± 50.45)
500	5000	(16.67, 60.00) (36.36 ± 50.45)

Table 5.2 shows the result of several experiments using MLP with parameter setting. While Table 5.3 shows the result of experiments using SVM with different kernel methods The data used for this simulation is the 100% time length data, used in training and validation. A manual grid search has been applied to the data for classical machine learning experiments. Increasing the number of nodes and iterations in MLP does not necessarily improve the classification accuracy, because MLP is considered to be a universal approximator [Lawrence 1997]. Furthermore MLP and SVM are non-temporal classifier and unsuitable for classifying temporal data.

Table 5.3: Experimental results using SVM with different parameter settings on 100% time length data

Kernel	**Accuracy (Class 1, Class 2) (Mean ± Std. Deviation)**
Polynomial. degree 1	(83.33, 20.00) (54.55 ± 52.22)
RBF, Gamma 3	(100.00, 0.00) (54.55 ± 52.22

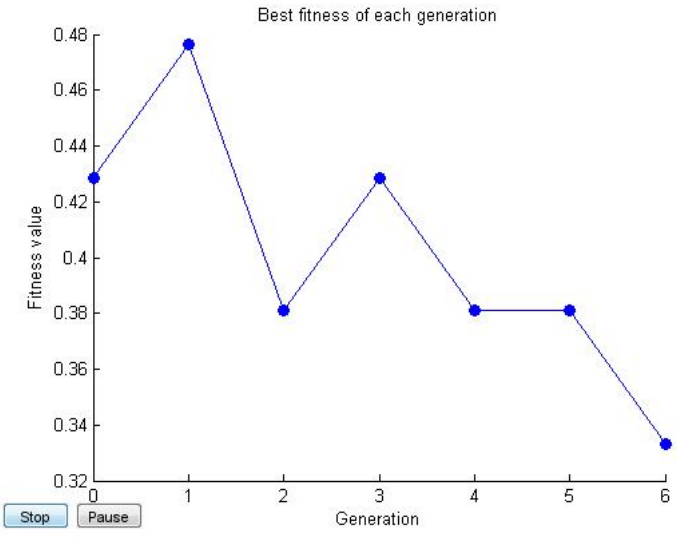

Figure 5.8: Best fitness graph for synthetic data using Genetic Algorithm Optimisation

Figure 5.8 depicts an example of a best fitness graph for synthetic data using Genetic Algorithm optimisation that has been developed recently in Neucube M1. Through this graph, we can see that the error is decreasing for each generation. In the beginning, the error is above 42% and after the 6th generation the error has decreased to 33% showing almost 70% percent accuracy with optimising just two parameter, STDP rate and drift value.

NeuCube M1 following the PMeSNNr architecture offers a more accurate prediction than other conventional methods. NeuCube M1 with less input data still produces better and more stable results. Unfortunately with less information the prediction becomes less accurate, but the result still shows that NeuCube M1 is capable of accurate predictive modelling at an earlier time point. In relation to the 'synfire chain' approach the method was demonstrated successfully, where the network learned connection pathways to produce a better prediction model.

5.4 Input Variable Mapping

Given a particular spatio-temporal data set, it is important to know how to map the data into the reservoir. For some special data such as brain EEG data, there is prior information about the location of each signal channel being collected and this information can be readily utilized for mapping the signal channels into the reservoir [Kasabov 2014b]. However for much more common applications such as climate data and ecological data, no prior knowledge about its signal location or relationship makes mapping difficult. Therefore a new method is introduced to map the input variables into the reservoir.

Suppose there are s samples in the data set and there are v variables for each sample and the observed time length of each variable is t. The basic process is choosing v input neurons from the SNNr, then map the variables into the SNNr following this principle: highly correlated spike trains are mapped to nearby input neurons. Because high correlation indicates that the variables are likely to be more time dependent with each other, this relationship should also be reflected in the reservoir. Spatially close neurons in the SNNr will capture in their connections more interactions between the input variables mapped into these neurons. Specifically, two

weighted graphs are constructed: the input neuron distance graph (NDG) and the signal correlation graph (SCG). In NDG, the input neurons' coordinate set, denoted by:

$$V_{NDG} = \{(x_i, y_i, z_i)|i = 1 \cdots N\} \tag{5.3}$$

is the graph vertex set and the graph edges are determined when: each input neuron being connected to its k nearest input neurons and the edges are weighted by the inverse of the Euclidean distance between them. For SCG, the Parzen window method is utilised to estimate the spike density function corresponding to each variable and then the graph vertex set, denoted by:

$$V_{SCG} = \{f_i|i = 1 \cdots N\} \tag{5.4}$$

The graph edges are constructed in this way: each spike density function is connected to its k highest correlated neighbours and the edges are weighted by the statistical correlation between the spike density functions of the input variables.

To find a good mapping between these two weighted graphs (NDG and SCG) under the mapping rule, a graph matching technique is adopted. This technique is a powerful tool in solving mapping problems and has been widely used in computer vision [Duchenne 2011], [Cho 2014] and pattern recognition [Sanfeliu 1983], [von der Malsburg 1988], [Lladós 2001]. For these two graphs, their adjacency matrices are computed and written as A_n and A_s. The graph matching method is aims to find a permutation matrix P that minimizes the following objective function:

$$minP||A_n - PA_sP^T||_F^2 \tag{5.5}$$

Where $||\bullet||_F$ denotes the Frobenius matrix norm. Exactly solving this problem is known to be of order NP due to its 0-1 combinatorial optimization property. Many algorithms have been proposed to find an approximated solution. Among these algorithms the Factor Graph Matching (FGM) algorithm [Zhou 2012] has been demonstrated to produce state-of-the-art results. So here we utilize the FGM algorithm to solve problem 5.5 with the following settings: suppose in NDG the sum of graph

edge weights of an vertex, vertex $i_{NDG} \in V_{NDG}$, to all other vertices is $d(i_{NDG})$, and similarly, in SCG the sum of graph edge weights of vertex $i_{SCG} \in V_{SCG}$, to all other vertices is $c(i_{SCG})$, then the difference between $d(i_{NDG})$ and $c(i_{SCG})$ reflects the similarity of the positions in each graph.

The similarity of the two vertices is computed using the formula:

$$exp(-|d(i_{NDG}) - c(i_{SCG})|^2/2\sigma_n^2); i_{NDG}, i_{SCG} = 1 \cdots v \tag{5.6}$$

and the graph edge similarity:

$$exp(-|a_{ij}^{NDG} - a_{kl}^{SCG}|^2/2\sigma_e^2); i, j, k, l = 1 \cdots v \tag{5.7}$$

Where: a_{ij}^{NDG}, a_{kl}^{SCG} are graph edge weights in NDG and SCG, respectively; σ_n and σ_e are two parameters to control the affinity between neurons and edges respectively.

Figure 5.9 shows the matching result. The left graph is the input NDG and the right graph is SCG. We can see that after matching, high correlated features are mapped to nearby input neurons.

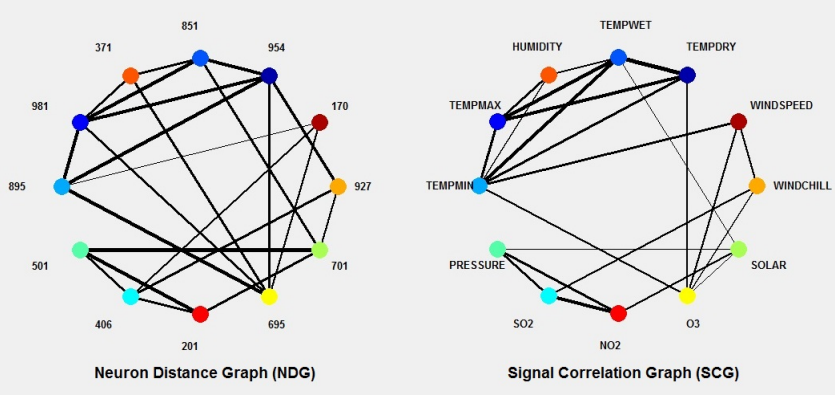

Figure 5.9: Input variable mapping panel.

5.5 Visualisation

Visualisation of the neuron activities, the connection weight changes and the structure of the SNNr are important for the understanding of the data and the processes that generated it. In NeuCube, each time moment of the neuron's spiking state in the SNNr and their connection adjustment can be visualised. Figure 5.10 shows

snapshots of the instantaneous neuron spiking state; Figure 5.11 shows the connection weight change before reservoir training and after training for a particular neuron to other neurons in the reservoir; Figure 5.12 depicts positive and negative spike emitted by each neurons; and Figure 5.13 shows the neuron's activation level. This is much different from conventional methods such as SVM which have been used for similar tasks but without offering facilities to trace the learning processes for the sake of data understanding.

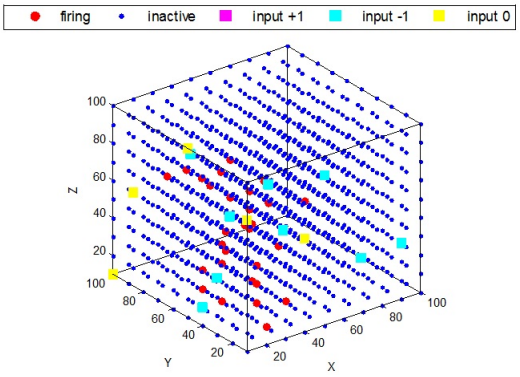

Figure 5.10: Neuron spiking state.

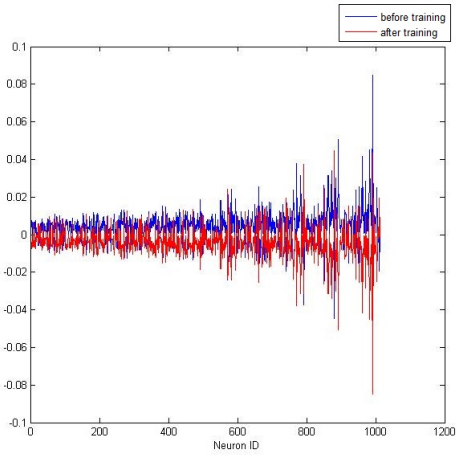

Figure 5.11: Neuron weight changed between particular neuron and other neurons in the reservoir before and after training.

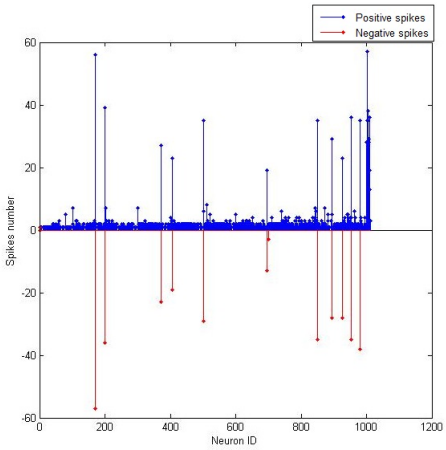

Figure 5.12: Spike emitted from each neurons either positive or negative spike.

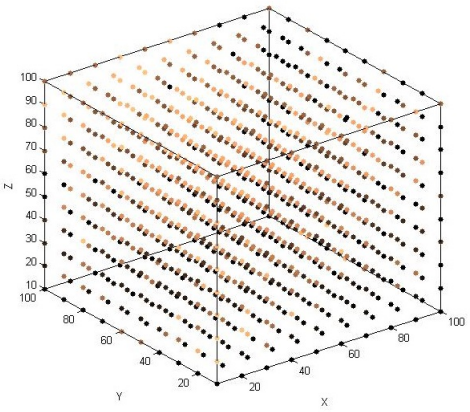

Figure 5.13: Activation level of each neuron where the brighter the neuron's color the more spikes the neuron emitts during training or validation and black represents no firing.

After it is trained, the SNNr has captured spatial and temporal relationships from the data. The information spreading algorithm [Zhou 2004] is used to analyze the network structure and to explore the relation between a learnt structure and a data set used for training. This algorithm has been demonstrated as a powerful tool for analysis dynamic activity spreading (here spike spreading) in networks [Shrager 1987].

5.6 Chapter Summary

NeuCube M1 in particular can learn incrementally from data streams, can include 'on the fly' new input variables, new output class labels or regression outputs, can continuously adapt its structure and functionality, can be visualised and interpreted for new knowledge discovery and for better understanding of the data and processes that generated it. NeuCube M1 can be used for predictive modelling where early event prediction is possible due to the ability of the SNN to spike early, before whole input vectors are presented (i.e. it can respond when not all the test vector is yet presented after it has been previously trained on full vector input). More details on the NeuCube system can be found in Appendix A. The following two chapters respectively outline the application of NeuCube to two real world case studies, firstly for stroke prediction and secondly for aphid population prediction.

Chapter 6

NEUCUBE-BASED DATA MODELLING FOR STROKE RISK PREDICTION

"The true parents of creativity are curiosity and necessity."

- Max McKeown

6.1 Introduction

According to the WHO global report, health related problems like chronic diseases are the major cause of death in almost all countries and it is projected that 41 million people will die of a chronic disease by 2015 [Organization 2005]. Chronic disease, like stroke has become a leading cause of death and adult disability in the world [McArthur 2010]. While in New Zealand, stroke has become the third leading cause of death after cancer and heart disease, and it is the greatest cause of disability in older people [Tobias 2002]. Each year it is predicted that over 7,000 New Zealanders will experience a stroke event, and of this population at least three-quarters will die or be dependent on others for care one year after the event [Gommans 2003]. Health costs associated with stroke both to individuals and the health system is high and increasing. When severe, stroke has a substantial impact on the psychological and physical well-being of both patients and their families [Hackett 2000].

Following a brief introduction to stroke, this chapter reviews various studies related to stroke occurrences that mostly use statistical methods for predicting stroke risk and outcome. This chapter also introduces an Auckland Regional Community Stroke (ARCOS) study on which the NeuCube method has been applied.

6.2 Review on Stroke Disease

6.2.1 What is Stroke?

In brief, stroke was defined according to the World Health Organization criteria as rapidly developing signs of focal (or global) disturbance of cerebral function, lasting longer than 24 hours or leading to death, with no apparent cause other than vascular [Investigators 1988]. This definition includes spontaneous subarachnoid hemorrhage but excludes subdural and extradural hematomas and transient ischemic attacks [Hackett 2000].

In medical terms a stroke is an acute vascular injury of the brain where it can be caused by a blood-clot blockage, by narrowing of the blood vessels (clogging), by both a blockage and narrowing, or by a rupture of blood vessels, resulting in a lack of adequate blood supply to the brain [Feigin 2011]. The symptoms of stroke may vary depending on the size of the damage be it physical, psychological, behavioral or any combination of them. Below is a list of typical symptoms of a person having a stroke that appears suddenly:

1. severe headache
2. memory loss
3. paralysis
4. weakness or patient clumsiness (loss of balance or co-ordination)
5. a loss of sensation in the face, arm or leg on one side of the body
6. difficulty of talking and/or understanding (without hearing problems)
7. difficulty of swallowing and a partial loss of vision on one side

If a person developed one or a combination of the above symptoms that lasts 24 hours or longer, the person is said having a stroke. While, if the symptoms disappear within 24 hours the person is said having a transient ischaemic attack (TIA) which is a minor stroke that is a major risk factor for an ischaemic stroke. TIA symptoms come on quickly and last 10 seconds to 15 minutes, occasionally they can last for 24 hours [Feigin 2011]. Two types of brain strokes are ischaemic and haemorrhagic

stroke. Figure 6.1 below depicts these two types of stroke. Ischaemic stroke or brain infarction is most commonly occurring in both women and men because it is caused by either a blockage from a blood clot or narrowing of an artery or arteries leading to the brain. While a haemorrhagic stroke is caused by bleeding into the brain tissue (also called as intracerebral haemorrhage or intracerebral haematoma) or into the narrow space between brain surface and the layer of tissue that covers the brain (called a subarachnoid haemorrhage).

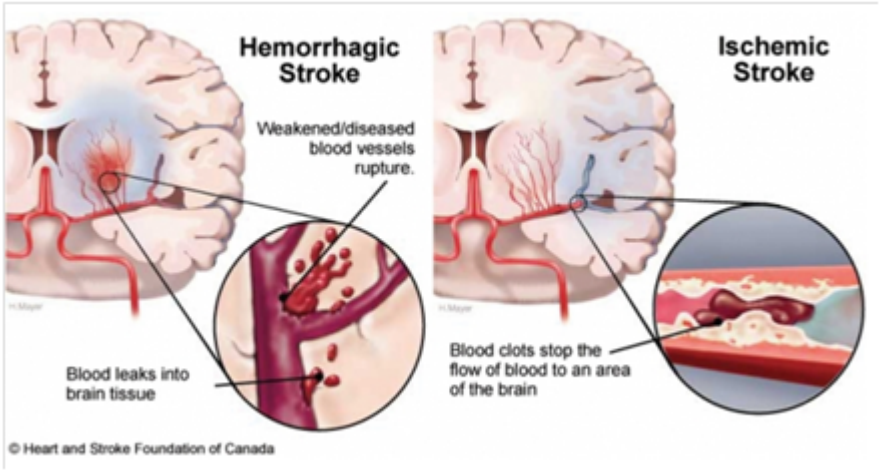

Figure 6.1: Types of brain stroke [Ritter 2015].

6.2.2 Risk Factors of Stroke

Much research has been done to identify risk factors of stroke and most agree that strokes result from a combination of modifiable risk factors and non-modifiable risk factors. Some risk factors can be controlled or eliminated completely by either medical treatment or changes of lifestyle. However they are certain risk factors that cannot be controlled such as ageing, genetic factors, ethnicity, gender and external environment. The modifiable risk factors [Gomes 2014b], [O'Donnell 2010] are:

1. High blood pressure
2. Heart disease
3. Heart rhythm disorders e.g.: atrial fibrillation
4. Smoking
5. Diabetes

6. High blood cholesterol levels
7. Oral contraceptives
8. Excessive alcohol intake
9. Obesity
10. Psychosocial factor
11. Unhealthy diet
12. Sedentary lifestyle

It is often observed that individuals with similar medical risk factors suffer a stroke at different time points to each other. It has been suggested that the external environment plays a significant role as a trigger of stroke. The influence of climate (weather changes) and air pollution upon stroke risk is biologically plausible [McArthur 2010]. Based on this principle, many studies have been carried out to evaluate the association of environmental variables with stroke occurrences. These studies reported conflicting results in different populations/regions. Most of the studies found increased of stroke incidents in low temperature and cold weather [Feigin 2000], [Hong 2003], [Gill 2013], [Gomes 2014a], [Jimenez-Conde 2008]. However, several studies found increased stroke mortality in relation to high temperature or during heat waves [Berginer 1989], [Dawson 2008], [Lim 2013], [Chen 2013]. While a study considered changes in temperature are more important than absolute value which a change in temperature of more than 5°C is associated with increased risk of ischemic stroke regardless whether the change is negative or positive [Kyobutungi 2005].

Other than temperature, some other studies found evidence of association between other meteorological factors such as atmospheric pressure and solar radiation towards stroke incidents [Shaposhnikov 2014], [Feigin 2014]. Several studies also investigated the association of air pollution variables with stroke incidents and most of the studies originated from Asian countries. Several studies found evidence suggesting association between air pollution with increased risk of stroke on warm days [Chen 2014], [Low 2006], [Yang 2014], [Tsai 2003].

Most of the studies above employed univariate analysis [Berginer 1989], [Hong 2003], [Gill 2013], [Lim 2013], [Feigin 2014] and multivariate analysis [Low 2006], [Çevik 2014], [Shaposhnikov 2014] in finding associations between environmental variables toward the increase of stroke risk. In the case of multivariate analysis, the studies observed and analysed more than one statistical outcome variable at a time, by analysing a single variables effect while controlling other variables. Hence analysing single variable outcomes for a complex problem seems insufficient. Given that human physiology is complex, we believe that combinations of environmental variables can increase the risk of stroke events rather than a single variable.

This hypothesis corresponds to the simplified framework of the causal relations between climate-related factors and stroke [Gomes 2014b] depicted in Figure 6.2. The framework aims to clarify the relations between environmental factors and stroke by grouping the risk into different categories that have a cascading effect that leads to stroke occurrence. They believed modifiable risk factors could influence how environmental factors directly or indirectly affect human homeostasis, acting as effect modifiers, this means each individual may experience different effects from environmental influence according to the presence or absence of certain risk factors. Their study also suggests that transitory changes in environment can trigger stroke occurrences by influencing biological factors directly or indirectly depending on the individual exposure and current conditions. They also agree that the study about the direct or indirect effect of environmental condition according to the presence of certain modifiable risk factors could lead toward the adaptation of therapeutic and preventive measures according to individual characteristics.

Given this increasing evidence, we hypothesised that environmental variables do contribute in influencing the incremental risk of stroke incident especially in individual with prevalence of conventional risk factors for stroke.

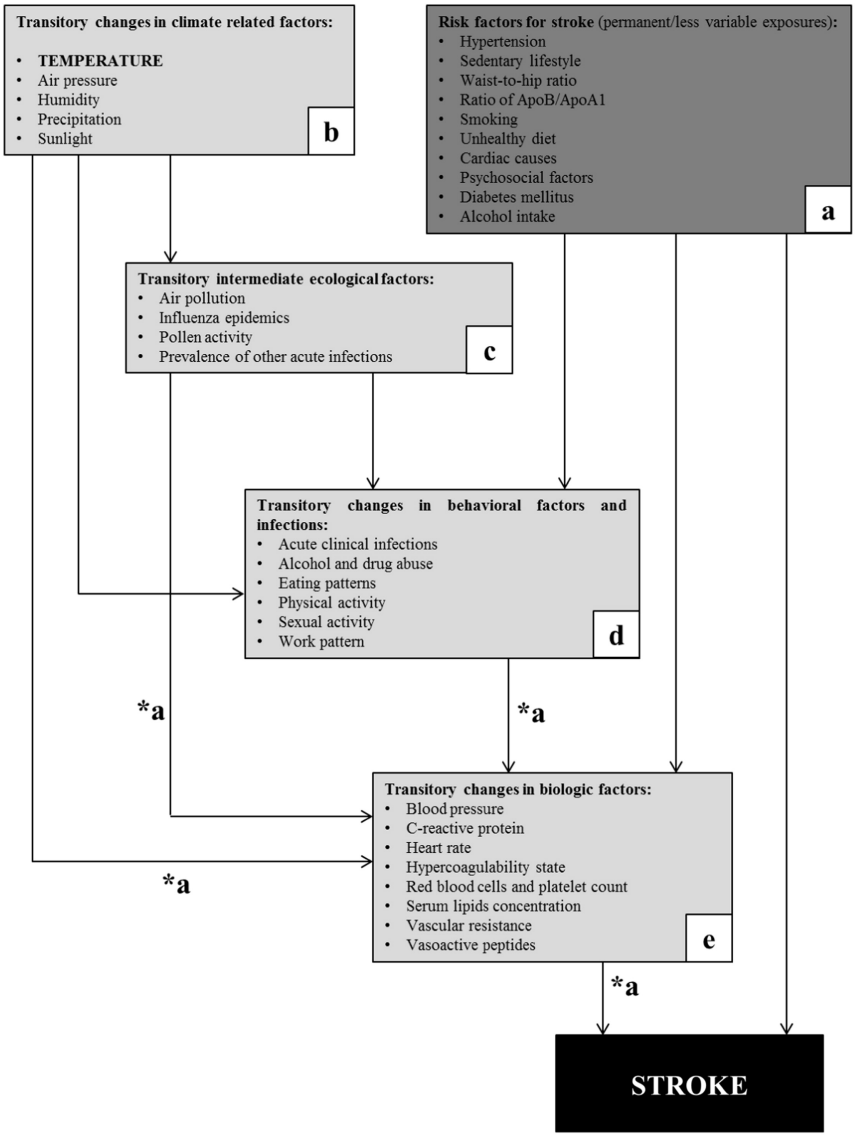

Figure 6.2: Simplified diagram of the causal relations between climate-related factors and stroke [Gomes 2014b].

6.3 Stroke Risk Prediction Case Studies

Moving toward personalised preventive measures, in this study we apply an individualised approach to stroke risk assessment using personalised modelling based on spiking neural networks method (PMeSNNr). The first objective of this study is to verify that the increment of stroke risk in an individual with modifiable risk factors for stroke is triggered by a combination of environmental factors. Secondly, to un-

derstand the association of combined environmental variables toward a stroke event on individual level of risk. Thirdly, exploring the influence of prolonged exposure to inclement environmental conditions several days before a stroke event that can help us determine the earliest time point at which we can best predict the risk of stroke incident in individuals.

6.3.1 Data Description

The dataset consists of 2805 samples (all with first-ever occurrence of stroke) taken from the Auckland Regional Community Stroke Study (ARCOS) population. This data set has been published in several studies where they apply different methods to the case study, for example statistical methods [Anderson 2005], FaLK-SVM (global method) [Liang 2011], and a combination of evolving personalised method (evoPM) with gravitational search algorithm and recurrent reservoir [Kasabov 2014a]. In exploring the feasibility of NeuCube for predictive modelling of SSTD this study has been published recently in 2014 International Joint Conference on Neural Network (IJCNN) [Othman 2014]. This chapter will further report on the network analysis for personalised modelling.

The population-based stroke data incidences were recorded in Auckland (NZ) between the years 1981-1982, 1991-1992 and 2002-2003. The third ARCOS study was chosen which is between 2002 and 2003, consisting of 1207 subjects. These subjects were then divided into several subgroups stratified by season, history of hypertension, smoking status and age to explore the susceptibility of groups to the influence of environmental changes. The approach followed the concept of personalised modelling, based on an individuals measured features several nearest neighbour are selected that have similar features with the individual yielding a reduction in the number of samples used for further investigation. The reasons for selecting different age groups are to explore the effect of environmental variables on different age groups in different seasons and to demonstrate the applicability of NeuCube regardless of the size of the data.

Subject were selected covering the winter season and that were aged between 50 and 70 years that have a history of hypertension and smoking at the time of stroke, while summer subjects (age 50 to 70 years old) and spring subjects (age 35 to 50

years old) are selected if they have a history of hypertension but smoking status is not considered. As for the autumn subjects, they are selected among subjects aged 20 to 35 years old with no history of hypertension and smoking status is not considered (refer to Table 6.1). Each sample is described by a total of eighteen (18) features which consist of six (6) static features (categorical data) along with twelve (12) environmental (temporal) features (continuous daily data). The NeuCube model is built based only on the temporal environmental data.

Static features are listed as follows:

1. Age.
2. Gender.
3. History of hypertension.
4. Smoking status.
5. Season.
6. Date of stroke.

Temporal features are listed as follows:

- Eight (8) daily mean weather measures:
    1. Wind speed (WINDSPEED, Knots).
    2. Wind chills (WINDCHILL, Degree).
    3. Temperature dry (TEMPDRY, °C).
    4. Temperature wet (TEMPWET, °C).
    5. Temperature max (TEMPMAX, °C).
    6. Temperature min (TEMPMIN, °C).
    7. Humidity(HUMIDITY, %).
    8. Atmospheric pressure in hectopascals (PRESSURE, hPA).
- Three (3) daily mean air pollution data:
    1. Sulfur dioxide (SO_2) (µg/m^3).

2. Nitrogen dioxide (NO_2) (µg/m^3).
3. Ozone (O_3) (µg/m^3)

- One (1) planetary geomagnetic activity:
  1. Solar radiation (Ap Index).

Table 6.1: Stroke Occurrences Case Studies

No	Season	Age Range	History of Hypertension	Smoking Status	Number of Selected Subject (control and case)
1	**Winter**	50-70	Yes	Current Smoker	20
2	**Summer**	50 -70	Yes	NA	46
3	**Spring**	35-50	Yes	NA	26
4	**Autumn**	25-35	NA	NA	16

Since the data only consisted of stroke subjects, the case-crossover design method is implemented to create control subjects. As a result the 'case' subject will act as 'control' subject discriminated by different time windows. Time windows are segregated based on the daily temporal environmental data collected for 60 days prior to stroke event. Figure 6.3 illustrates the segregation of the time windows where each time window consists of 20 days. Time window 1 refers to 'low risk' stroke class (Class 1), time window 2 for 'high risk' stroke class (Class 2) and a transition window between the two. The transition window represents the changes in the environment over time while the exact time point that influences the stroke event is uncertain. Therefore to minimise misclassification error, the transition window is excluded.

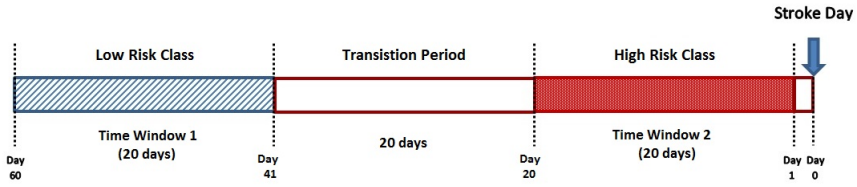

Figure 6.3: Time windows to discriminate between 'low risk' and 'high risk' stroke class [Othman 2014].

6.3.2 Brief Data Overview

This section will briefly take a look at the winter data for several types of environmental features for certain subjects that suffered a stroke in the winter season. Figure 6.4 depicts the four types of temperature readings for a male subject age 51 over the period of 60 days before the stroke event. The temperature readings were temperature maximum, temperature minimum, temperature dry and temperature wet.

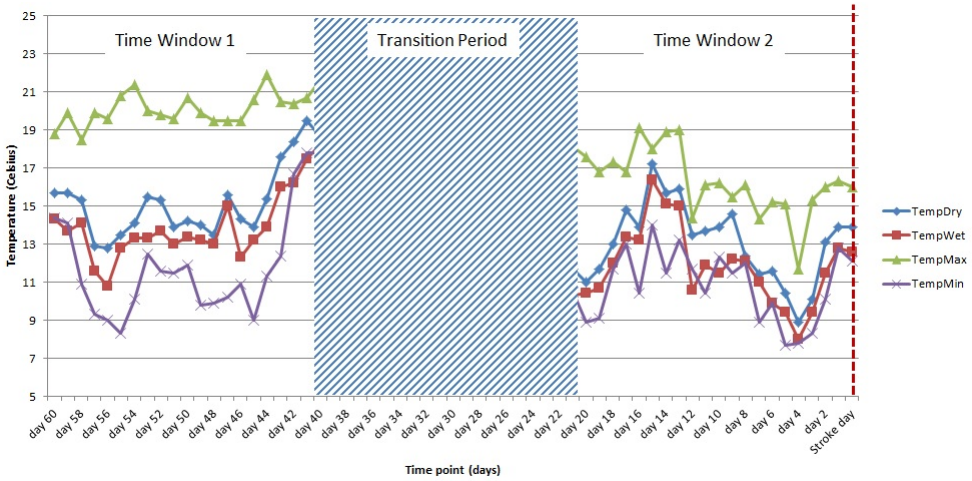

Figure 6.4: Four types of temperature reading 60 days preceding winter stroke event for male subject, age 51 [Othman 2014].

The variable patterns between time window 1 (20 days pre-stroke event) and time window 2 show a significant difference where in time window 2, the temperature readings are relatively chaotic compared to time window 1. Within the 20 days of time window 2 the reading shows several drastic drops in the temperatures. Many researchers studied these variables to find a correlation with stroke attack and statistically they agreed that a significant dropped in temperature readings can increase the risk of stroke attack [Feigin 2000], [Hong 2003], [Gill 2013], [Gomes 2014a], [Jimenez-Conde 2008].

Beside temperature, other variables may trigger the stroke event. Figure 6.5 depicts the atmospheric pressure reading for several patients who were 60 years old. The atmospheric pressure clearly dropping within the period of 20 days (time window 2) preceding the stroke event compared to the earlier 20 days (time window 1). Moreover, the readings are hectic in the 20 days period before stroke with several

drastic drops for all patients.

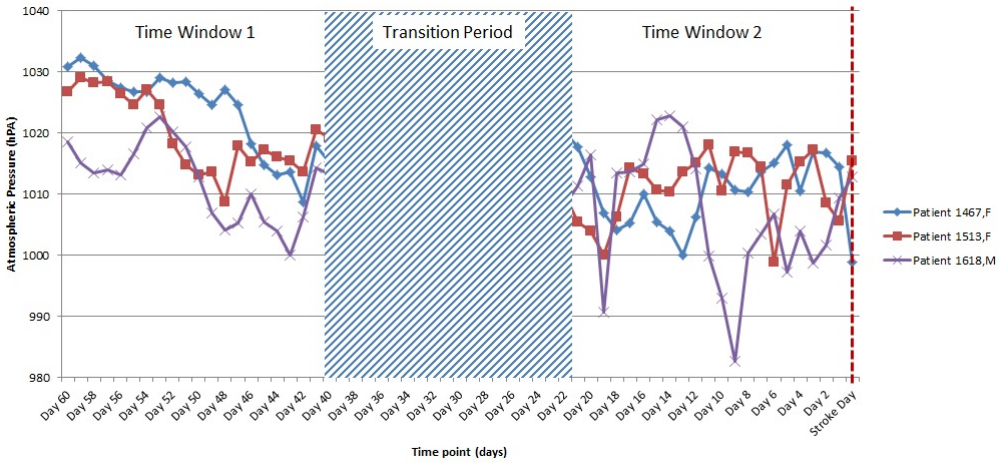

Figure 6.5: Atmospheric pressure reading 60 days before the winter stroke event for several subjects in age group 60 [Othman 2014].

Figure 6.6 shows solar radiation readings for different patients, which illustrated that the patients were exposed to many high solar radiation periods before the stroke attack. The exposure to solar radiation over time surely has effects on human health. As confirmed by research that prolonged exposure could lead to skin cancer, eye disease and decreases in the immune system, consequently increase the risk of cardiovascular disease [Norval 2001],[Juzeniene 2011].

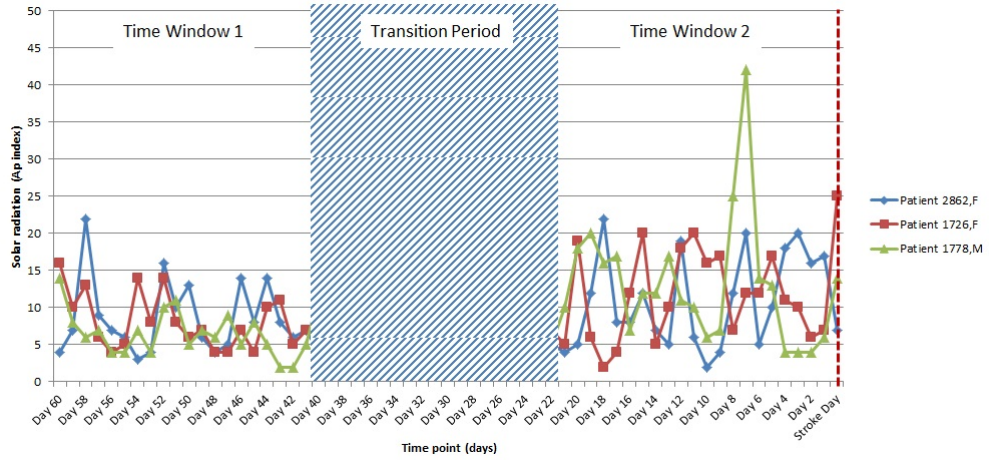

Figure 6.6: Solar radiation reading 60 days preceding winter stroke event for three subjects [Othman 2014].

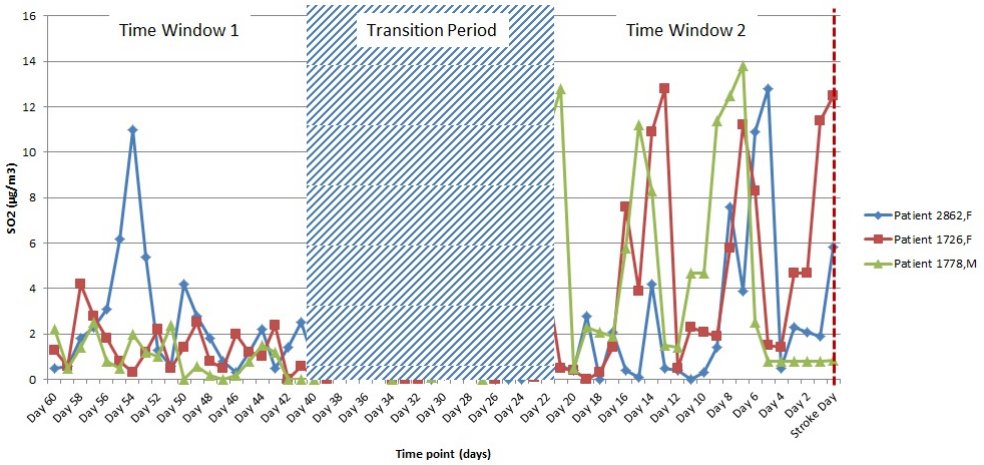

Figure 6.7: Sulfur oxides gas reading 60 days preceding winter stroke event for three subjects [Othman 2014].

From the three types of air pollutant variables, SO_2 is illustrated in Figure 6.7. SO_2 is a highly reactive gas known as an "oxide of sulfur". These particles penetrate deeply into sensitive parts of the lungs and can cause or worsen respiratory disease, and can aggravate existing heart disease, leading to increased hospital admissions and premature death. The graphical representation of three subjects showed increased exposure to SO_2 in time window 2 compared to the exposure during time window 1. This may further clarify our assumption that SO_2 in combination with other variables increases the risk of stroke. All the graphs depict severe environmental conditions where human body was exposed to extreme weather, high solar radiation, and excessive air pollutions. These variables in combination are very hazardous to human health.

Another two types of air pollution, O_3 and (NO_2) are also included as contributing factors toward stroke occurrences. NO_2 is one of a group of highly reactive gasses known as "oxides of nitrogen" or "nitrogen dioxide (NOx) that forms quickly from emissions from cars, trucks and buses, power plants, and off-road equipment. In addition to contributing to the formation of ground-level ozone and fine particle pollution, NO_2 is linked with a number of adverse effects on the respiratory system. Some studies confirm relationships between NO_2 exposure and ischaemic stroke [Hong 2002], [Andersen 2012]. O_3 gas pollution also known as ground level or "bad" ozone is created by chemical reactions between oxides of nitrogen (NOx) and

volatile organic compounds (VOC) present in sunlight. Emissions from industrial facilities and electric utilities, motor vehicle exhaust, gasoline vapors, and chemical solvents are some of the major sources of NOx and VOC. Breathing ozone can trigger a variety of health problems, particularly for children, the elderly, and people of all ages who have lung diseases such as asthma. Several studies confirm a relationship between low level O_3 exposure and ischaemic stroke in high vascular risk subgroups with a linear exposure-response relationship [Suissa 2013], [Henrotin 2007].

6.3.3 Experimental Design

Two experiments, depicted in Figure 6.8, were conducted for each case study of stroke occurrences from different seasons to verify the validity of NeuCube and its ability to predict accurately the risk of stroke occurrences as early as possible. Based on temporal data collected in the given time windows, the first experiment takes the full time length variables, for both training and testing under the assumption that the perfect environmental data can be obtained, which is an ideal case. The first experiment attempts to predict accurately the risk of stroke occurrence 1 day ahead. The predictive ability of NeuCube is further justified by training the full time length variables (100%) and validating 75% (6 days earlier of the time length). This method preserved the temporal relationship residing between the input features where each time period describes all features related to each time point.

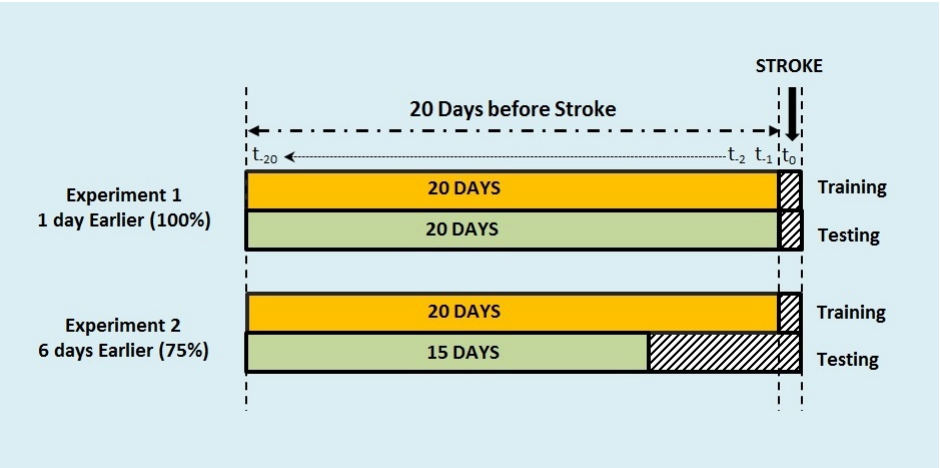

Figure 6.8: Experimental design for NeuCube M1. The yellow bars represent the time length for training samples and the green bars represent the time length for testing samples.

Each case study has its own customised reservoir size and optimised parameters (refer to Appendix B). All were trained and tested in a leave-one-out cross validation (LOOCV) mode, which is more suitable for small datasets. The parameters for each case study were optimised through a Grid Search optimisation procedure in NeuCube. After attaining optimised parameter settings, the case studies are trained and recalled during LOOCV mode recurrently until NeuCube obtained the best model. The classifier selected for this experiment was deSNNs_wknn where the parameter for mod is 0.04, drift is 0.25, k is 3 and Probability threshold is 0.5.

As comparative experiment, conventional machine learning methods such as Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Evolving Classification Function (ECF) are used as a baseline for performance and accuracy measures. Experiments used the same time length. It was observed that for these baseline algorithms the time length of the training samples and testing samples needs to be equal. These methods cannot tolerate different length of feature vectors for training and recall which inhibits predictive modelling. Furthermore, in these methods the temporal relationship of a particular time point is marginalised.

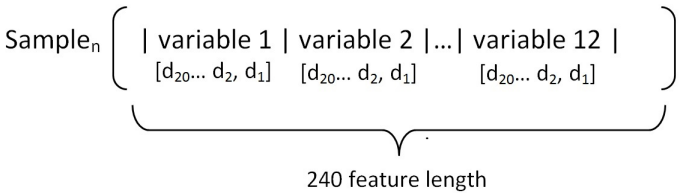

Figure 6.9: Experimental design for classical machine learning

The data is arranged in one vector where the temporal variables for each sample were concatenated one after the other (refer to Figure 6.9). This means for Experiment 1, instead of taking 20 days full time length variables (100%), it takes 240 time points (20 days x 12 features) for each sample. The second experiment takes 180 time points (75%). All experiments are executed through LOOCV without feature selection or normalisation applied to the dataset. The parameters are optimised in a Grid Search method. For SVM method, the kernel is a first degree polynomial. The MLP method used 20 hidden nodes and one output node, with learning rates of 0.01 and 500 iterations, while ECF used 4 epochs and 2 membership functions.

6.3.4 Result and Analysis

Table 6.2 lists the overall accuracy for all experiments. The best accuracy obtained for summer is 100% (100% for the TP - high risk, class 2; 100% for the TN - low risk - class 1); winter obtained 95% (100% for the TP - high risk, class 2; 90% for the TN - low risk - class 1); spring obtained 92% (85% for the TP - high risk, class 2; 100% for the TN - low risk - class 1); and autumn obtained 88% (88% for the TP - high risk, class 2; 88% for the TN - low risk - class 1). The lower accuracy for autumn is assumed to be caused by several factors associated with seasonal environmental conditions and clinical features for each subject. Even though some conventional machine learning methods like SVM and ECF achieved high prediction accuracy, the methods did not take into account the temporal relationship of the features, so the temporal influence between features is lost.

Figure 6.10 depicts the best fitness graph for summer data optimised using a Genetic Algorithm. The graph shows a decrease in error for each generation, where for the first generation the error value is approximately 0.22 and finally after the 9th generation the error is less than 10%. This mean the classification accuracy is more than 90%. This validates that performance of NeuCube in classifying summer data.

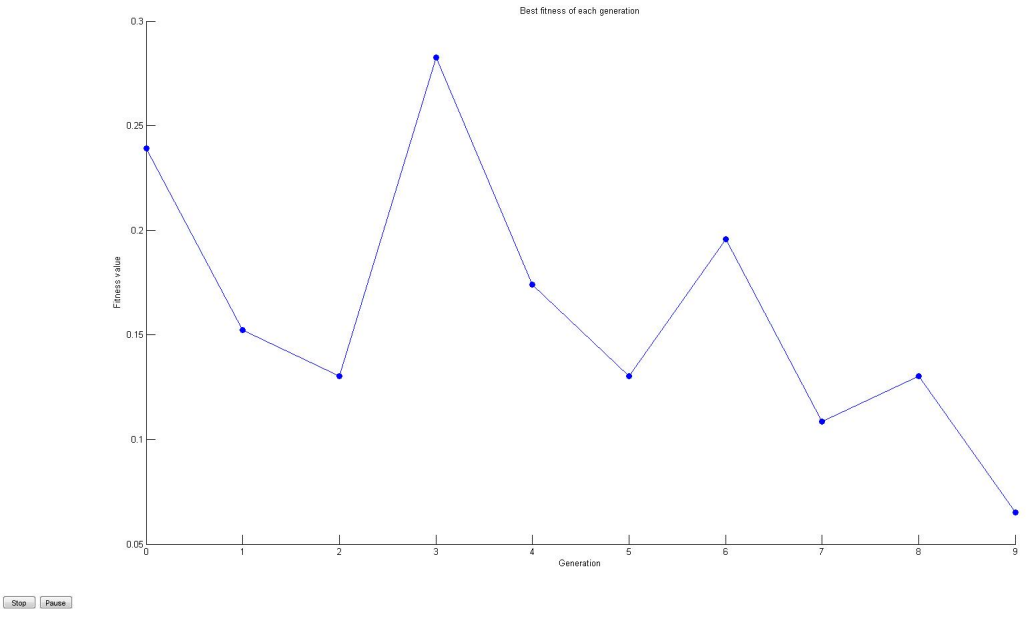

Figure 6.10: Best fitness graph for summer data using Genetic Algorithm Optimisation.

As previously mentioned, autumn achieved an overall lower accuracy compared to winter and summer may have been caused by several factors. The winter and summer season in New Zealand can be rather harsh and extreme whereas autumn weather is mild, this may suggest several drastic changes in the time windows. Based on previous study, the sudden change in temperature (increased or decreased) may cause stroke incidents [Kyobutungi 2005]. Another contributing factor is the age of the subjects in the case studies; autumn subjects involve a younger group of individuals compared to the other seasons. It may suggest that environmental changes cause more harmful effect to older individuals then younger individuals.

Table 6.2: Comparative Experimental Results for Stroke Risk Prediction

Days Earlier	**Accuracy % (Low Risk, High Risk) (Mean ± Standard Deviation)**			
	NeuCube	**SVM**	**MLP**	**ECF**
Summer				
1	**(100.00, 100.00) (100.00 ± 0.00)**	(91.30, 100.00) (95.65 ± 20.62)	(60.87, 43.48) (52.17 ± 50.50)	(82.61, 100.00) (91.30 ± 28.49)
6	**(100.00, 100.00) (100.00 ± 0.00)**	(100.00, 95.65) (97.82 ± 14.74)	(69.57, 43.48) (60.86 ± 49.34)	(73.91, 100.00) (86.96± 34.05)
Spring				
1	**(100.00, 84.62) (92.31 ± 27.17)**	(84.62, 92.31) (88.46 ± 32.58)	(38.46, 69.23) (53.85 ± 50.83)	(69.23, 69.23) (69.23 ± 47.06)
6	**(92.31,92.31) (92.31 ± 27.17)**	(84.62, 92.31) (88.46 ± 32.58)	(38.46, 46.15) (42.31 ± 50.38)	(69.23, 76.97) (73.08 ± 45.23)
Winter				
1	**(90.00, 100.00) (95.00 ± 22.36)**	(70.00, 40.00) (55.00 ± 51.04)	(50.00, 40.00) (45.00 ± 51.04)	(30.00, 90.00) (60.00 ± 50.26)
6	**(90.00, 70.00) (80.00 ± 41.04)**	(70.00, 30.00) (50.00 ± 51.30)	(20.00, 30.00) (25.00 ± 44.43)	(30.00, 60.00) (45.00 ± 51.04)
Autumn				
1	**(87.50, 87.50) (87.50 ± 34.16)**	(62.50, 50.00) (56.25 ± 51.23)	(12.50, 50.00) (31.25 ± 47.87)	(75.00, 50.00) (62.50 ± 50.00)
6	**(75.00, 62.50) (68.75 ± 47.87)**	(37.50, 62.50) (50.00 ± 51.64)	(50.00, 37.50) (43.75 ±51.23)	(75.00, 50.00) (62.50 ± 50.00)

Looking at the result further, early prediction of stroke risk is important to prevent stroke incident in the future. A low risk group and a high risk group are taken from a different time windows. NeuCube could demonstrate the temporal relationships between these two groups. For example if we take 75% of the data which indicates 6 days earlier prediction, the high risk group refers to data points between day 20 and day 15 while a low risk group refer to data points between day 60 and day 45. Six (6) days earlier prediction means that we predict whether the risk of the individual having a stroke 6 days in the future. During the low risk period summer subjects were accurately classified as having a low risk of stroke with 100% accuracy while spring and winter subjects almost had an accurate risk prediction with 92% and 90% respectively. Some of the subjects were classified as having a high risk of stroke 6 days in the future, and for some reason they survived but suffered a stroke 40 days later.

However for the high risk group, prediction at 6 days earlier was rather low for winter and autumn with 70% and 63% respectively. This may suggest these two groups are not exposed to harsh environmental condition in that particular time window because for 1 day earlier prediction all subjects have a high prediction accuracy, which is 100% for winter and 88% for autumn. This gives as indication that sudden change in environment can causes harm to humans. Summer and spring yielded 100% and 92% accuracy respectively for 6 days earlier prediction suggesting that harsh environmental condition during the summer season could be a substantial influence toward stroke event for this particular older group age of subjects and that is true in these case where all subjects had a stroke attack 6 days later.

Autumn achieved a lower accuracy (88%) of 1 day earlier prediction for the high risk group, verifying that younger subjects are less affected by sudden environmental changes and their stroke may be caused by other contributing factors such as medical conditions. As earlier mentioned other contributing factors have a cascading effect toward a stroke event (refer to Figure 6.2). Looking at the age difference between these groups where winter, summer and spring subjects are much older compared to autumn subjects, we are in agreement with other researchers in claims that older people are more effected by seasonal conditions than younger people [Chen 2014], [Sobel 1987].

At 6 days earlier prediction a good indicator to take some preventive measures when compared to other commercial products that only predict stroke risk within a 5 to 10 years. Moreover, as the time goes by newly collected data can be added directly to the testing sample to give a better prediction without re-training the model as would be the case with SVM or MLP methods. This capability is important when modelling real time data that makes our methods suitable for prediction of real time events.

Another advantage of our method is the visualisation of the pattern learned instead of the black box learning method applied in most conventional methods. PMeSNNr method proves its applicability in modelling complex data and is more stable compared to other classical machine learning methods. This also adds weight to the premise that NeuCube M1 is robust to noise, because through the PMeSNNr method continuous value data is input directly into NeuCube without preprocessing. It is acknowledged that most conventional machine learning methods are vulnerable to noise, resulting in a lower accuracy if no preprocessing is applied such as normalization or feature selection to filter unwanted signal. Furthermore, the limited capability of conventional machine learning methods in learning only static and vector-based data, unlike SNN that can integrate time and space together and learn its spatial-temporal relationships. Further investigation into the spike transmission may reveal new knowledge about the influence of environmental factors in the prediction of stroke and possibly its prevention.

6.3.5 Group Level Network Analysis

Neurons are clustered based on the spike transmission in the reservoir. The input neurons (temporal variables) are treated as information sources in the SNNr and a spreading algorithm [Zhou 2004] is utilized to determine neuron clusters belonging to each input neuron. As a result of the unsupervised training, the more the spikes transmitted between two neurons the stronger they will be connected to each other.

The four following figures (6.11 - 6.14) illustrate the percentage of neurons that belong to the same cluster from which it receives the most spikes or more connection weight changes in the neurons.

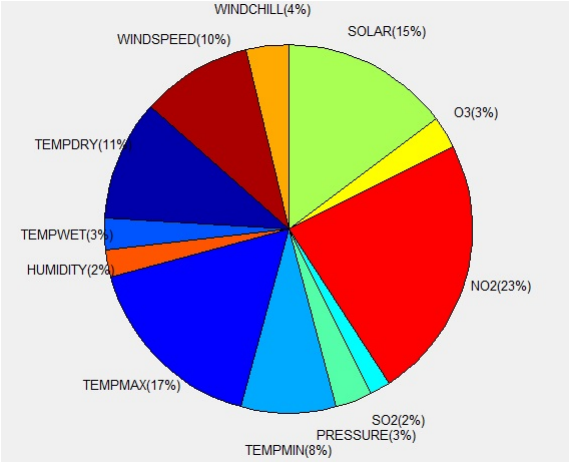

Figure 6.11: Neuron proportion for summer subjects.

For summer (refer to Figure 6.11), the neurons belonging to nitrogen dioxide gas pollution have the highest activation percentages of 23%, followed by temperature maximum 17%, solar radiation 15% and temperature dry 11%. During summer season the temperature is always higher that other seasons and drastic changes of the temperature maximum may contribute to the higher risk of stroke. Several researches indicate there is an association between increase of stroke in warmer seasons and polluted environment [Chen 2014], [Low 2006], [Yang 2014], [Tsai 2003].

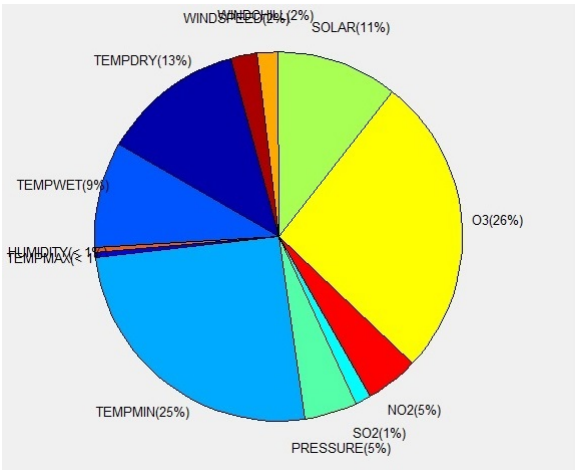

Figure 6.12: Neuron proportion for winter subjects

For winter (refer to Figure 6.12), neurons belonging to temperature min have a higher activation percentage which of 25% compared to other temperature readings. Also the highest neuron proportion percentage belongs to ozone gas pollution with 26%, while temperatures dry at 13% and solar radiation at 11%. There are numerous

studies that relates between cold temperature and stroke occurrences.

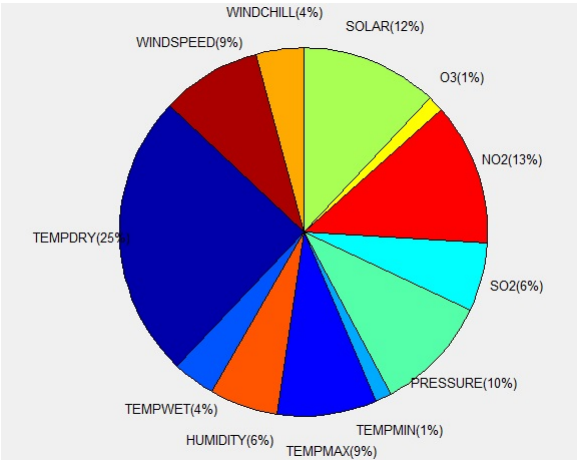

Figure 6.13: Neuron proportion for spring subjects

As for the spring (Figure 6.13) the highest neuron activation is temperature dry with 25%, followed by NO_2 with 13%, solar radiation 12% and temperature maximum 9%. This season also has a high activation of NO_2 and solar radiation. The spring season is always mild and changes in temperature dry and temperature maximum together with exposure to pollution and high absorption of solar radiation increase the risk of stroke.

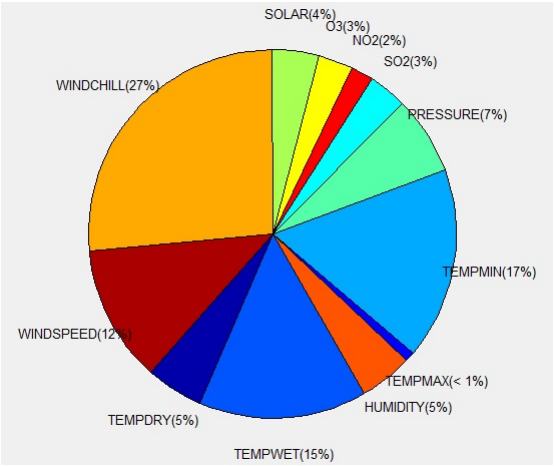

Figure 6.14: Neuron proportion for autumn subjects.

While in autumn (Figure 6.14), we can see the highest neuron activations belonging to wind chill with 27% followed by temperature minimum 17%, temperature wet 15% and wind speed 12%. The autumn season is known as a wet season in New Zealand where rains and storms frequently occur. This indicates that this particu-

lar group is exposed to extreme weather changes before stroke occurrences and this poor weather condition may have worsened their health leading to a stroke.

Winter, summer and spring graphs show a high percentage of neuron activation for solar radiation with 11% percentage of connection weight changes, according to current research, geomagnetic storm influenced the increment of stroke risk [Feigin 2014].

Interaction between each input neuron can be comprehended through analysis of the interaction graph. The interactions between input features are represented by the black lines, where the thicker line means more interactions between those features. The interactions of spike signals depend on the mapping of the input neurons in the reservoir. Based on the input variable mapping (see section 5.4), two graphs will be generated which are the input features similarity graph and the input neuron similarity graph. The spike density of the signals is calculated to produce the input feature similarity graph. Then the input neuron similarity graph is produced to show where input neurons are mapped inside the recurrent 3D reservoir.

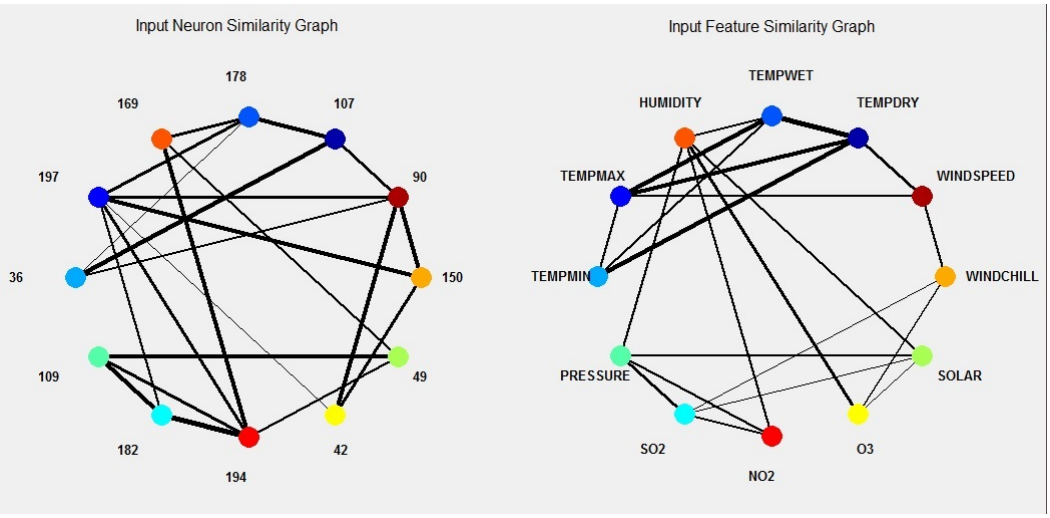

Figure 6.15: Best input neuron mapping for summer.

For example for winter depicted in Figure 6.15 are the best input neuron mapping graphs that have been generated by NeuCube. Analysing this input feature similarity graph further, it clearly shows that the correlations between temperature variables are very strong. So mapping these temperature variables to input neurons in close proximity is vital for better learning of the SSTD patterns. This mapping produced the highest prediction accuracy (100%), showing that temperature

variables have a high influence in predicting stroke occurrences. The input neuron similarity graphs have numbers to indicate which neurons are allocated for those particular features.

Figure 6.16 illustrates the total interaction that occurs between each neuron. Through the interaction graph we can see that input neuron for temperature maximum has a high interaction with the input neuron of NO_2 gas, indicating that these features have a strong relationship with each other. NO_2 also has a high interaction with solar radiation and temperature dry. One study has uncovered that NO_2 gas plays an important role is solar radiation absorption under polluted conditions [Solomon 1999] which is borne out here by the higher level of interaction. Other examples of this interaction are illustrated in Figures 6.17, 6.18 and 6.19 for winter, spring and autumn respectively.

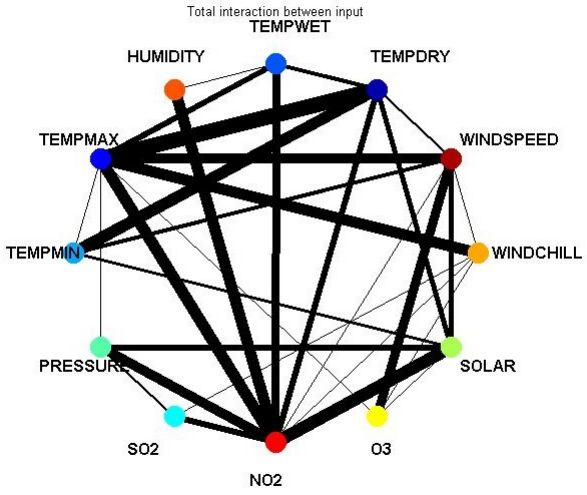

Figure 6.16: Total interaction graph for summer.

In winter (Figure 6.17), temperature minimum has a high interaction with O_3 gas and two other temperatures which are temperature wet and temperature dry. While in spring (Figure 6.18), temperature dry has a very high interaction with humidity, while temperature max has a high interaction with NO_2 gas.

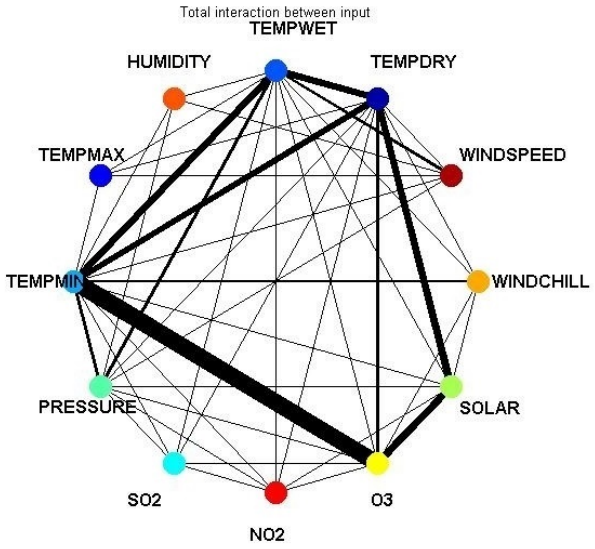

Figure 6.17: Total interaction graph for winter.

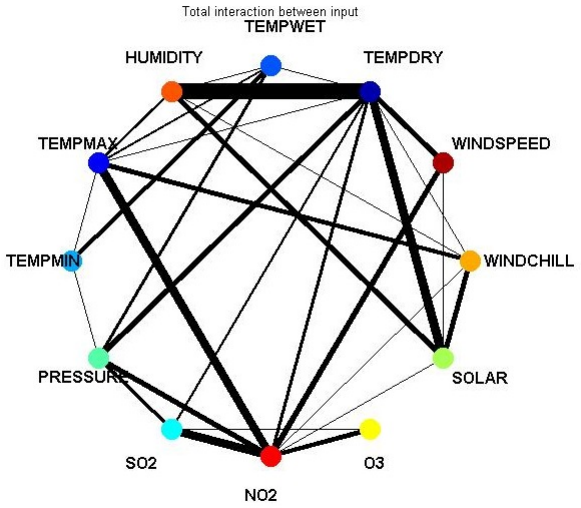

Figure 6.18: Total interaction graph for spring.

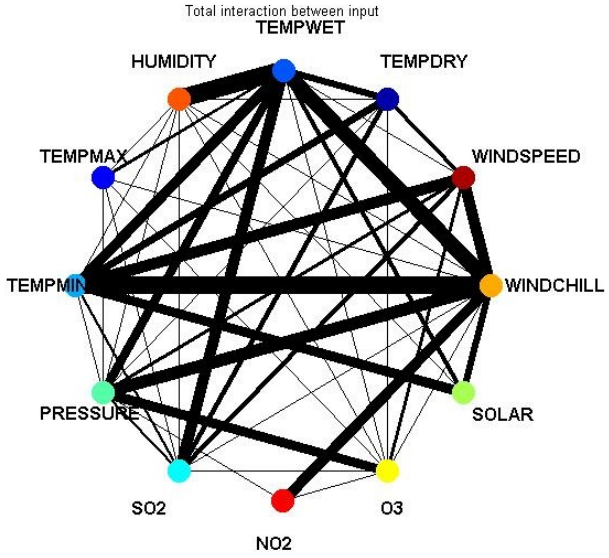

Figure 6.19: Total interaction graph for autumn.

In autumn (Figure 6.19) we can see, wind chill has the highest interaction with temperature minimum, followed by temperature wet. This graph also revealed high interactions between temperature wet and humidity. Wind chill also has quite a high interaction with atmospheric pressure.

6.3.6 Personalised Level Network Analysis

Other than analysing the network as a whole, through NeuCube the network of neurons can be analysed individually for each subject. The input features are ranked based on weighted neurons. The more the neuron spikes the larger the neurons connection are weighted and the brighter the neurons' color in display. The firing order among the input neurons are recorded and represented by a colored number where the darker the blue color means the higher the rank of firing. The relative firing order of input neurons revealed which input neuron is initially triggered for certain individuals that influences whether that individual is at low risk or high risk of stroke.

We will discuss only two cases for personalised level network analysis, covering the summer and spring seasons. The first example is a subject taken from the summer season. Figure 6.20 shows the network of trained neurons for subject 20 who is a woman aged 55 years, having a history of hypertension and never smoked.

She was classified as high risk of stroke when temperature maximum and NO_2 neuron is weighted larger and wind chill are triggered first followed by atmospheric pressure, O_3, NO_2, temperature dry, temperature wet, temperature maximum, wind speed, SO_2 and humidity (see Figure 6.20b). In total ten (10) input neurons are triggered for this subject to be classified as high risk while only nine (9) input neurons are triggered when the subject was classified as low risk (see Figure 6.20a) when temperature maximum was not triggered at all.

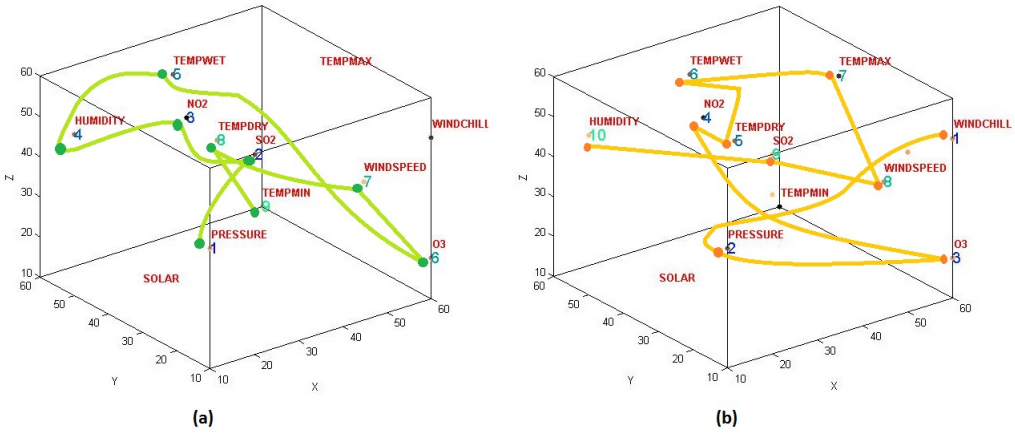

Figure 6.20: Individual analysis of subject 20 for summer season in (a) low risk class, and (b) high risk class.

Figure 6.21 depicts a personalised level network for subject 1 who is a woman aged 49, that also has a history of hypertension and never smoked. The subject was classified as high risk of stroke when O_3, temperature wet, humidity and wind speed neurons are weighted larger (refer to Figure 6.21b) compared to the same neurons when the subject was classified as low risk (see Figure 6.21a). The first input neuron triggered when the subject is classified as high risk was O_3 then followed by temperature wet, humidity, temperature minimum, temperature dry, wind speed, SO_2, NO_2, wind chill, solar radiation, temperature maximum and atmospheric pressure. All input neurons (12) are triggered in the high risk class compared to the subject in low risk class where only eleven (11) input neurons are triggered and most of the input neurons are weighted small. The high risk network suggests, the subject was exposed to high pollution of O_3 gas before stroke occurrence. All of these figures suggest collective changes in input neurons influence the increment of stroke risk.

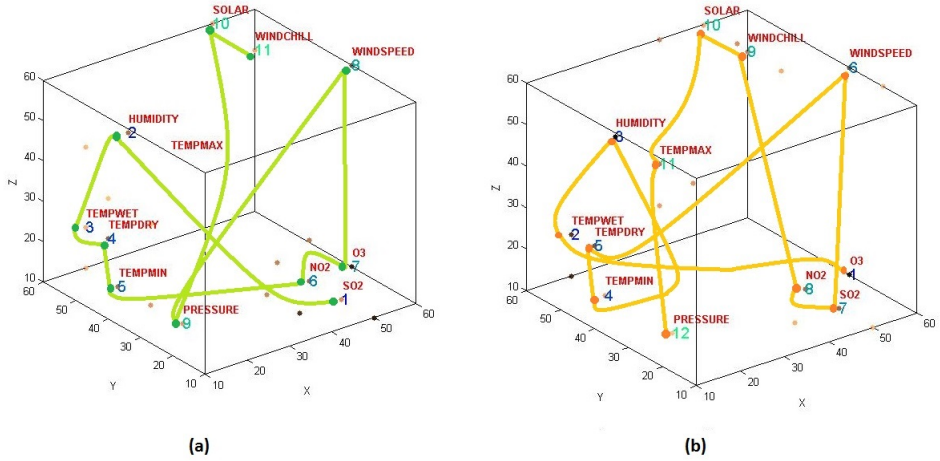

Figure 6.21: Individual analysis of subject 1 for spring season in (a) low risk class, and (b) high risk class.

All of this visualisation has revealed new knowledge about the relationship between environmental features and subjects in the given time windows. We also can determine which features are dominant in a particular season, which input feature neurons fire earliest and which input neurons spike the most for a particular individual. As a result new understanding has been gained about the environmental contributions that trigger stroke on a personalised level.

6.3.7 Seasonal Variation Analysis

It could be argued that each season forms a different population grouping due to its different environmental conditions. To determine whether any differences were significant or not a two sample t-test with unequal variances was carried out. Using this t-test with all variant couplings of seasons, it was found that it is possible to group the seasons in to two separate populations, see Table 6.3. Firstly summer and spring, and secondly winter and autumn. As can be seen from Table 6.4 seasonal comparisons outside these groupings did show a statistically significant difference in population means, with an alpha value of 0.05. When comparing the standard deviation of the environmental data within each season autumn showed the largest value at 4.19, see Table 6.5. This indicates that there is a greater level of variation within the data in this season than in any of the others, which may in turn affect the model built upon this data and its accuracy of prediction.

Table 6.3: Results of t-test for significant difference between seasonal groups

	Summer	Autumn	Winter
Spring	No Difference	Difference	Difference
Summer		Difference	Difference
Autumn			No Difference

Table 6.4: Two sample t-test with unequal population variances

		t-Value	Degree of Freedom	p-Value
Summer	Spring	-0.2844	68.1202	0.7777
	Autumn	3.1458	23.0886	0.0045
	Winter	2.4985	61.0024	0.0152
Spring	Autumn	3.3944	20.9996	0.0027
	Winter	3.0028	43.8541	0.0044
Autumn	Winter	1.7578	20.168	0.0941

Table 6.5: Mean and standard deviation for environmental data within each season

	Mean	Standard Deviation	Population Size
Spring	122.72	2.37	26
Summer	122.52	3.58	46
Autumn	118.83	4.19	16
Winter	120.82	1.92	20

Each of the two major seasonal groups has subjects with an age range of 50 to 70 years along with a younger band. The difference is in the spring subjects with an age range of 35 to 50 years and autumn with an age range of 25 to 35 years. The age range for autumn subjects may also be a significant factor in the difference of model accuracy. This factor is equally likely to be balanced with individual lifestyle factors as well as the environmental ones.

6.4 Chapter Summary

The implementation of the stroke occurrences case study through PMeSNNr method and NeuCube system has proven its capability to analyse a complex SSTD problem. Through NeuCube visualisation functions underlying patterns and relationships amongst environmental features are revealed and can be analysed further. The novelty of this research is in proving there are hidden associations amongst the environmental features that increase the risk of stroke for an individual. The new knowledge revealed is interpreted based on computer science perspectives and limited scientific knowledge. Since previous scientific knowledge about the influence of these environmental features is either limited or non-existence, except for some studies confirming certain weather features do influenced stroke incidents. Nonetheless this research has successfully achieved the objectives by predicting accurately the risk of stroke occurrences at the earliest time point at 6 days prior, and demonstrates that analysing all the features collectively can accurately predict stroke risk. We also confirmed that there is a cascading effect unique for each individual depending on their exposure to certain environmental factors within a specific time window. To further clarify these relationships and its cascading effects, further scientific research needs to be done. Enhancement of the optimisation functions is essential to minimise computation time in obtaining an optimised model.

Chapter 7

NEUCUBE-BASED DATA MODELLING FOR ECOLOGICAL EVENT PREDICTION

"Without knowledge action is useless and knowledge without action is futile."

- Abu Bakar

7.1 Introduction

In ecology, certain events involving sudden outbreaks or extinction of a species in specific geographical location are significant. The events not only contribute to new discoveries in scientific knowledge but also its direct association with economic consequences. Events like global warming lead to changes in climatic conditions. One of the consequences of climate change is that production losses caused by some pests and pathogens will increase [Finlay 2011]. The causes of these increases change, and often involve complicated relationships.

Take for example, the outbreak of some species of Aphids more precisely Rhopalosiphum padi, can pose significant increase in damage to autumn agricultural produce, particularly wheat crops. Rhopalosiphum padi transmits Yellow Dwarf Viruses (YDV) that destroys wheat crops. Thus, the prediction of an aphid outbreak in a harvest season is an important part for timely agricultural management [Stufkens 2000], [Teulon 2008]. Predicting such events requires knowledge of the relevant species and their environment. The ecological study of a species and their environment covers a wide range of topics, such as the population dynamics of a species in a certain

geographical location, the patterns and changes of species distributions and the characterisation and the changing aspects of their habitual environments that influence the population of the species. This knowledge can be applied to make timely predictions that help reduce the damages caused by such pests.

Following a brief introduction to the aphid species, this chapter reviews various studies that analysed the complex and dynamic relationships between aphid and their environment. Most of these studies use conventional statistical methods for predicting aphid outbreak, which rarely give a satisfying answer. This chapter also summarises a case study of the species of aphids called the bird cherry-oat aphid (Rhopalosiphum padi) caught in suction trap located in Lincoln, Canterbury between 1982 and 2000 utilising NeuCube as analysis tool. This case study was published recently in the 2014 International Joint Conference on Neural Network (IJCNN) [Tu 2014] to explore the feasibility of the NeuCube system.

7.2 Review of the Aphid Species

7.2.1 What is an Aphid?

Aphids are small soft-bodied insects that feed by inserting their slender mouthparts into phloem cells, the food conduits of plants [Stern 2008]. Certain aphid species feed on specific species of plant. Once an aphid finds the correct plant species it would feed and produce offspring that then spawn into large colonies. A newborn aphid sheds their skin four times before becoming an adult aphid. Most aphids are parthenogenetic reproduction cases where they are born pregnant and produce females without the intervention of males. Only once a year in autumnal conditions, aphids will mate and produce offspring. Aphids are masters of polyphenism where they can switch between sexual and asexual reproduction. Most females look like an asexual female that produce eggs, while some females undergo meiosis to produce males. Another example of polyphenism is that aphids can produce up to eight different alternative morphs or shapes. For example, aphids that develop without wings focus in producing more offspring, but when the colonies grow and attract the attention of predators, they produce winged offspring that can fly to new plants. About one percent of the 4,000 species of aphids produce a specialized soldier caste

that fights predators through either lethal injection of protein-digesting cocktail jab with their sharp horn, or they squeeze their enemy into submission [Stern 2008].

Aphids are likely to be effected by climate change given their short generation times, low developmental threshold temperatures and efficient dispersal capabilities [Bezemer 1998], [Harrington 2007] in [Stern 2008]. As climate warms, pest insects like aphids become more abundant [Fuhrer 2003].

7.2.2 Overview of Rhopalosiphum Padi

Rhopalosiphum padi (R. padi) commonly known as the bird cherry-oat aphid is one of many species from the family Aphididae characterised as herbivorous that feed on plant phloem with a rapid turnover of generations [Stern 2008]. R. padi listed among the 14 aphid species considered to have the most agricultural significance [Blackman 2007] because of its worldwide distribution. Polymorphism in R. padi is reduced to live-bearing winged (alate) (Figure 7.1) and non-winged (apterae) females (Figure 7.2).

Figure 7.1: R. padi winged (alate) females.

Figure 7.2: R. padi non-winged (apterae) females.

The R. padi life cycle can be varied according to climate. As depicted in Figure 7.3, an aphid populations commonly reproduce combining sexual and asexual phases (holocycly) alternating between winter tree hosts and secondary summer grass host. However under specific circumstances such as where the primary host is scarce and winters are mild, R. padi is completely parthenogenetic (anholocyclic) [Finlay 2011]. Asexual reproduction occurs continuously with the winged females colonising cereal crops when available in the autumn and winter then moving to grasses during the spring and summer as depicted in Figure 7.4 [Finlay 2011].

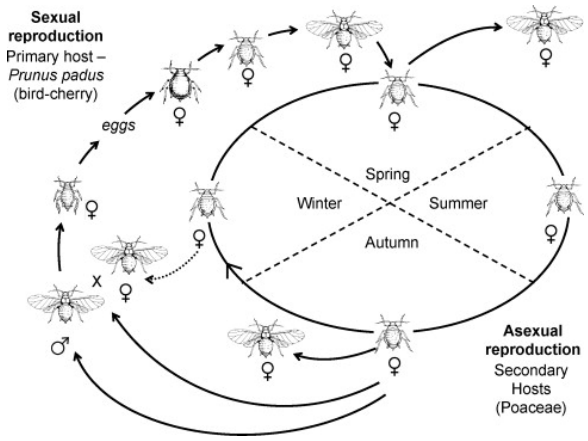

Figure 7.3: Holocycly life cycle [Finlay 2011].

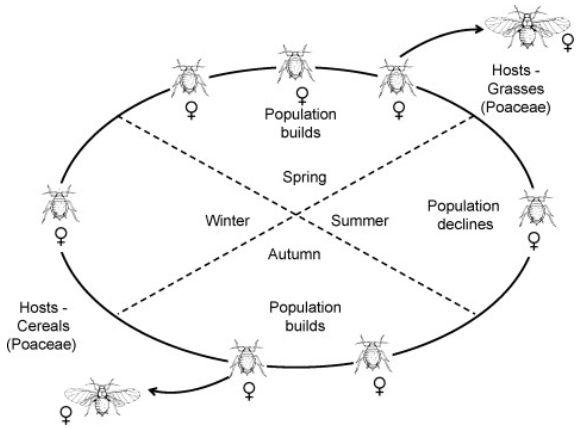

Figure 7.4: Anholocyclic life cycle [Finlay 2011].

Around 50% of insect-vectored plant viruses are transmitted by aphids [Nault 1997], [Brunt 1996] in [Ng 2004] and many of these cause diseases to major economic crops. The greatest causes of yield loss in New Zealand wheat crops is barley yellow dwarf virus (BYDV) which is vectored by several aphid species [Lankin 2001],

[Worner 2002]. One of the important species that transmit this virus is R. padi. In New Zealand, more precisely the Canterbury region, R. padi typically has a bi-model seasonal flight pattern with distinct peaks in autumn (April, May) and late spring (October) [Teulon 1999] in [Lankin 2001].

7.2.3 Factors Impact on R. Padi Population

Various factors directly or indirectly impact R. padi abundance and dispersal. Many has investigated the direct influence on R. padi development and behaviour using climate change variables either from temperature increment [Bale 2002], [Hazell 2010], [Jamieson 2012], elevated CO_2 [Sun 2011], [Ryan 2014], [Newman 2003], or altered precipitation patterns and an increased incidence of severe weather events (see [Finlay 2011] and references therein). By way of illustration in Figure 7.5, [Finlay 2011] has identified what is known regarding direct impacts of R. padi as a result of climate change, and the indirect impacts through climate-induced changes to the interactions between R. padi and the virus along with R. padi and the host plant.

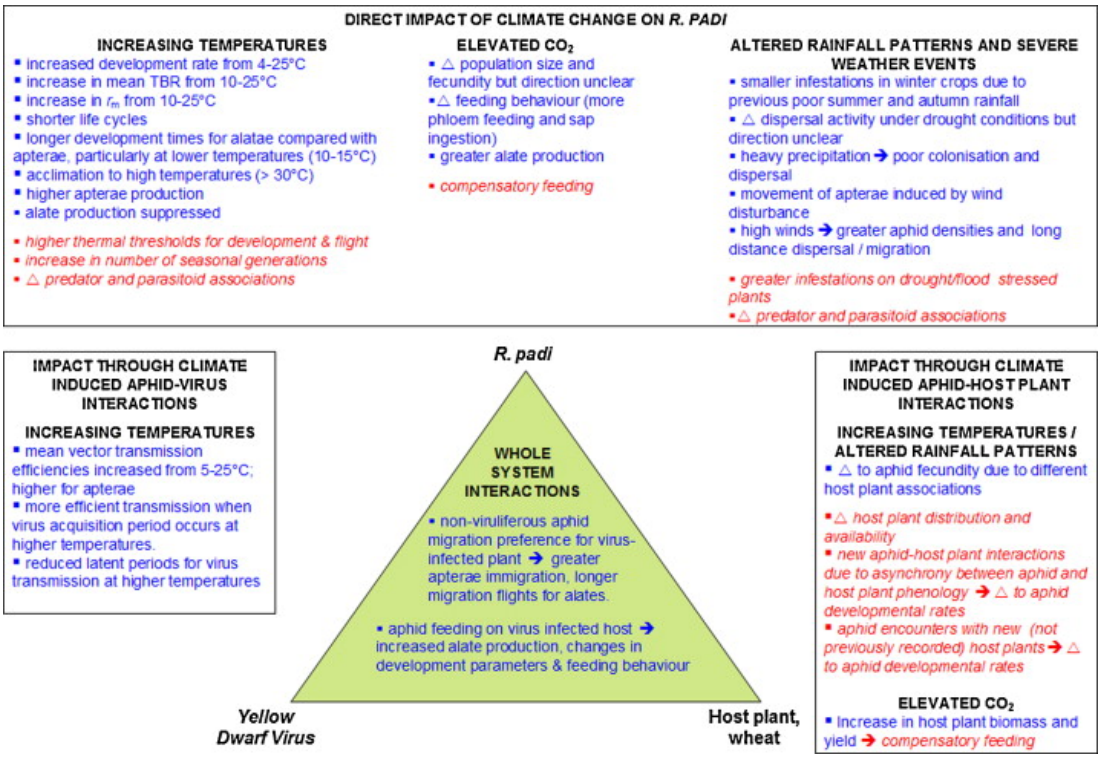

Figure 7.5: Observed (blue text) and expected (red text) impacts of climate change either directly on the aphid vector (Rhopalosiphum padi) or indirectly though the aphid-wheatvirus pathosystem interactions [Finlay 2011].

R. padi has become a global pest that threatens major food crop production. Review at Finlay's paper revealed significant gaps in knowledge of the pests' basic biology making the potential impact of climate variability harder to interpret. For example the paper raises question regarding development and behaviour of R. padi under higher temperature regimes where it is vital to identify the thermal thresholds for development, reproduction, fecundity, persistence, dispersal, flight and morph determination at higher temperatures (beyond 26°C) particularly under realistic diurnal temperature regimes to allow more accurate predictions of the timing and magnitude of dispersal events. The paper also revealed the need to clarify any compensatory feeding response, as well as the magnitude and direction of the developmental and reproductive response, of R. padi feeding on different species of CO_2-enriched plants. Also emphasized was the usefulness of defining any impact of severe weather events such as drought, increased wind speed; on R. padi colonisation, dispersal and migration. Furthermore, the paper suggested that it is useful to determine the extent and magnitude of potential distributional shifts in host crop and non-crop plant species and whether this could lead to new host plant encounters or changes in host plant preferences including host plant virus-encounters that result in vegetation shift. Another important factor is the response of predators and parasitoids of R. padi to large scale vegetation shifts.

Based on previous researches, it is clear that the aphid problem is complex and no single factor can be the determinant of aphid abundance and dispersal events. In order to integrate all the influences on aphid populations, a predictive model need to be developed. This predictive model will assist in understanding the overall inter-relation between factors that leads to such events. Furthermore, the predictive model aids in timely agricultural management to reduce aphid outbreaks in the future.

7.3 Case Study on Aphid Prediction

Various initiatives has been taken to forecast either aphid outbreak or flight/migration patterns through statistical methods such as linear regression, linear discriminant analyses [Klueken 2009], multiple regression [Howling 1993], artificial neural net-

works [Lankin 2001], [Worner 2002] and Simple Evolving Connectionist Systems (SECoS) [Watts 2007a], [Watts 2007b]. Although artificial neural networks performed better than other statistical methods, the method is unsuitable in their current implementation to analyse temporal variables which are used to do the prediction of pattern. Only certain vectors are taken into consideration that fulfill particular conditions, for example weather sequences that had a correlation coefficient (r) higher than +0.65 or lower than -0.65 were chosen [Worner 2002]. This indicates that the weather variables not satisfying this condition will not be included in the modelling. The result from ANN methods provided a good prediction of the expected size of aphid populations but the continuous temporal values related to that condition are not included in the model. Classical machine learning methods have limited capabilities in analysing temporal data as continuous data and are unable to incorporate all aspect of the temporal data that will still result in unsatisfying solutions. Thus, for event prediction in ecology which involves handling complex SSTD, a suitable methods and computational models are always in demand.

The case study of aphids in this chapter will focus on aphids caught in the suction trap located in Lincoln, Canterbury [Worner 2002]. The numbers of aphids caught is positively correlated with the damages caused by their presence in the field regardless of local variations in flight activity [Teulon 2004a], [Teulon 2004b]. The numbers of aphids caught in the suction traps can be used to estimate the aphid populations in the region investigated by previous researchers [Worner 2002], [Watts 2007a], [Watts 2007b]. The time scale for prediction was on weekly basis. However the aphid outbreak can persist for a longer period of time and has been observed that there is a distinctive periodic pattern of development where the aphid count quickly increases and decreases corresponding with seasonal changes. The population increased during autumn and spring, and declined in winter and summer. Even though it is clear that the life span of an aphid is within season, the periodical changes of the aphid count over different seasons within a year can be observed. Thus, the population variation can be described by yearly data and the pattern of aphid development in a year can be treated as a complete period.

In New Zealand, aphids can pose significant danger to autumn agricultural produce. Thus, the prediction of a possible abundance of aphids in the autumn season

becomes important. The numbers of autumn aphids are considered to be correlated with environmental conditions and the numbers of aphids in the previous spring [Worner 2002]. This suggests that the state of an aphid population in the autumn season is influenced by both the previous state of the aphid population in spring and external environmental conditions.

7.3.1 Data Description

The raw data which will be used in the prediction study includes two sets of temporal data. One set is weekly aphid trap catches recorded by Crop & Food Research, Lincoln, Canterbury, New Zealand. The other is made of weekly records for weather variables at the Canterbury Agricultural Research Centre, Lincoln, New Zealand. Data of weekly aphid caught in the suction traps in Lincoln, Canterbury, was collected for the same study period as the weather data. To study aphid in a season, the whole data set for the period of 1981-2000 was segmented into yearly periods. Each of such time periods covers 52 weeks and a data segment for an individual will be yearly data. For each year, the aphid counts rise in both spring and autumn seasons. In spring, large numbers of winged aphids are trapped in late November and early December, but numbers drop off dramatically by late December or early January (i.e., late spring and early summer). Then the numbers of aphids start to increase again in late summer or early autumn. In autumn, the aphid patterns are rather complicated. In some years the earliest small spikes can be observed around week 9 or 10. Then there are also relatively small spikes around week 14. The most significant spikes of autumn aphids are around week 17 and week 20. Occasionally, in some years, the large spikes were reached at week 22 or later before easing off into the winter season.

Based on the weekly records of aphids for the whole period of 1981-2000, the patterns of aphids in spring and autumn in these yearly segments were examined and categorised. The categorisation of the yearly data takes into consideration both the timings of the peaks of aphid counts in seasons as well as the abundance of aphids in the same season. The abundance of aphids in a season were measured based on weekly counts, the number of aphids over variously specified time windows (i.e. aphid counts over more than one week), and the total aphids in the seasonal

period. Out of the total 19 years, 11 years were selected for further experiments with NeuCube. These 11 years are considered to have similar spring aphid patterns. Each of the time periods will start from week 31 to week 30 the following year. Their peak weeks of aphids appear at week 45, 46, and 47, centered at week 46. But the 11 years also have different autumn patterns (see Figure 7.6). Based on their patterns of autumn aphids as shown in the data, these 11 years are further classified into two groups. One is for aphid autumn patterns which are associated with medium-to-high numbers of aphid in autumn and medium-to-high number of aphid in spring (Class 1). The other represents the aphid autumn patterns which are associated with low numbers of aphid in autumn and medium-to-high number of aphid in spring (Class 2). The autumn patterns of aphids of these 11 years will be learned by the NeuCube. For any future year, when its spring aphid numbers peak at about the week 46, the potential autumn aphid pattern can be predicted.

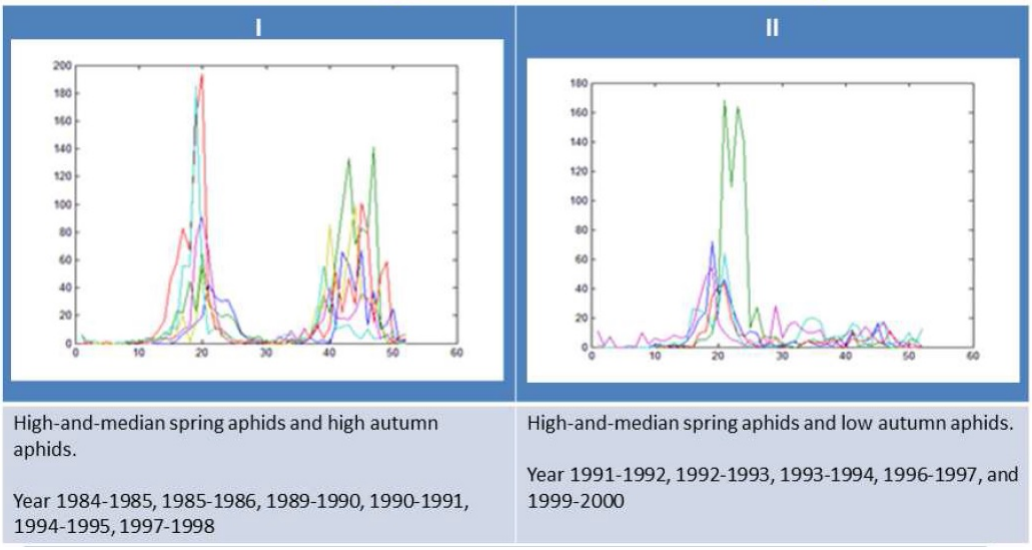

Figure 7.6: Two categories of autumn aphid patterns with similar spring time patterns, categories are defined in terms of numbers of aphids caught in the suction trap, Lincoln, Canterbury.

In addition to the weekly numbers of aphids caught the following raw weather data is recorded:

1. Average rain fall (mm).
2. Cumulative weekly rainfall (mm).

3. Maximum air temperature (°C).
4. Minimum air temperature (°C).
5. Mean air temperature (°C).
6. D-days cumulative temperature for the week (°C).
7. Grass temperature (°C).
8. Soil temperature at 100 cm below ground (°C).
9. Penman potential evaporation (mm).
10. Vapour pressure (hPa).
11. Potential deficit of rainfall (i.e. accumulated excess of Penman over rainfall).
12. Solar radiation (MJ/ m).
13. Wind (km/day).

Some pre-processing has been applied to the raw dataset used in this experiment. Firstly, some errors in the original data records are addressed, such as the negative records for solar radiation. Secondly, pre-processing is carried out in order to take into consideration the relevant results from previous research on aphid prediction. Specifically, for each of the 13 environmental variables, six derived variables were calculated. According to [Worner 2002], during the flight period for each year, the number of aphids caught in a week was correlated with the mean value of each weather variables over the previous N weeks, where N can be from 1 to a maximum of 10 weeks. In this study, for each environmental variable four variables were generated for the averages of the previous 1, 2, 3, and 4 weeks. Furthermore, according to [Watts 2007a], it also appears that the numbers of aphids caught in the suction trap are closely related to the changes of some weather variables, rather than the exact measurements in the corresponding weeks. Thus, two other variables the first order derivative and second order derivative are also generated. Among these variables, 14 of them are selected for the current study, which based on two rules.

- Rule 1: for each of the original weather variable, either the original variable itself or one of the six generated variable should be selected. Thus, the attribute dimension which is represented by them can be kept.
- Rule 2: to select a variable from either the original variable or its six derived variables, the one with the maximum correlation co-efficiency with aphid counts will be chosen.

For the following analysis, 14 variables are chosen, which are listed below:

1. Average rain fall (AR, mm).
2. Cumulative weekly rainfall (CR, mm).
3. Maximum air temperature (MaxT, °C).
4. Minimum air temperature, average of two weeks (MinT, °C).
5. Mean air temperature (MeanT, °C).
6. D-days cumulative temperature for the week (DCT, °C).
7. Grass temperature, average of four weeks (GT, °C).
8. Soil temperature at 100 cm below ground (ST, °C).
9. Penman potential evaporation (PPE, mm).
10. Vapour pressure, average of five weeks (VP, hPa).
11. Potential deficit of rainfall (PDR), first order derivative (i.e. accumulated excess of Penman over rainfall).
12. Solar radiation (SR, MJ/ m).
13. Wind (WR4), average of four weeks (km/day).
14. Wind (WR5), average of five weeks (km/day).

7.3.2 Experimental Design

Three experiments were conducted on this real world aphid data set to show the validity of using NeuCube and how early our model can achieve a good prediction. In the first experiment, we aim to show the predictive ability of NeuCube for full time length variables. The whole time length is used, i.e. 52 weeks, for both training and testing under the assumption that a perfect weather forecast for the autumn season can be obtained which is an ideal case, but not a realistic one. In the following two experiments, we aim to show the predictive ability of NeuCube and how early the model can predict spikes in aphid populations. In these two experiments, NeuCube is trained with the use of 100% time length samples (full year), but only 80% and 75% of the yearly data was used to predict the aphid population pattern in the last 25% year, as illustrated in Figure 7.7. The blue bars represent training data length and the yellow bars validation data length.

In all these experiments, the size of the SNNr is set to 2000 neurons ($5 \times 20 \times 20$) and a 1-NN-deSNN network is selected as the classifier [Kasabov 2013]. It is trained and tested in leave-one-out cross validation (LOOCV) mode. This is based on the assumption that the climate change trend during these 11 years can be ignored and the weather variables are independent between different years.

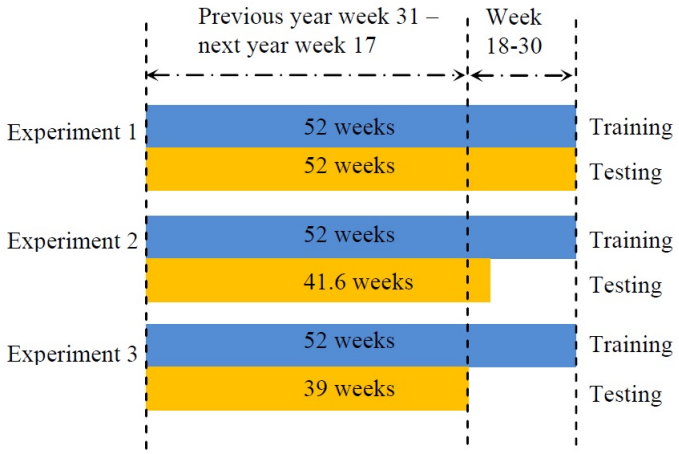

Figure 7.7: Experiments design for NeuCube. Blue bars represent the time length of training samples and the yellow bars represent the time length of testing samples.

Three experiments are design to compare between traditional modelling methods and NeuCube modelling for early event prediction. Multiple-linear regression

(MLR), support vector machine (SVM), multilayer perceptron (MLP), k-nearest neighbors (kNN) and weighted k-nearest neighbors (wkNN) are used as the baseline algorithms. The three experiments are as illustrated in Figure 7.8. Noting that for these baseline algorithms, the time length of training samples and testing samples has to be the same as these methods cannot tolerate different lengths of feature vectors for training and recall.

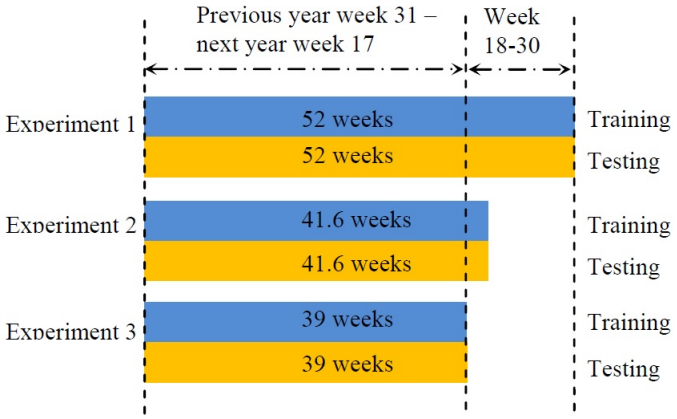

Figure 7.8: Experiment design for baseline machine learning algorithms. Blue bars represent the time length of training samples and the yellow bars represent the time length of testing samples.

The data set for experiment 1 was prepared by concatenating the weather variables one after another to form a long feature vector for each sample, as shown in Figure 7.9. For each weather variable, there are 52 data points in one year (one data point per week), so the final feature length counting 14 variables will be 578. For experiments 2 and 3, we just remove the corresponding percentage of data points from the back of each year data from experiment 1 and then perform the same training and testing procedures as in experiment 1.

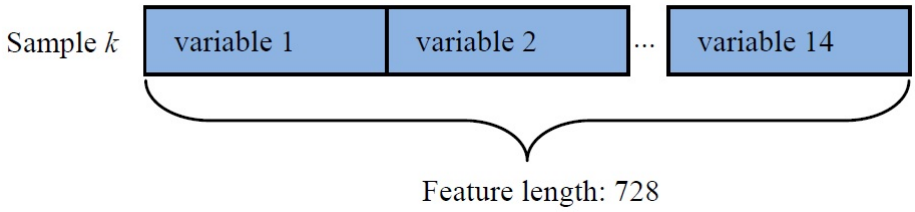

Figure 7.9: Data set preparation for baseline algorithms.

We tune the parameters of the baseline algorithms using a grid search optimisation algorithm yielding the final parameter settings:

- polynomial kernel with degree 2 for SVM;
- 30 hidden neurons and 500 training cycles for MLP;
- k=5 for both kNN and wkNN.

7.3.3 Result and Analysis

The early event prediction accuracy on the aphid data set is shown in Table 7.1 below. NeuCube experimental results showed better performance for early event prediction compared to other classical modelling methods. A realistic early event prediction should be that as the time length of observed data increases, the prediction accuracy will also increase. However for classical modelling methods, we can see that as the time length of training data increases, these methods do not necessarily produce better results (some even worsen), because they cannot model the whole spatio-temporal relationship of the prediction task. They can only model a certain time segment. Because NeuCube models the whole spatio-temporal relationship of the data, even a smaller amount of input data can trigger spiking activities in the SNNr that will correspond to the fully learned temporal patterns resulting in a better prediction with more temporal data entered.

Table 7.1: Prediction Accuracy of Aphid Data Set (%)

Weeks	**52**	**41.6**	**39**
NeuCube M1	**100**	**90.91**	**81.82**
SVM	72.73	72.73	63.64
MLP	81.82	81.82	81.82
kNN	72.73	63.64	63.64
wkNN	72.73	63.64	63.64

Looking at the NeuCube result for the third experiment, 75% of the temporal input data (39 weeks) is used - from week 31 in previous year to week 18 in following year, where week 18 is at the beginning of the autumn period. At this time of the

year an accurate prediction of whether there will be a high aphid population peak in later weeks is very important for both ecology and agriculture, because early actions can be taken to prevent the spread of the aphid population. For this time length, with 75% of time observed, our model can give 81.82% prediction accuracy. In the second experiment, 80% of time length (41.6 weeks) is from week 31 in previous year to week 20.6 in following year, where week 20.6 is in the front part of the aphid flight period. At this time of the year, some small aphid peaks are observed. From this we can see that our model can make an early prediction of this peak before the peak appears. With 80% of data observed (early in the aphid flight period), we can have more than 90% confidence in making an early decision. Furthermore, if new data is collected it can be added directly to the testing sample to give a better prediction without re-training the model using both old and new data as it would be the case with SVM or MLP methods. This is a pivotal difference between the new method and traditional methods such as SVM and MLP.

7.3.4 Network Analysis

Figure 7.10 shows the input variable mapping result of the minimal x coordinate face of the cubic reservoir. Note that the two main groups of weather variables, e.g. temperature (MaxT, minT, MeanT, DCT, GT, ST) and rainfall (AR, CR, PDR), are mapped to nearby neurons. The solar radiation (SR) is mapped in the middle of the temperature variables because temperature is greatly determined by solar radiation.

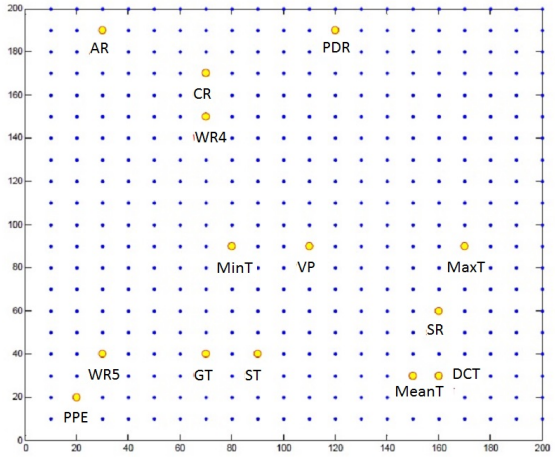

Figure 7.10: Input variable mapping of x coordinate face.

To demonstrate how the optimal mapping suggested by the proposed graph matching algorithm can influence the overall performance, we designed two experiments to compare results of optimal mapping with results of random mapping. In the first experiment, we use the same group of input neurons and run the NeuCube learning twice: in the first run we randomly mapped the features to the input neurons while in the second run we used the proposed mapping to compute the optimal input mapping. This process is repeated 10 times and the accuracies of each run are shown in Figure 7.11(a). In the second experiment, we also run the NeuCube twice in the same way as in the first experiment, but the group of input neurons is randomly generated at each time. The accuracies of 10 times replication of the experiment are shown in Figure 7.11(b).

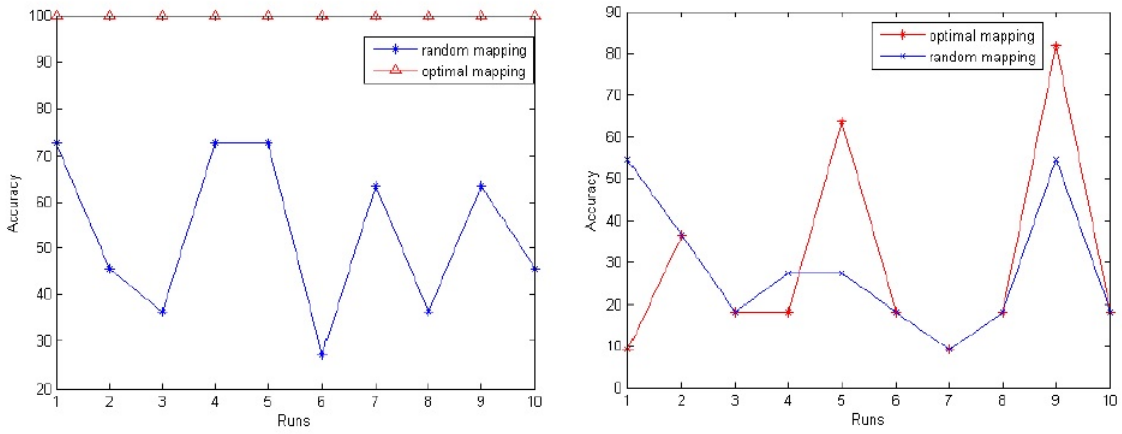

Figure 7.11: Comparative accuracy of pattern recognition using random mapping (in blue) versus the proposed mapping method (in red).

In Figure 7.11(a) the graph matching is obtained through a deterministic algorithm. So given the same group of input neurons, it can always produce the same optimal mapping and the accuracy will not change. But for random mapping, the results change across experiments because each time the mapping is different. This result indicates that input mapping plays an important role for obtaining increased accuracy of the model. In Figure 7.11(b) the group of input neurons is randomly generated at each time. This is why the accuracy of the optimal mapping is lower than the random mapping in runs 1 and 4. In runs 5 and 9 the accuracies obtained

with the use of the proposed mapping are much higher (i.e., 36.36% and 27.27% respectively) than the results with the use of just random mapping. This result also indicates that not only does the mapping play a key role, but the group of input neurons selected is equally important. How to optimally choose a set of input neurons in relation to specific input data is an interesting problem to address in the future, as it os beyond the scope of the present research.

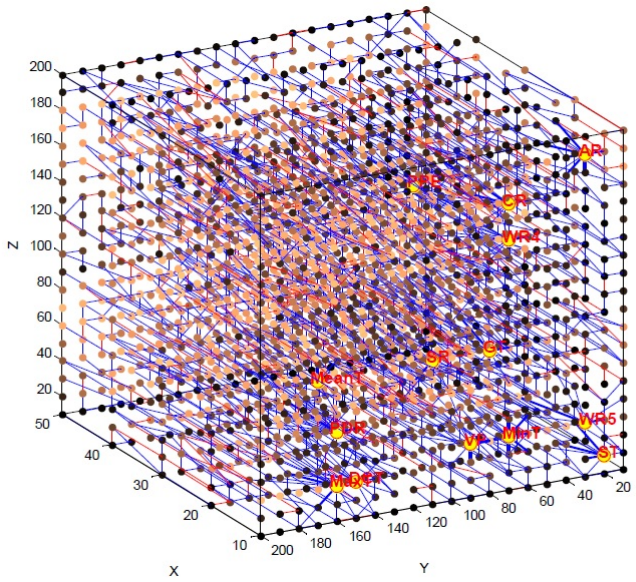

Figure 7.12: Reservoir connections after training. The red are negative weight connections and the blue are positive weight connections.

Figure 7.12 depicts the neuron connections with absolute weights larger than 0.07 after the reservoir is trained in one run, and Figure 7.13 plots the total spike amount of each neuron emitted during training process. In Figure 7.12, blue lines represent connections with positive weight and red lines represent connections with negative weights. The line thickness indicates the strength of the connection. The neurons are coloured to indicate their connection strength with other neurons in the reservoir. Brighter neurons have stronger connections with other neurons while darker neurons have weaker connections. From this visualisation, we can reach conclusions related to a better understanding of the data and the problem:

1. Connections between nearby input neurons are denser than the connections between far away input neurons. This indicates that there are more interactions between input variables mapped to closer neurons.

2. Neurons in the middle of the reservoir are generally more active than those on the edges. This indicates that the neurons in the reservoir are not uniformly activated, which is determinant upon the input data.

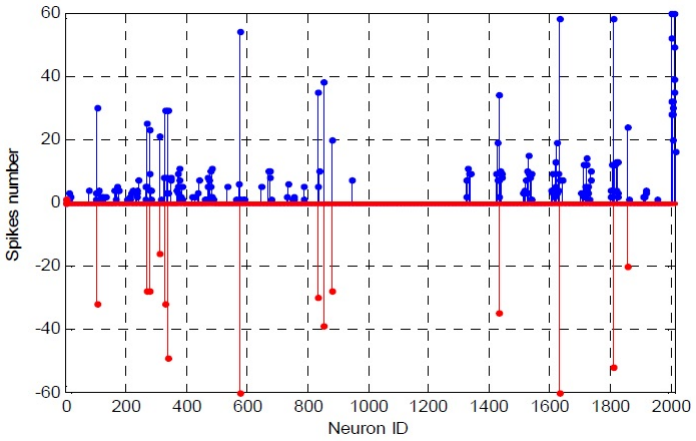

Figure 7.13: Reservoir connections after training. The red are negative weight connections and the blue are positive weight connections.

We can analyze the SNNr structure in terms of information distribution/ spreading [Shrager 1987]. We first treat the input neurons as the information sources in the network and then use the spreading algorithm from [Zhou 2004] to determine neuron clusters belonging to each information source based on the spike transmission in the network. The more spikes that are transmitted between two neurons, the tighter they will be connected together. After this we link a neuron to the one in the same cluster from which it receives most spikes.

Figure 7.14 shows the network structure after unsupervised training. The big solid dots represent input neurons and other neurons are displayed in the same color as the input neuron from which it receives most spikes. The black dots mean there are no spikes arriving at it.

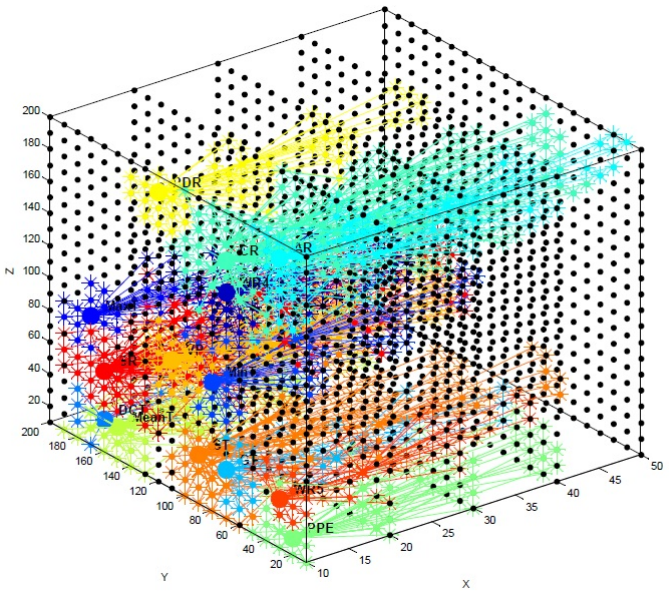

Figure 7.14: The SNNr structure after unsupervised training.

In Figure 7.15, the left pane is the number of spikes for each variable after encoding and the right pane is the number of neurons belonging to each input variable cluster. From this figure we can see the consistency between the input signal spike train and the reservoir structure. It is worth noting that variable 12 (solar radiation) is emphasized in the reservoir suggesting a greater impact of solar radiation on the aphid population. This was also observed in a previous study [Worner 2002].

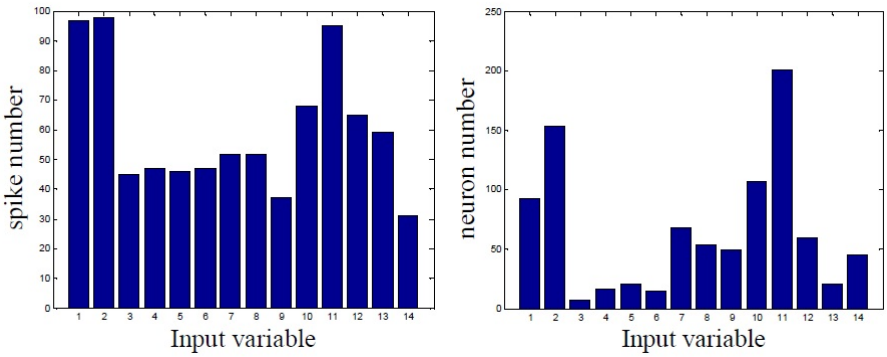

Figure 7.15: Input spike amount of each feature (left) and neuronal connections of each input neuron (right).

7.4 Chapter Summary

The second case study verified NeuCube's capability in modelling any type of spatio-temporal data. A method of mapping input variables into neurons from the SNNr is proposed so that similar input variables based on their temporal similarity, are mapped into spatially closer neurons. The closer the neurons in the SNNr, the more temporal relationships they learn from data. This is illustrated on ecological data in aphid pest abundance prediction. Future work includes experiments on other ecological and environmental data collected in New Zealand, China and other countries related to other ecological and environmental event predictions, such as earthquakes, tsunami, volcanoes.

Chapter 8

CONCLUSION AND FUTURE STUDY

"Research is creating new knowledge."

- Neil Armstrong

This research has presented a novel framework of evolving spiking neural network methods for predictive personalised modelling (PMeSNNr) to model spatio/spectro temporal data. To the best of my knowledge, this study is the first comprehensive study of personalised modelling based on spiking neural network methods that resulted in several contributions to the areas of information science and bioinformatics. The result has been successful feasibility analysis of evolving spiking neural networks for SSTD data analysis and for predictive personalised decision support.

During the course of this study, several novel approaches have been applied, including an evolving spiking neural network system call NeuCube M1, predictive personalised modelling techniques, an innovative input variables mapping techniques utilizing Factor Graph Matching (FGM) algorithm, a novel extended dynamic evolving spiking neural network method called deSNNs_wkNN for multi-NN classification and regression problems, and visualisation of spiking networks on group and personalised levels. The methods have been applied to two real world case studies involving the need for accurate early predictions, in determining stroke occurrences and aphid pest population size. This research is not the end, but just a beginning in exploring the field of personalised modelling based on spiking neural networks for knowledge discovery.

8.1 Summary of Thesis

Every research endeavor starts with objectives that guide the direction of the work. The ultimate objective of this research was to develop a novel framework of an information method and system to analyse SSTD for personalised knowledge interpretation and prognosis. Within this the main objective was to develop a generic modelling environment to analyse SSTD (medical, brain, financial, geological or ecological data, etc.) using personalised modelling based on spiking neural network methods.

In brief, this thesis has presented the following main contributions for modelling and analysing SSTD:

1. Analysed the problems related to global modelling and conventional PM for SSTD analysis, identified areas of where shortcomings are present, and proposed potential solutions;

2. Participating in the development of a novel integrated evolving spiking neural network system module called NeuCube M1; including a new multi-NN algorithm for classification and regression problem, a predictive personalised method for early event prediction, input variable mapping and network visualisation.

3. Application to stroke occurrence for risk prediction;

4. Application to aphid pest abundance prediction;

5. Assisted in identifying areas where further investigation is warranted.

The proposed PMeSNNr framework and NeuCube M1 is the platform that integrates novel machine learning and modelling techniques for specific research problems where the following are desirable:

- predictive personalised modelling;
- SSTD pattern learning;
- classification;
- regression;

- disease risk prediction;
- early event prediction;
- quick and accurate adaptation to new data;
- knowledge discovery and model validation;
- personalised profilling and visualisation.

As an important part in this research, Chapter 4 described a novel PMeSNNr framework and Chapter 5 described the NeuCube M1 system for data analysis and knowledge discovery of spatio/spectro temporal patterns. NeuCube M1 is the first prototype that combines several spiking neural network methods into one system these include:

1. Address Event Representation (AER) method for encoding continuous data into spike trains.
2. Liquid State Machine (LSM) concepts connecting leaky-integrate and fire model (LIFM) spiking neurons with recurrent connections based around a recurrent 3D reservoir.
3. STDP rule implemented as a learning method for unsupervised learning in the reservoir.
4. Several evolving spiking neural network methods are currently available for classification or regression; such as deSNNs, deSNNm, deSNNs_wkNN and SPAN.

NeuCube M1 implemented a new input variable mapping technique that calculates the optimum mapping for SSTD features. The mappings use the graph concept where two weighted graphs are constructed to represent the similarity of spike trains following the principle that highly correlated spike trains should be mapped to nearby input neurons. Then a Factor Graph Matching (FGM) algorithm is used to match the two graphs for optimum mapping that will result in highly correlated features mapped to nearby input neurons.

NeuCube M1 is the first prototype to provide the visualisation tools that enable deep network learning of the spiking patterns. This will help enhance our understanding of the data and process that generated it. The visualisation including neuron spiking state, connection weight change, positive and negative spikes, a neuron's activation level, neuron clusters, neuron interactions, a neuron's firing order and the neuron connection weights for each sample.

NeuCube M1 also provides multi-NN classification/regression through the development of a novel method (deSNN_wkNN) extended from dynamic evolving neural networks. This method offers flexibility during the supervised learning phase and an increased number of modelling outputs.

This study investigated a variety spiking neural network methods for encoding data and classification models during the development of NeuCube M1. Such algorithms for encoding data have included AER, ROC, POC, BSA etc. and classification algorithms including eSNN, eeSNN, reSNN, deSNN etc. Encoding methods are important and their usefulness is dependent upon the data itself; and the purposes for the specific investigation. Classification models are also important, but are not the decisive factor for NeuCube construction.

NeuCube has shown an ability to reveal previously unknown and/or unproven associations amongst features in SSTD. This is borne out in the study on stroke prediction. We also discovered that there is a cascading effect unique to each individual depending on their exposure to certain environmental factors within a specific time window. This current model has successfully predicted the risk of stroke occurrences at an earlier time point and demonstrated that analysing all the features collectively can accurately predict stroke risk. The second case study with ecological data in aphid pest population prediction verified NeuCube's capability in modelling any type of SSTD. The result is the earlier prediction of aphid pest abundance to assist in timely agricultural management.

Our hypothesis outlined in chapter 1 has been verified utilising the two case studies (see Chapter 6 and 7) where:

1. An individual model is created for every new individual is more accurate than a global model.

2. An earlier prediction of a future event is possible through predictive personalised modelling.

3. Analysing all data collectively without the need for filtering of noise with SSTD is more accurate.

4. The visualisation of interaction patterns amongst input neurons assists in increasing human understanding of previously hidden relationships.

It was also demonstrated that NeuCube can analyse SSTD, discover new knowledge and produce a better personalised knowledge representation and risk prognosis for a person/event than classical machine learning methods such as SVM, MLP, MLR, kNN etc.

In summary, personalised modelling based on spiking neural network methods offers a novel integrated methodology that comprises different computational techniques for SSTD analysis and knowledge discovery. Compared with the results obtained by other published methods, our new algorithms and methods have produced improved outcomes in terms of prediction accuracy, earlier prediction and discovered more useful knowledge, because they take into account all aspects of the information dimensionality for each input sample within time and space. This provides more precise information for analysing each data sample. NeuCube is evolving and adaptive techniques in which a new data sample can be continuously added to the training dataset and subsequently contribute to the learning process for personalised modelling. These qualities makes personalised modelling based on evolving spiking neural network methods a promising medical decision support system, especially for complex human disease diagnosis and prognosis such as stroke, cancer, cardiovascular disease etc. Furthermore this proposed method can be applied to other problems that involve SSTD such as earthquakes, volcanic eruptions and other environmental event prediction, ecological problems, contagious disease spread along with many other possibilities.

However, in its current form NeuCube not only creates a unique (personalised) model for each testing data sample, but also a repetitively trained reservoir to learn from each training sample which requires more computational power and performance time than traditional global modelling methods this is especially true to train the models on large data sets. The proposed methods have shown the great potential for solving SSTD problems that require individual testing. This study is the first step in this research direction and needs more in-depth understanding in science for validating the experimental findings and knowledge discovery.

8.2 Directions of Future Research

This section presents some promising future directions for further enhancement of the NeuCube system for personalised modelling. Problems involving SSTD are in principle very challenging due to the dimensional complexity and inconsistency of the data. Even though this study has proposed new algorithms and systems for personalised modelling in SSTD analysis, there are limitations and open research questions to be investigated and solved in future research.

8.2.1 Optimisation Strategies

In this thesis, a grid search method has been used as the technique to find optimal parameters for personalised models. However this method requires expensive computational time and resources where multiple runs need to be executed to find optimal model and parameter settings. Many other optimisations algorithm can be implemented such as a Genetic Algorithm and Quantum-Inspired Evolutionary Algorithm (QEA). GA based algorithms are often criticized for its high computational cost, which results in the difficulty of testing large datasets. QEA present itself as a more promising solution for the two reasons. Firstly, it can dynamically adapt the learning speed leading to a smooth and robust convergence behavior. Secondly, QEA manipulates more complex distributions of solutions than with a single model approach leading to more efficient optimization of problems with interacting variables [Platel 2009]. Saving in computational efficiency has highlight QEA as worth incorporation into NeuCube.

8.2.2 Dealing with Variability in Data and Achieve Consistent Results

Utilising spiking neural networks for near optimal result can be different because of the stochastic nature of spiking neurons including the mapping of input neurons in the reservoir. The application of partial solutions and multiple execution runs for consistency is worth further investigation.

Model prediction accuracy is affected by the variability in the measured data. Further testing can be done on the significance of the contribution of each attribute to prediction accuracy and the inherent natural variability within each attribute. As this variability in environmental factor is likely to be different for each season, seasonal models should be maintained, along with investigation into a combined model of each of the two seasonal groups that showed no significant difference in population (refer Section 6.3.7).

8.2.3 Dealing with Multiple Types of Data

Some research encompasses multiple types of data for example, static data and spatio/ spectro temporal data that describe the new data samples. In this study we only deal with either spatio/spectro temporal data or temporal data. The static data is filtered outside of NeuCube to find the nearest neighbourhood of known samples and then only the temporal data of this neighbourhood is used for modelling. It is suggested as a further enhancement of NeuCube, that the process of selecting the nearest neighbourhood through static data is done concurrently with spatio/spectro temporal data. This can be done by incorporating additional personalised modelling techniques into a pre-temporal data processing stream, or by incorporating the static and dynamic data into a single "chromosome" or multidimensional vector where both the static and dynamic components form an integrated part of the modelling for prediction or classification.

8.2.4 SSTD Representation in Domain Knowledge

Currently SSTD is represented in domain specific knowledge bases, such as ontology databases, and spatio-temporal modelling and spatio-temporal reasoning falls under

data mining research. In the domain area of bioinformatics, the concerns of manipulating SSTD to represent knowledge is crucial because it could lead to the notion of improving and saving lives either for humans, animals or the environment. The concept of ontology is becoming more significant recently because of its knowledge pooling capability highlighting the relationship between ontologies and knowledge bases (KB). These definitions are based on the process followed to build the ontology on the basis of an application knowledge base. One definition is found in the framework of the KACTUS project [Schreiber 1995]: "it [an ontology] provides the means for describing explicitly the conceptualisation behind knowledge represented in a knowledge base" [Corcho 2005]. Ontology can be described as a structural framework to organize data and its relations systematically. However, most ontology are developed heterogeneously and designed for diverse purposes. There is an emerging demand for the integration and exploitation of heterogeneous biomedical information for improved clinical practice, medical research and personalised healthcare [Anjum 2007]. The main issue here is about how ontology can be utilized to represent SSTD since a spatio/spectro-temporal ontology can have many rules of qualitative reasoning on spatial and temporal data which provide a valuable source of domain independent knowledge that should be taken into account when generating and evaluating patterns [Andrienko 2006]. In biomedical areas, the ontology is used as the knowledge base that will not only give a common understanding among the bioinformatics community but also assist them in performing a variety of tasks such as processing and analysing new biological data based on historical stored data and to draw new knowledge from them [Anjum 2007], [Hartmann 2010].

As future enhancement, we suggest the development of an ontology-based system that can model space, time and what is suitable not only when collecting historical data but also continuous real time data. The ontology-based systems architecture should be general so that it provides a platform to model various types of SSTD. The ontology-based system should be application-dependent and linked with NeuCube M1 where it is utilised as a knowledge base and the outcome of the learning from NeuCube can a be updated back into the ontology-based system.

The ontology-based system should use several underlying ontologies that can be useful for representing and understanding the domain knowledge. An integrated

framework of ontology knowledge bases with personalised modelling was proposed by [Kasabov 2008] for bioinformatics decision support which was originally based on previous paper by [Gottgtroy 2006] (see Figure 8.1 and 8.2).

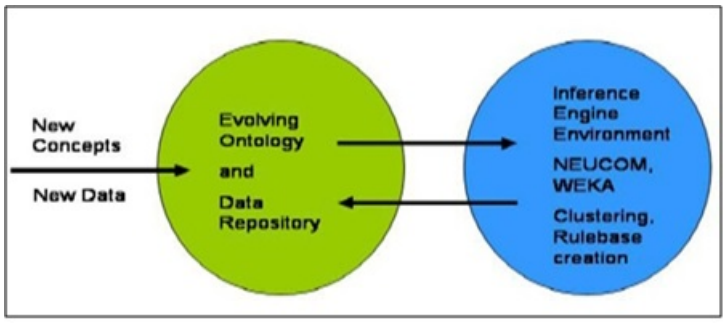

Figure 8.1: An ontology-based personalised decision support framework consisting of two interconnected parts: (i) an ontology/data base sub-system; (ii) a machine learning sub-system [Kasabov 2008].

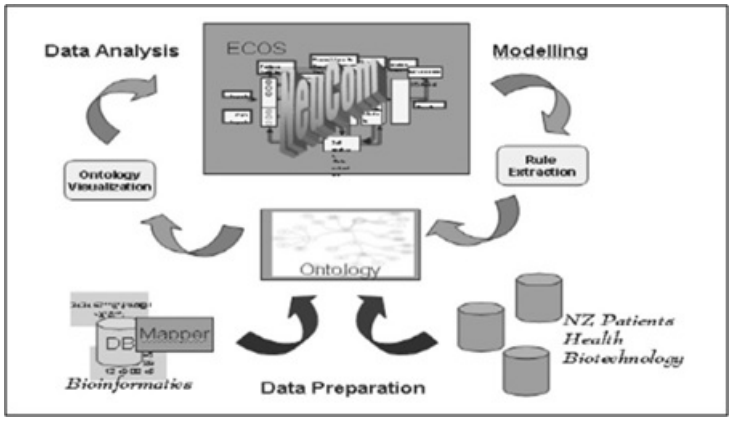

Figure 8.2: A sample ontology-based decision support system. The inference engine at the top utilized data retrieved from an Ontology in Protégé [Gottgtroy 2006].

Ontology integration in the field of Ontological Engineering (OE) has been referred to as several different approaches, such as building ontologies from other existing ontologies and the use of ontologies in applications. Three different situations in which integration has been used in relation to ontology [Pinto 1999]:

1. Integration of ontologies when building a new ontology reusing other available ontologies. The ontologies are publicly and can be extended, specialized or adapted.

2. Integration of ontologies by merging different ontologies about the same subject into a single one that unifies them all. This ontology tries to unify concepts, terminology, definition, and constraints from different ontologies into a single particular representation.

3. Integration of ontologies into applications where knowledge base applications refer to one or more ontologies.

Ontology is commonly built heterogeneously which can then be combined and shared. Figure 8.3 illustrates a proposed conceptual model of a Spatio-Temporal Ontology-based System (STOS) for stroke occurrences case study. STOS adopts the Peuquet Pyramid Framework [Mennis 2000] and the Object Oriented Spatial Temporal Data Model [Li 2002]. The ontology-based system can evolve further by adding more related ontologies that can be queried to learn more about the patient's condition. The ontology-based system could reuse an existing ontology for example the Chronic Disease Ontology [Verma 2009] and extend it by adding a new branch of knowledge for stroke and its temporal data representation. Other publicly available ontologies that can be incorporated inside this ontology-based system are the Time Ontology, Gene Ontology (GO) and Foundational Model of Anatomy (FMA) ontology which can be decided upon during the development stage of the ontology-based system.

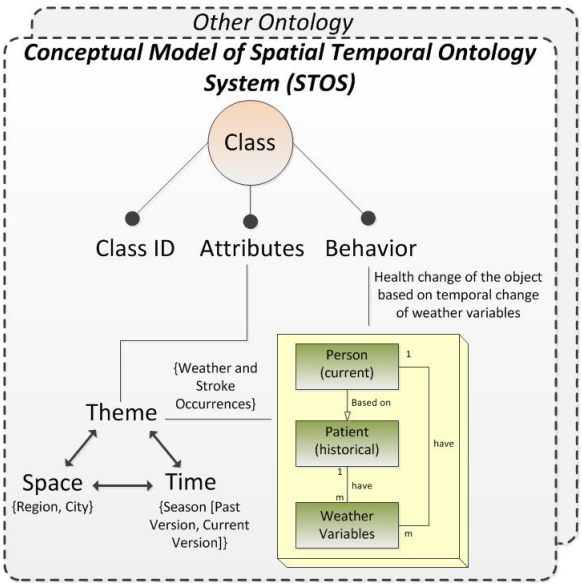

Figure 8.3: Spatial-Temporal Ontology System module.

The ontology integration can adopt a third approach which uses several existing ontologies that underlies the ontology-based system. There is currently no single ontology that unifies all underlying ontologies (see Figure 8.4). The system will query the ontologies on a need-to-know basis and the links between ontologies modifies dependent upon on its context and compatibility.

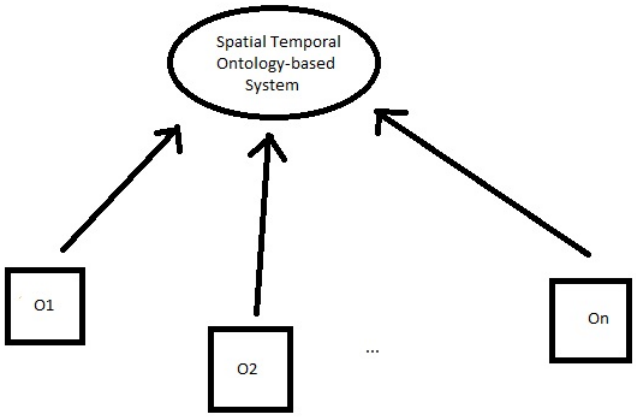

Figure 8.4: Ontology integration approach for use in application.

This ontology-based system should be used as a reference engine to be queried by NeuCube M1 where the ontology must be able to incorporate new findings or relationships that the NeuCube M1 has discovered. There are also several challenges that need to be addressed. Some challenging issues in integrating these two elements are addressed by [Kasabov 2008]:

1. How to use the new knowledge to further evolve the existing ontology?
2. How to link the ontology with other machine learning engine?
   (a) Is specific interface needed (web based or offline)?
   (b) How to make sure the personalised modelling system obtains the right data from STOS?
3. How can changes in a concept be detected?
4. How does the inference engine obtain sufficient information to act in a context-aware manner?

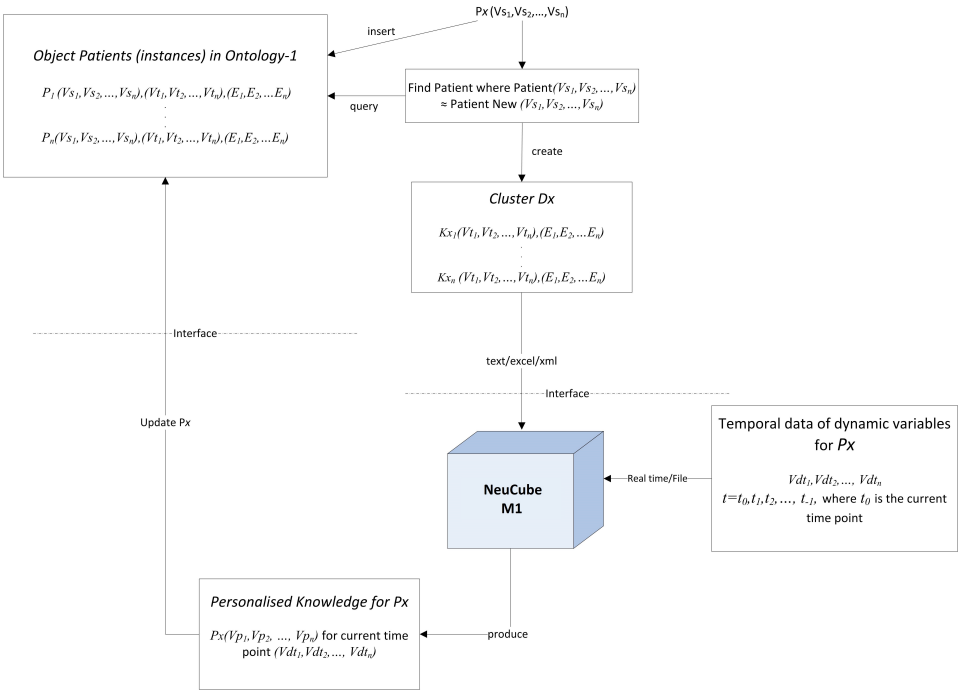

Figure 8.5: Conceptual view of STOS-NeuCube integration.

The integration issues can be addressed conceptually by keeping in mind the technological challenges that need to be overcome in the development stage. In the case study of stroke occurrences, there are several types of historical patient data; static data which can be continuous and/or categorical data and temporal data. This is structured inside the ontology-based system as instances of the object Patient. New patients will be inserted into the object as new instances with a current history of hypertension and other factors that could trigger a stroke become available. Next the ontology will be queried to extract the historical patient data and create a cluster of nearest neighbours which have similar features according to static variables for the new patient. The queried data will then be fed into NeuCube. During the data analysis stage, temporal input for the current time point will be fed into NeuCube together with the temporal variables of the historical data to classify the output class for a new patient. These temporal variables should be real time data streaming and NeuCube should flag a warning if there is a high possibility of a stroke occurring for the new person. Finally the output of the process which is the personalised knowledge consists of the event/class predicted; parameters; and personalised risk profiling for the current time point must be conveyed back into the ontology-bases system. The ontology will grow based on the growth of its instances

(patients) personalised knowledge and this knowledge can be reused for analysing future stroke occurrences for that person or other new patients with similar features. The main idea is

Figure 8.6 depicts a draft of a Patient Ontology for stroke occurrences case study. The coloured area shows the branch of the ontology that will be evolved or updated according to the personalised knowledge produced by NeuCube. This is the first draft of this system and may later become an extension to the Chronic Disease Ontology.

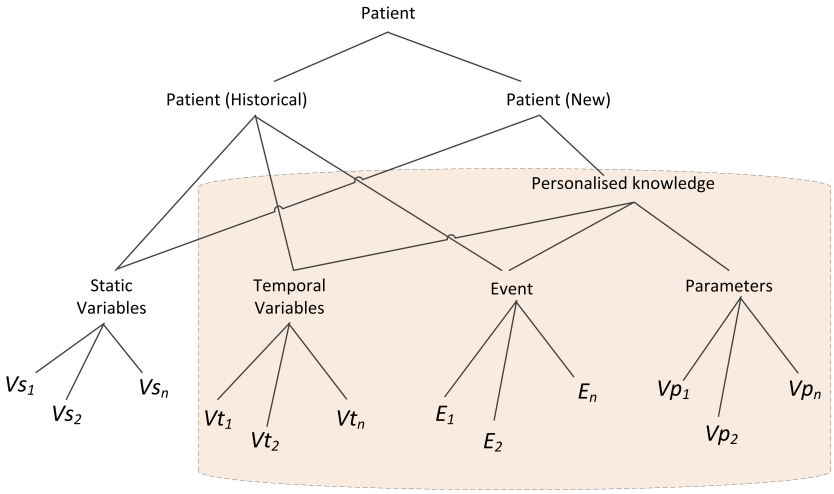

Figure 8.6: A draft of patient ontology for stroke occurrences.

Appendix A

NeuCube Module 1

A.1 Introduction

NeuCube M1 is a Spatio- and Spectro-Temporal Data (SSTD) processing system programmed in the MatLab environment and was developed by a team effort where I have participate in the development. This documentation describes how to use the system to perform supervised classification of SSTD data, as well as the dynamic visualisation of the activities in NeuCube.

A.2 Data Set Format

A valid data set contains three parts: sample data matrix, class label vector and variable (or feature) names. The first two parts are indispensable while the third part can be omitted (better provided if available). The data set must to be saved in MatLab data file format (*.mat) and its parts have to be prepared as a 3-dimension matrix. The first dimension is the time dimension; the second is the variable dimension and the third is the sample number dimension. Class label is a row or column vector which indicates the category of each sample. Variable name are a cell vector, in which each cell is a string, representing the name of the corresponding variable of the sample matrix.

A.3 User Interface

The user interface of NeuCube system is shown in Figure A.1 below:

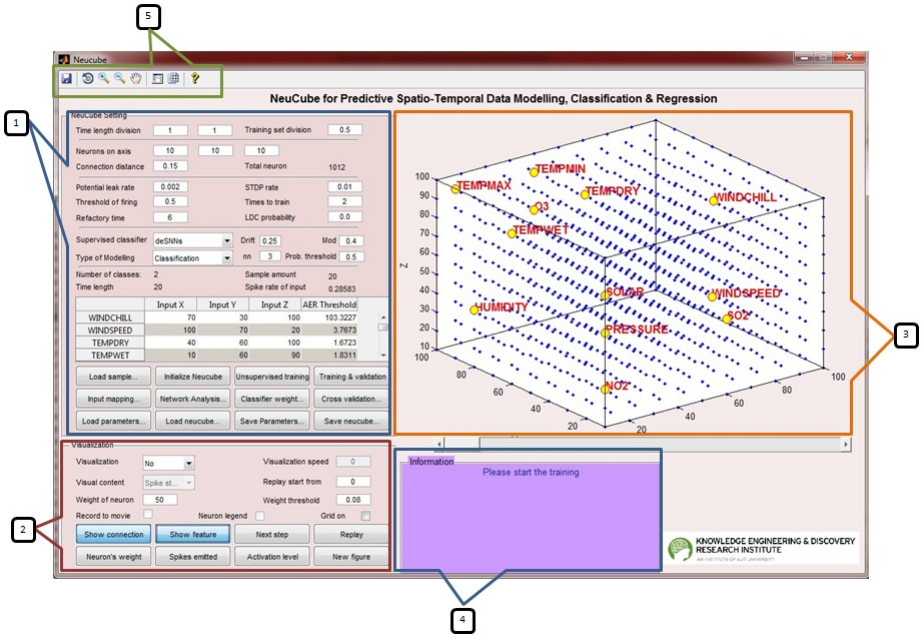

Figure A.1: NeuCube Interface for predictive modelling

1. The algorithm control area, including data loading, algorithm execution, and results analysis & saving.

2. The visualisation control area, including visualisation type, neuron activity showing, connections & weights showing and new figure.

3. The display area, including Neucube illustration and results exhibition.

4. Information area, including some operational tips and explanation of the results.

5. Toolbar area, including some utilities for illustration controls and figure saving.

A.4 Basic Operations

Basic operations contain 4 steps listed below:

1. Load Data

2. Initialize NeuCube

3. Unsupervised Training (cube)

4. Training and Validation (classifier/regression)

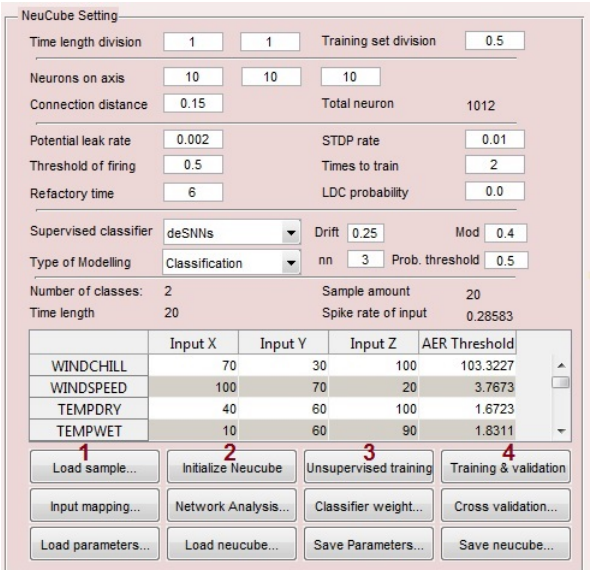

Figure A.2: Area 1 for parameters setting and basic operation

The procedure is indicated by the numbers in the Figure A.2. This procedure is for random sub-sampling validation techniques where the user needs to set the percentage for training set division between the value of 0 & 1. For predictive modelling user can set the time length division where it refers to training and testing time length, the value is between 0 & 1. This means the user can specify a different time length for training and testing that is suitable for their data. Before starting the initialisation procedure, the reservoir requires several parameters setting which are reservoir size, small world connection distance, potential leak rate, threshold of firing, refractory time, STDP rate, times to train the reservoir and long distance connection (LDC). Neuron on axis refer to each axis x, y and z. The value for small world connection, potential leak rate, threshold of firing, STDP rate and LDC is all between 0 & 1. If any of these parameters are changed the procedure needs to be repeated again from the beginning to reset the cube.

After the reservoir is trained, the user can change the parameters for supervised training. For a classification problem, the user can choose from the list of available algorithms such as deSNNs, deSNNs_wknn, deSNNm and SPAN. However for a prediction/regression problem the user can only use deSNNs_wkNN for supervised learning at this point in time. The relevant parameters are drift, modulation (mod), k-value (nn) and probability threshold (P_threshold), depending on which algorithm

is chosen. Then press the Training & Validation button to train the classifier and perform validation.

A.5 Visualisation

The visualisation panel (see Figure A.3) can be used when the user wants to dynamically visualise the neuron activity in the reservoir. This can be done after the initialisation stage, where the user can change the options to 'continuous' or 'step wise' after the Unsupervised training button is clicked. Visualisation content also can be change dynamically between 'spike state', 'connection' and 'weight change' during unsupervised training. Unfortunately this procedure may slow down the training process. Other buttons are also available during the training (except for Next step and New figure).

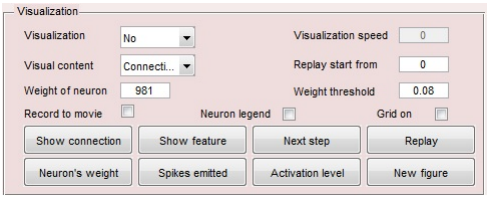

Figure A.3: Area 2 for visualisation parameter setting

A.6 Input Mapping

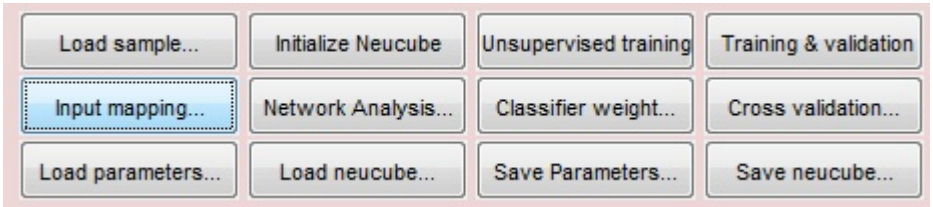

Figure A.4: Input mapping button

The input mapping is normally automatically calculated after user uploads the dataset. However the user can redefine the input mapping by clicking the input mapping button shown in Figure A.4. The input mapping panel shown in Figure A.5 will be opened. Numbers in left-hand graph indicate the corresponding neuron ID in the reservoir and line thickness indicates the similarity of the neurons (in

the sense of inverse Euclidean distance). Edges in the right-hand graph indicate the similarity of the input features (method of similarity measure can be chosen with options below the graph). For similarity measured with spike trains, the AER threshold in the last column of the table can be changed to obtain different encoding spike rates and hence different feature similarity graphs. The table reflects the current input mapping. The user can manually change the mapping either by changing the neuron ID or by changing the neurons coordinates.

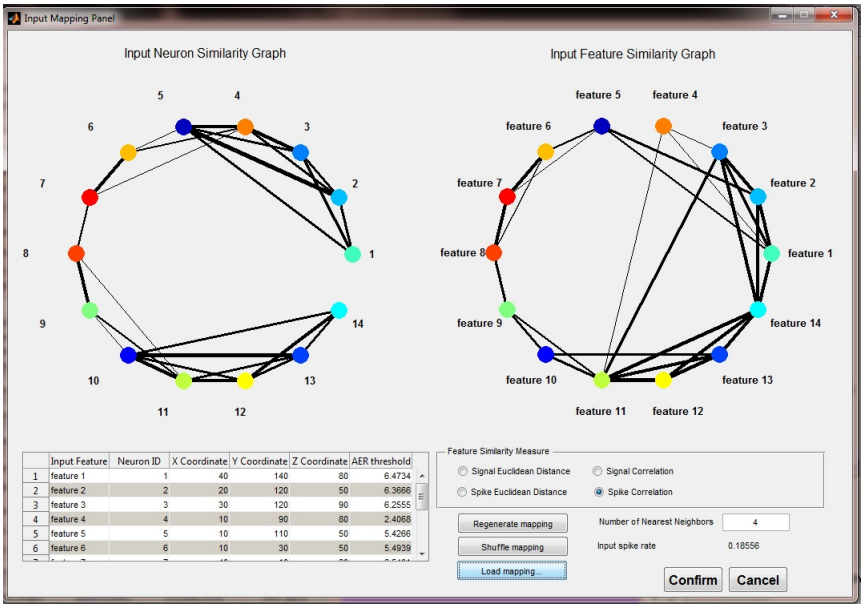

Figure A.5: Input Mapping Panel

The user can generate randomly a new group of mapping neurons or just shuffle the existed mapping correspondence between input neurons and features. Predefined mapping can also be loaded into the system through the 'Load mapping' button. Shown in Figure A.5 is the mapping of EEG neural signal coordinates. The two options provided to load a predefined mapping:

1. Load by txt file: Put the input coordinates first, followed by all neuron coordinates.

2. Load by MatLab data file: The MatLab data file has to include two variables, neuinput and neuron_location, containing input coordinates and all neuron coordinates, respectively.

A.7 Deep Learning

One of the advantages of NeuCube is the capability to go deeper inside the system for deep learning and knowledge discovery. The user can see how networks of neurons are structured and interact with each other, and how connection weight changes influence the modelling result. Two functions can be utilised which are network analysis and classifier weight analysis.

A.7.1 Network Analysis

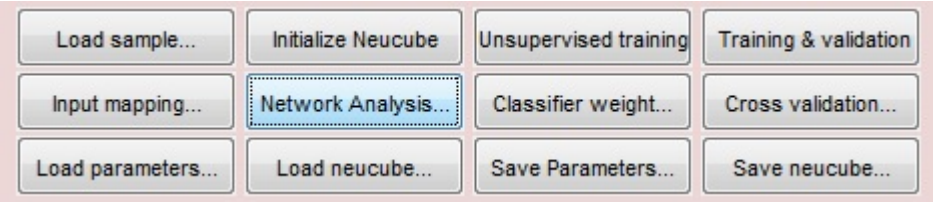

Figure A.6: Network Analysis button

After unsupervised training, the user can observe the network structure in the reservoir by clicking on the network analysis button shown in Figure A.6. The network analysis panel (Figure A.7) will be opened.

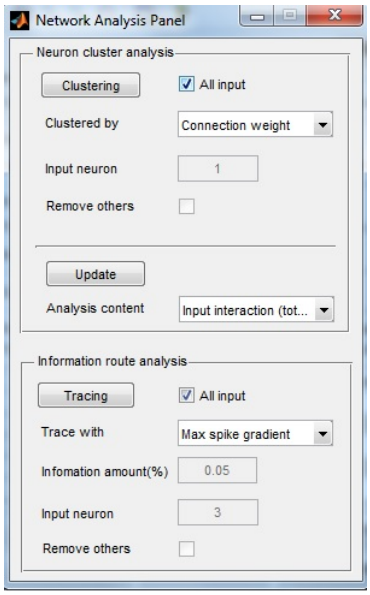

Figure A.7: Network Analysis Panel

The neurons can be clustered by connection weights or by spike communication

between them in the Neuron cluster analysis section shown in Figure A.8 as an example of neuron clustered by connection weights.

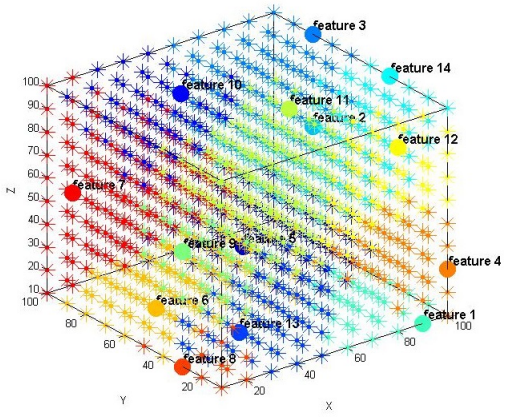

Figure A.8: An example of neuron cluster by connection weight

If the user un-checks the All input option, each input neuron cluster can be observed individually. The user can also choose to show the mean/overall interaction between the input clusters (Figure A.9) or the neurons belonging to each input cluster (Figure A.10).

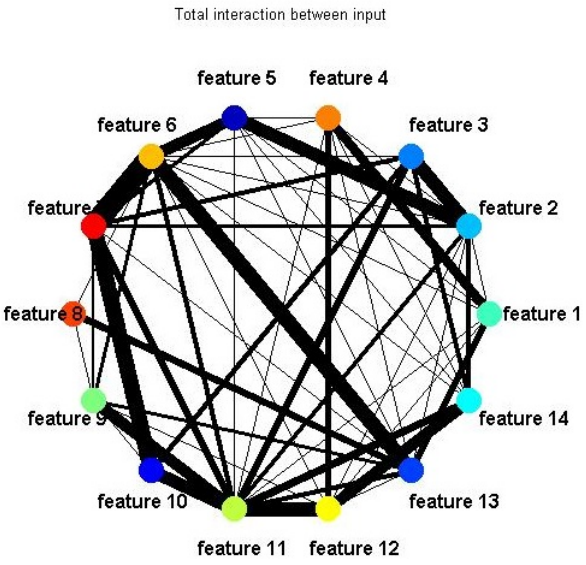

Figure A.9: An example of total input neurons interaction

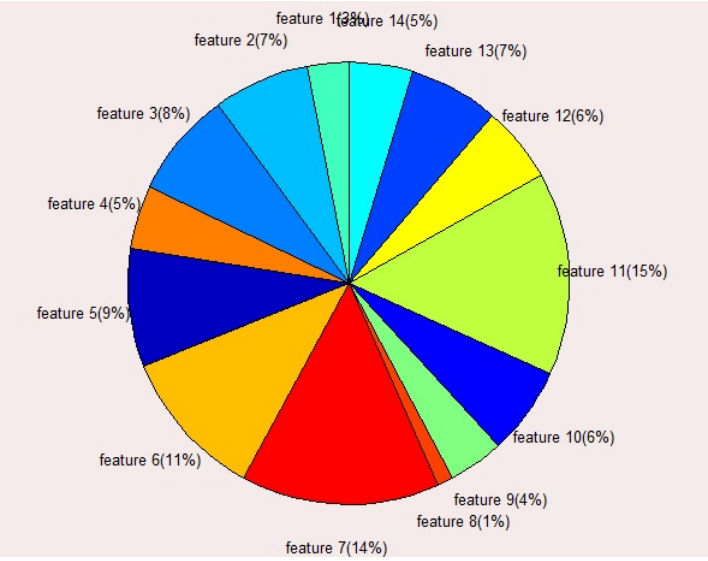

Figure A.10: An example of neurons belonging to each input cluster

In the Information route analysis section, the user can observe the information spreading hierarchy from each input neuron to other neurons (shown in Figure A.11), either for all inputs or for each individual input.

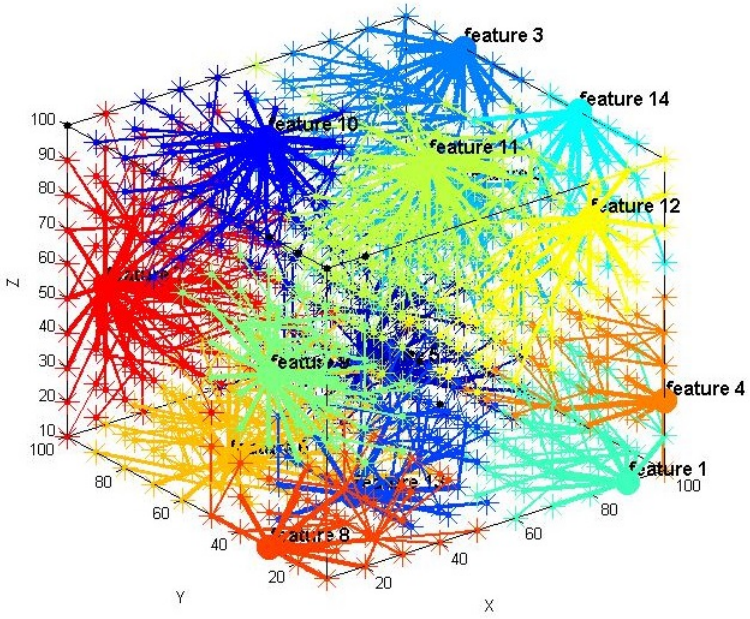

Figure A.11: An example of information spreading hierarchy from each input neuron to other neurons

A.7.2 Classifier Weight Analysis

Figure A.12: Classifier Weight Analysis button

Click the button Classifier Weight shown in Figure A.12 to observe the output classifier connection weight from each neuron to each training or validation sample. The Classifier Weight Analysis panel shown in Figure A.13 lets the user choose which sample to analyse. This function enable user to analyse the network of each sample on a personalised level and comprehend the neurons weight changes for each input feature related to each sample. The user can only activate this function after the training and validation stage.

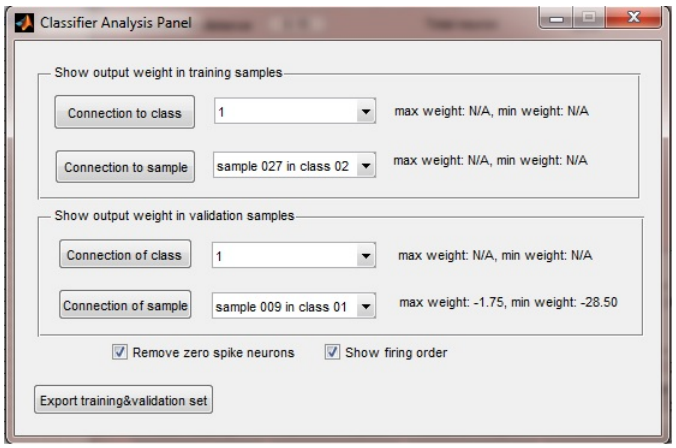

Figure A.13: Classifier Weight Analysis panel

In the Classifier Weight Analysis panel, the user can choose to show the overall connection weights for all the samples in one specific class to each neuron, or show the weights between a specific training or validation sample and all neurons. The brighter colored neurons have stronger output weights and the number ranks means the relative firing order of the neurons. If 'Remove zero spike neurons' andor 'Show firing order' are clicked (only valid for connection of sample) and then the 'Connection to sample' button is clicked, the user can see the corresponding neuron

reservoir structure (refer to Figure A.14). The training and validation sets can also be exported to a file.

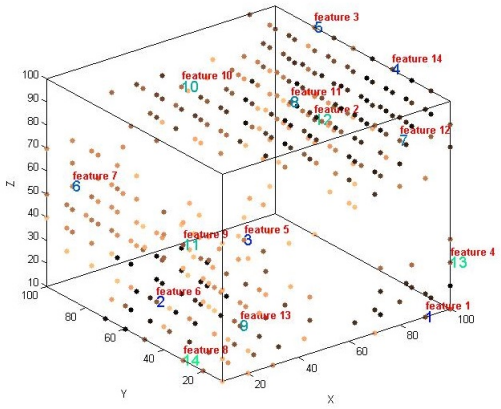

Figure A.14: Classifier Weight Analysis Reservoir

A.8 k-fold Cross Validation

Figure A.15: Cross validation button

A k-fold cross validation can be performed on a data set, by pressing the Cross Validation button shown in Figure A.15 and enter the number of neighbours to use (If k is 1, the system will perform leave-one-out cross validation) in cross validation panel shown in Figure A.16. The system will automatically perform the whole training and validation process. The progress will be displayed in both the information area (Area 4 shown in Figure A.1) and the command window.

If the system obtains a good result and the user wants to repeat it again, the weight_connection.mat file will need to be saved as it is normally overwritten each time. To reload saved connection weights, replace the weight_connection.mat file with the previously saved one in the NeuCube folder, then enter 'Yes' in the third input optimal in Cross validation dialog box.

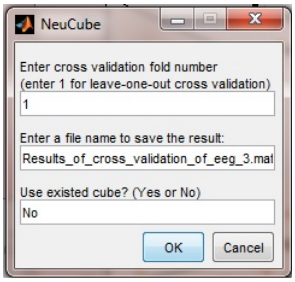

Figure A.16: Cross validation dialog box

A.9 Parameter Optimization

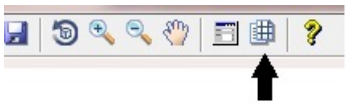

Figure A.17: Parameter optimization button

On the toolbar shown in Figure A.17, clicking the parameter optimization button (indicated by the arrow) will open the parameter optimization panel shown in Figure A.18. In this panel choose which parameters to be optimized, and set the searching interval and searching step number. Then the software will automatically run the chosen algorithm repeatedly using the parameters chosen uniformly from the interval. Progress is displayed in the MatLab command window. The best result will be saved in a text file and the best parameter values will be saved in a MatLab file.

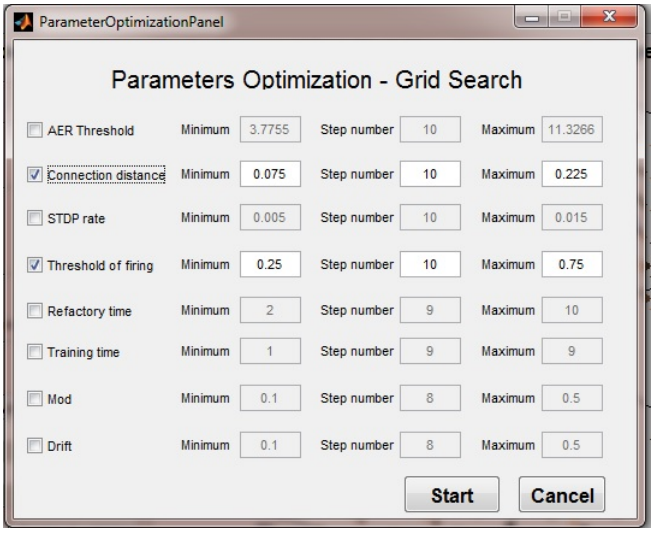

Figure A.18: Parameter optimization panel

A.10 Other Functions

A.10.1 Reuse of Middle Result

Parameter tuning is vital to achieve better accuracy without time and high resource computation memory. Different data sets may have different optimal parameters and system settings. There are two types of middle results that can be reused:

1. Parameter setting: when a good group of parameters is obtained, the user can save them to a file in order to tune them repeatedly next time and in order to avoid forgetting them. Please load the parameters first and then load the data set if wanting to use an existing parameter setting.

2. NeuCube system: when a good result is achieved, the result and the cube can be saved and reused directly without training next time. To reload an existing cube:

   (a) Load the save cube.

   (b) Set the parameter Training Set Size to 0 value.

   (c) Load new data samples and press the button Training & Validation directly.

A.10.2 Training or Validation Only

To train the system using the full data set without creating a validation set by setting the parameter Training Set Size to 1 before loading the data set. For pure validation on the data set can be done by using an existing NeuCube and setting the parameter Training Set Size to 0 before loading data set, and then follow the steps in Section A.4.

Appendix B

Optimised Parameters for Stroke Risk Prediction Study

B.1 Optimised Parameters

The Stroke Risk Prediction study that was discussed in Chapter 6 has four experiments executed for feasibility study of PMeSSNr method and NeuCube M1 System. Listed below are the optimised parameters that have been selected through grid search optimisation method. These parameters can be reused for recall/validation in future experiments with the NeuCube MI System.

Season	Winter		Summer		Spring		Autumn	
Year	2002		2002		2002		2002	
Number of Sample	20		46		26		16	
Age of Patient	50-70		50-70		35-50		20-35	
History of High Blood Pressure	Yes		Yes		Yes		No	
Smoking status	Current		NA		NA		NA	
Prediction data ahead	**1 day**	**6 days**	**1 day**	**6 days**	**1 day**	**6 days**	**1 day**	**6 days**
Time Length Training Percentage	100%	100%	100%	100%	100%	100%	100%	100%
Time Length Testing Percentage	100%	75%	100%	75%	100%	75%	100%	75%
Neuron on Axis (cube size)	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6	6x6x6	10x10x10	10x10x10
SWC	0.3	0.15	0.15	0.15	0.158	0.158	0.208	0.208
Potential Leak Rate	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Threshold of firing	0.5	0.472	0.5	0.5	0.417	0.417	0.306	0.306
Refactory Time	6	6	6	6	6	6	6	6
STDP Rate	0.01	0.005	0.01	0.005	0.006	0.006	0.006	0.006
Time To Train	2	2	2	2	2	2	2	2
LCD	0	0	0	0	0	0	0	0
Classifier	deSNNs	deSNNs	deSNNs_wknn	deSNNs_wknn	deSNNs_wknn	deSNNs_wknn	deSNNs_wknn	deSNNs_wknn
Drift	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Mod	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
k-value(Nearest Neighbour)	-	-	3	3	3	3	3	3
Probability Threshold	-	-	0.5	0.5	0.5	0.5	0.5	0.5

Figure B.1: Optimised parameter for stroke risk prediction study

Bibliography

[Abeles 2004] Moshe Abeles, Gaby Hayon and Daniel Lehmann. *Modeling compositionality by dynamic binding of synfire chains.* Journal of Computational Neuroscience, vol. 17, no. 2, pages 179–201, 2004. xiii, 43, 44

[Abeles 2009] M. Abeles. *Synfire chains.* Scholarpedia, vol. 4, no. 7, page 1441, 2009. revision 91850. 43

[Andersen 2012] Zorana J Andersen, Luise C Kristiansen, Klaus K Andersen, Tom S Olsen, Martin Hvidberg, Steen S Jensen, Matthias Ketzel, Steffen Loft, Mette Sørensen, Anne Tjønneland*et al. Stroke and long-term exposure to outdoor air pollution from nitrogen dioxide a cohort study.* Stroke, vol. 43, no. 2, pages 320–325, 2012. 104

[Anderson 2005] Craig S Anderson, Kristie N Carter, Maree L Hackett, Valery Feigin, P Alan Barber, Joanna B Broad, Ruth Bonita*et al. Trends in stroke incidence in Auckland, New Zealand, during 1981 to 2003.* Stroke, vol. 36, no. 10, pages 2087–2093, 2005. 99

[Anderson 2006] Judy E Anderson, Lise Lotte Hansen, Frank C Mooren, Markus Post, Hubert Hug, Anne Zuse and Marek Los. *Methods and biomarkers for the diagnosis and prognosis of cancer and other diseases: towards personalized medicine.* Drug Resistance Updates, vol. 9, no. 4, pages 198–210, 2006. 2

[Andrienko 2006] Gennady Andrienko, Donato Malerba, Michael May and Maguelonne Teisseire. *Mining spatio-temporal data.* Journal of Intelligent Information Systems, vol. 27, no. 3, pages 187–190, 2006. 1, 148

[Anjum 2007] Ashiq Anjum, Peter Bloodsworth, Andrew Branson, Tamas Hauer, Richard McClatchey, Kamran Munir, Dmitri Rogulin and Jetendr Shamdasani. *The requirements for ontologies in medical data integration: A case study.* In Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International, pages 308–314. IEEE, 2007. 148

[Arel 2010] Itamar Arel, Derek C Rose and Thomas P Karnowski. *Deep Machine Learning-A New Frontier in Artificial Intelligence Research.* IEEE Computational Intelligence Magazine, 2010. 25

[Bale 2002] Jeffery S Bale, Gregory J Masters, Ian D Hodkinson, Caroline Awmack, T Martijn Bezemer, Valerie K Brown, Jennifer Butterfield, Alan Buse, John C Coulson, John Farrar*et al. Herbivory in global climate change research: direct effects of rising temperature on insect herbivores.* Global Change Biology, vol. 8, no. 1, pages 1–16, 2002. 125

[Bartfay 2006] E Bartfay, WJ Mackillop and JL Pater. *Comparing the predictive value of neural network models to logistic regression models on the risk of death for small-cell lung cancer patients.* European Journal of Cancer Care, vol. 15, no. 2, pages 115–124, 2006. 47

[Berginer 1989] Vladimir M Berginer, John Goldsmith, Uri Batz, Hilell Vardi and Yair Shapiro. *Clustering of strokes in association with meteorologic factors in the Negev Desert of Israel: 1981-1983.* Stroke, vol. 20, no. 1, pages 65–69, 1989. 96, 97

[Bezemer 1998] T Martijn Bezemer, T Hefin Jones and Kevin J Knight. *Long-term effects of elevated CO2 and temperature on populations of the peach potato aphid Myzus persicae and its parasitoid Aphidius matricariae.* Oecologia, vol. 116, no. 1-2, pages 128–135, 1998. 123

[Bi 1998] Guo-qiang Bi and Mu-ming Poo. *Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type.* The Journal of Neuroscience, vol. 18, no. 24, pages 10464–10472, 1998. xiii, 41

[Bi 2001] Guo-qiang Bi and Mu-ming Poo. *Synaptic modification by correlated activity: Hebb's postulate revisited.* Annual Review of Neuroscience, vol. 24, no. 1, pages 139–166, 2001. 39

[Bishop 1999] Christopher M Bishop and Wolfgang Maass. Pulsed neural networks. MIT Press Cambridge, MA, 1999. 38

[Blackman 2007] RL Blackman, VF Eastop, HF van Emden, R Harrington*et al.* *Taxonomic issues.* Aphids as Crop Pests, pages 1–29, 2007. 123

[Boahen 2007] K Boahen. *The brain and the computer.* In Device Research Conference, 2007 65th Annual, pages 235–235. IEEE, 2007. 27

[Bohte 2002] Sander M Bohte, Joost N Kok and Han La Poutre. *Error-backpropagation in temporally encoded networks of spiking neurons.* Neurocomputing, vol. 48, no. 1, pages 17–37, 2002. 38, 42, 47, 48

[Boser 1992] Bernhard E Boser, Isabelle M Guyon and Vladimir N Vapnik. *A training algorithm for optimal margin classifiers.* In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pages 144–152. ACM, 1992. 18

[Brader 2007] Joseph M Brader, Walter Senn and Stefano Fusi. *Learning real-world stimuli in a neural network with spike-driven synaptic dynamics.* Neural Computation, vol. 19, no. 11, pages 2881–2912, 2007. xiv, 56

[Brunt 1996] AA Brunt, K Crabtree, MJ Dallwitz, AJ Gibbs, L Watson and EJ Zurcher. *Plant viruses online: descriptions and lists from the VIDE database*, 1996. 124

[Burkitt 2006] Anthony N Burkitt. *A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input.* Biological Cybernetics, vol. 95, no. 1, pages 1–19, 2006. 33, 34

[Cawley 2010] Gavin C Cawley and Nicola LC Talbot. *On over-fitting in model selection and subsequent selection bias in performance evaluation.* The Journal of Machine Learning Research, vol. 11, pages 2079–2107, 2010. 18

[Çevik 2014] Yunsur Çevik, Nurettin ´Ozg´ur Doğan, Murat Daş, Asliddin Ahmedali, Seval Kul and Hasan Bayram. *The association between weather conditions and stroke admissions in Turkey.* International Journal of Biometeorology, pages 1–7, 2014. 97

[Chang 2011] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, page 27, 2011. 16

[Chen 2013] Renjie Chen, Cuicui Wang, Xia Meng, Honglei Chen, Thuan Quoc Thach, Chit-Ming Wong and Haidong Kan. *Both low and high temperature may increase the risk of stroke mortality*. Neurology, vol. 81, no. 12, pages 1064–1070, 2013. 96

[Chen 2014] Szu-Ying Chen, Yu-Lun Lin, Wei-Tien Chang, Chung-Te Lee and Chang-Chuan Chan. *Increasing emergency room visits for stroke by elevated levels of fine particulate constituents*. Science of The Total Environment, vol. 473, pages 446–450, 2014. 96, 109, 111

[Cho 2014] Minsu Cho, Jian Sun, Olivier Duchenne and Jean Ponce. *Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers*. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2091–2098. IEEE, 2014. 88

[Corcho 2005] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez and Angel López-Cima. *Building legal ontologies with METHONTOLOGY and WebODE*. In Law and the Semantic Web, pages 142–157. Springer, 2005. 148

[Cortes 1995] Corinna Cortes and Vladimir Vapnik. *Support-vector networks*. Machine Learning, vol. 20, no. 3, pages 273–297, 1995. 14, 15

[Culurciello 2001] Eugenio Culurciello, Ralph Etienne-Cummings and Kwabena Boahen. *Arbitrated address-event representation digital image sensor*. Electronics Letters, vol. 37, no. 24, pages 1443–1445, 2001. 65

[Das 2003] Ananya Das, Tamir Ben-Menachem, Gregory S Cooper, Amitabh Chak, Michael V Sivak Jr, Judith A Gonet and Richard CK Wong. *Prediction of outcome in acute lower-gastrointestinal haemorrhage based on an artificial neural network: internal and external validation of a predictive model*. The Lancet, vol. 362, no. 9392, pages 1261–1266, 2003. 2

[Dawson 2008] J Dawson, C Weir, F Wright, C Bryden, S Aslanyan, K Lees, W Bird and M Walters. *Associations between meteorological variables and acute stroke hospital admissions in the west of Scotland.* Acta Neurologica Scandinavica, vol. 117, no. 2, pages 85–89, 2008. 96

[Delbruck 2007] T Delbruck and Patrick Lichtsteiner. *Fast sensory motor control based on event-based hybrid neuromorphic-procedural system.* In Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, pages 845–848. IEEE, 2007. 65

[Dhoble 2012] Kshitij Dhoble, Nuttapod Nuntalid, Giacomo Indiveri and Nikola Kasabov. *Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning.* In Neural Networks (IJCNN), The 2012 International Joint Conference on, pages 1–7. IEEE, 2012. 4, 26, 56, 59, 68

[Duchenne 2011] Olivier Duchenne, Armand Joulin and Jean Ponce. *A graph-matching kernel for object categorization.* In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 1792–1799. IEEE, 2011. 88

[Dudani 1976] Sahibsingh A Dudani. *The distance-weighted k-nearest-neighbor rule.* Systems, Man and Cybernetics, IEEE Transactions on, no. 4, pages 325–327, 1976. 3

[Elman 1991] Jeffrey L Elman. *Distributed representations, simple recurrent networks, and grammatical structure.* Machine Learning, vol. 7, no. 2-3, pages 195–225, 1991. 27

[Feigin 1997] Valery L Feigin and David O Wiebers. *Environmental factors and stroke: a selective review.* Journal of Stroke and Cerebrovascular Diseases, vol. 6, no. 3, pages 108–113, 1997. 3

[Feigin 2000] Valery L Feigin, Sergey V Shishkin, Georgii M Tzirkin, Tatyana E Vinogradova, Alexey V Tarasov, Sergey P Vinogradov and Yury P Nikitin. *A population-based study of transient ischemic attack incidence in Novosibirsk, Russia, 1987–1988 and 1996–1997.* Stroke, vol. 31, no. 1, pages 9–13, 2000. 96, 102

[Feigin 2011] Valery Feigin. When lightning strikes: An illustrated guide to stroke prevention and re covery. HarperCollins Australia, 2011. 94

[Feigin 2014] Valery L Feigin, Priya G Parmar, Suzanne Barker-Collo, Derrick A Bennett, Craig S Anderson, Amanda G Thrift, Birgitta Stegmayr, Peter M Rothwell, Maurice Giroud, Yannick Bejot*et al.* *Geomagnetic Storms Can Trigger Stroke Evidence From 6 Large Population-Based Studies in Europe and Australasia.* Stroke, vol. 45, no. 6, pages 1639–1645, 2014. 96, 97, 113

[Finlay 2011] KJ Finlay and JE Luck. *Response of the bird cherry-oat aphid (Rhopalosiphum padi) to climate change in relation to its pest status, vectoring potential and function in a crop–vector–virus pathosystem.* Agriculture, Ecosystems & Environment, vol. 144, no. 1, pages 405–421, 2011. xvii, 121, 124, 125

[Fix 1951] Evelyn Fix and Joseph L Hodges Jr. *Discriminatory analysis-nonparametric discrimination: consistency properties.* Rapport technique, DTIC Document, 1951. 3, 20

[Floreano 2006] Dario Floreano, Yann Epars, Jean-Christophe Zufferey and Claudio Mattiussi. *Evolution of spiking neural circuits in autonomous mobile robots.* International Journal of Intelligent Systems, vol. 21, no. 9, pages 1005–1024, 2006. 47

[Fuhrer 2003] J́urg Fuhrer. *Agroecosystem responses to combinations of elevated CO 2, ozone, and global climate change.* Agriculture, Ecosystems & Environment, vol. 97, no. 1, pages 1–20, 2003. 123

[Fung 2001] Glenn Fung and Olvi L. Mangasarian. *Proximal Support Vector Machine Classifiers.* In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, pages 77–86, New York, NY, USA, 2001. ACM. 16

[Fuortes 1962] MGF Fuortes and Francoise Mantegazzini. *Interpretation of the repetitive firing of nerve cells.* The Journal of general physiology, vol. 45, no. 6, pages 1163–1179, 1962. 35, 36

[Fyfe 2008] Colin Fyfe, Wesam Barbakh, Wei Chuan Ooi and Hanseok Ko. *Topological mappings of video and audio data.* International Journal of Neural Systems, vol. 18, no. 06, pages 481–489, 2008. 47

[Gant 2001] Vanya Gant, Susan Rodway and Jeremy Wyatt. *Artificial neural networks: practical considerations for clinical applications*. Clinical Applications of Artificial Neural Networks, pages 329–356, 2001. 28

[Gerstner 1995] Wulfram Gerstner. *Time structure of the activity in neural network models.* Physical Review E, vol. 51, no. 1, page 738, 1995. 4, 28

[Gerstner 1996] Wulfram Gerstner, Richard Kempter, J Leo van Hemmen and Hermann Wagner. *A neuronal learning rule for sub-millisecond temporal coding.* Nature, vol. 383, no. LCN-ARTICLE-1996-002, pages 76–78, 1996. 40

[Gerstner 2002] Wulfram Gerstner and Werner M Kistler. Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press, 2002. xiii, 4, 26, 28, 35, 36, 38

[Ghahramani 2001] Zoubin Ghahramani. *An introduction to hidden Markov models and Bayesian networks.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 01, pages 9–42, 2001. 26

[Gill 2013] Randeep S Gill, Hali L Hambridge, Eric B Schneider, Thomas Hanff, Rafael J Tamargo and Paul Nyquist. *Falling temperature and colder weather are associated with an increased risk of aneurysmal subarachnoid hemorrhage*. World Neurosurgery, vol. 79, no. 1, pages 136–142, 2013. 96, 97, 102

[Gomes 2014a] Joana Gomes, Albertino Damasceno, Carla Carrilho, Vitória Lobo, Hélder Lopes, Tavares Madede, Pius Pravinrai, Carla Silva-Matos, Domingos Diogo, Ana Azevedo*et al. The effect of season and temperature variation on hospital admissions for incident stroke events in Maputo, Mozambique*. Journal of Stroke and Cerebrovascular Diseases, vol. 23, no. 2, pages 271–277, 2014. 96, 102

[Gomes 2014b] Joana Gomes, Albertino Damasceno, Carla Carrilho, Vitória Lobo, Hélder Lopes, Tavares Madede, Pius Pravinrai, Carla Silva-Matos, Domingos

Diogo, Ana Azevedo*et al.* *On the Causal Paths Underlying the Relation between Atmospheric Temperature and Acute Stroke.* Journal of Stroke and Cerebrovascular Diseases, vol. 23, no. 1, pages 195–197, 2014. xvi, 95, 97, 98

[Gommans 2003] John Gommans, Alan Barber, Harry McNaughton, Carl Hanger, Patricia Bennett, David Spriggs and Jonathan Baskett. *Stroke rehabilitation services in New Zealand.* The New Zealand Medical Journal, vol. 116, no. 1174, pages U435–U435, 2003. 93

[Gottgtroy 2006] Paulo Gottgtroy, Nikola Kasabov and Stephen MacDonell. *Evolving ontologies for intelligent decision support.* Capturing Intelligence, vol. 1, pages 415–439, 2006. xviii, 149

[Hackett 2000] Maree L Hackett, John R Duncan, Craig S Anderson, Joanna B Broad and Ruth Bonita. *Health-related quality of life among long-term survivors of stroke results from the Auckland stroke study, 1991–1992.* Stroke, vol. 31, no. 2, pages 440–447, 2000. 93, 94

[Hamed 2011] Haza Nuzly Abdull Hamed, Nikola Kasabov, Siti Mariyam Shamsuddin, Harya Widiputra and Kshitij Dhoble. *An extended evolving spiking neural network model for spatio-temporal pattern classification.* In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 2653–2656. IEEE, 2011. xiv, 4, 26, 47, 51, 52, 59

[Harrington 2007] Richard Harrington, Suzanne J Clark, Sue J Welham, Paul J Verrier, Colin H Denholm, Maurice Hulle, Damien Maurice, Mark D Rounsevell and Nadege Cocu. *Environmental change and the phenology of European aphids.* Global Change Biology, vol. 13, no. 8, pages 1550–1564, 2007. 123

[Harris 2015] Terry Harris. *Credit scoring using the clustered support vector machine.* Expert Systems with Applications, vol. 42, no. 2, pages 741–750, 2015. 16

[Hartmann 2010] Sven Hartmann, Henning K´ohler and Jing Wang. *Ontology consolidation in bioinformatics.* In Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling-Volume 110, pages 15–22. Australian Computer Society, Inc., 2010. 148

[Hazell 2010] Steaphan P. Hazell, Bolette Palle Neve, Constantinos Groutides, Angela E. Douglas, Tim M. Blackburn and Jeffrey S. Bale. *Hyperthermic aphids: Insights into behaviour and mortality.* Journal of Insect Physiology, vol. 56, no. 2, pages 123 – 131, 2010. 125

[Henrotin 2007] Jean-Bernard Henrotin, Jean-Pierre Besancenot, Y Bejot and Maurice Giroud. *Short-term effects of ozone air pollution on ischaemic stroke occurrence: a case-crossover analysis from a 10-year population-based study in Dijon, France.* Occupational and Environmental Medicine, vol. 64, no. 7, pages 439–445, 2007. 105

[Hodgkin 1952a] AL Hodgkin and AF Huxley. *The components of membrane conductance in the giant axon of Loligo.* The Journal of Physiology, vol. 116, no. 4, pages 473–496, 1952. 31

[Hodgkin 1952b] Alan L Hodgkin and Andrew F Huxley. *A quantitative description of membrane current and its application to conduction and excitation in nerve.* The Journal of Physiology, vol. 117, no. 4, page 500, 1952. xiii, 28, 31, 32

[Hodgkin 1952c] Allan L Hodgkin and Andrew F Huxley. *Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo.* The Journal of Physiology, vol. 116, no. 4, page 449, 1952. 27, 31

[Hodgkin 1952d] Allan L Hodgkin and Andrew F Huxley. *The dual effect of membrane potential on sodium conductance in the giant axon of Loligo.* The Journal of Physiology, vol. 116, no. 4, pages 497–506, 1952. 31

[Hodgkin 1952e] Ao L Hodgkin, AF Huxley and B Katz. *Measurement of current-voltage relations in the membrane of the giant axon of Loligo.* The Journal of Physiology, vol. 116, no. 4, page 424, 1952. 31

[Hong 2002] Yun-Chul Hong, Jong-Tae Lee, Ho Kim and Ho-Jang Kwon. *Air pollution a new risk factor in ischemic stroke mortality.* Stroke, vol. 33, no. 9, pages 2165–2169, 2002. 104

[Hong 2003] Yun-Chul Hong, Joung-Ho Rha, Jong-Tae Lee, Eun-Hee Ha, Ho-Jang Kwon and HO Kim. *Ischemic stroke associated with decrease in temperature.* Epidemiology, vol. 14, no. 4, pages 473–478, 2003. 96, 97, 102

[Hopfield 1982] John J Hopfield. *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the National Academy of Sciences, vol. 79, no. 8, pages 2554–2558, 1982. 27

[Hornik 1989] Kurt Hornik, Maxwell Stinchcombe and Halbert White. *Multilayer feedforward networks are universal approximators.* Neural Networks, vol. 2, no. 5, pages 359–366, 1989. 3, 14

[Horváth 2003] Gábor Horváth. *CMAC: Reconsidering an old neural network.* Proc. Intell. Control Syst. Signal Process, pages 173–178, 2003. 18

[Hough 1999] Michael Hough, Hugo De Garis, Michael Korkin, Felix Gers and Norberto Eiji Nawa. *SPIKER: Analog waveform to digital spiketrain conversion in ATR's artificial brain (cam-brain) project.* In International Conference on Robotics and Artificial Life. Citeseer, 1999. 66

[Howling 1993] G. G. Howling, R. Harrington, S. J. Clark and J. S. Bale. *The use of multiple regression via principal components in forecasting early season aphid (Homoptera: Aphididae) flight.* Bulletin of Entomological Research, vol. 83, pages 377–381, 9 1993. 126

[Humble 2012] James Humble, Susan Denham and Thomas Wennekers. *Spatio-temporal pattern recognizers using spiking neurons and spike-timing-dependent plasticity.* Frontiers in Computational Neuroscience, vol. 6, 2012. 43

[Investigators 1988] WHO MONICA Project Principal Investigators*et al.* *The World Health Organization MONICA Project (monitoring trends and determinants in cardiovascular disease): a major international collaboration.* Journal of Clinical Epidemiology, vol. 41, no. 2, pages 105–114, 1988. 94

[Izhikevich 2004] Eugene M Izhikevich. *Which model to use for cortical spiking neurons?* IEEE Transactions on Neural Networks, vol. 15, no. 5, pages 1063–1070, 2004. 4, 28

[Izhikevich 2006] Eugene M Izhikevich. *Polychronization: computation with spikes.* Neural Computation, vol. 18, no. 2, pages 245–282, 2006. 28, 44, 59, 80

[Izhikevich 2008] Eugene M Izhikevich and Gerald M Edelman. *Large-scale model of mammalian thalamocortical systems.* Proceedings of the National Academy of Sciences, vol. 105, no. 9, pages 3593–3598, 2008. 28

[Jamieson 2012] Mary A Jamieson, Amy M Trowbridge, Kenneth F Raffa and Richard L Lindroth. *Consequences of climate warming and altered precipitation patterns for plant-insect and multitrophic interactions.* Plant Physiology, vol. 160, no. 4, pages 1719–1727, 2012. 125

[Jimenez-Conde 2008] J Jimenez-Conde, A Ois, M Gomis, A Rodriguez-Campello, E Cuadrado-Godia, I Subirana and J Roquer. *Weather as a trigger of stroke.* Cerebrovascular Diseases, vol. 26, no. 4, page 348, 2008. 96, 102

[Juzeniene 2011] Asta Juzeniene, Pål Brekke, Arne Dahlback, Stefan Andersson-Engels, J'org Reichrath, Kristin Moan, Michael F Holick, William B Grant and Johan Moan. *Solar radiation and human health.* Reports on Progress in Physics, vol. 74, no. 6, page 066701, 2011. 103

[Kasabov 1998a] Nikola Kasabov. *Evolving nuzzy neural networks-Algorithms, applications and biological motivation.* Methodologies for the Conception, Design, and Applications of Soft Computing, pages 271–274, 1998. 47

[Kasabov 1998b] Nikola K Kasabov. *The ECOS Framework and the ECO Learning Method for Evolving Connectionist Systems.* JACIII, vol. 2, no. 6, pages 195–202, 1998. 19

[Kasabov 2002] Nikola Kasabov. *Evolving connectionist systems for adaptive learning and knowledge discovery: methods, tools, applications.* In Intelligent Systems, 2002. Proceedings. 2002 First International IEEE Symposium, volume 1, pages 24–28. IEEE, 2002. 18, 19

[Kasabov 2007a] Nikola Kasabov. Evolving connectionist systems. Springer, 2007. xv, 3, 24, 48, 68, 69

[Kasabov 2007b] Nikola Kasabov. *Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach.* Pattern Recognition Letters, vol. 28, no. 6, pages 673 – 685, 2007. 3, 22

[Kasabov 2008] Nikola Kasabov, Qun Song, Lubica Benuskova, Paulo Gottgtroy, Vishal Jain, Anju Verma, Ilkka Havukkala, Elaine Rush, Russel Pears, Alex Tjahjana*et al. Integrating local and personalised modelling with global ontology knowledge bases for biomedical and bioinformatics decision support.* In Computational Intelligence in Biomedicine and Bioinformatics, pages 93–116. Springer, 2008. xvii, 149, 151

[Kasabov 2009a] Nikola Kasabov. *Integrative connectionist learning systems inspired by nature: current models, future trends and challenges.* Natural Computing, vol. 8, no. 2, pages 199–218, 2009. 47, 48

[Kasabov 2009b] Nikola Kasabov. *Soft Computing Methods for Global, Local and Personalized Modeling and Applications in Bioinformatics.* In Soft Computing Based Modeling in Intelligent Systems, pages 1–18. Springer, 2009. 13

[Kasabov 2010a] Nikola Kasabov. *To spike or not to spike: A probabilistic spiking neuron model.* Neural Networks, vol. 23, no. 1, pages 16–19, 2010. xiii, 4, 26, 27, 36, 37, 59

[Kasabov 2010b] Nikola Kasabov and Yingjie Hu. *Integrated optimisation method for personalised modelling and case studies for medical decision support.* International Journal of Functional Informatics and Personalised Medicine, vol. 3, no. 3, pages 236–256, 2010. xiii, 2, 4, 22, 23, 59

[Kasabov 2011] Nikola K Kasabov, Reinhard Schliebs and Hiroshi Kojima. *Probabilistic Computational Neurogenetic Modeling: From Cognitive Systems to Alzheimer's Disease.* Autonomous Mental Development, IEEE Transactions on, vol. 3, no. 4, pages 300–311, 2011. 47

[Kasabov 2012a] Nikola Kasabov. *Evolving spiking neural networks and neurogenetic systems for spatio-and spectro-temporal data modelling and pattern recognition.* In Advances in Computational Intelligence, pages 234–260. Springer, 2012. xiii, 33, 34, 36, 39, 40, 41

[Kasabov 2012b] Nikola Kasabov. *NeuCube evospike architecture for spatio-temporal modelling and pattern recognition of brain signals.* In Artificial Neural Networks in Pattern Recognition, pages 225–243. Springer, 2012. xiv, 57, 59

[Kasabov 2013] Nikola Kasabov, Kshitij Dhoble, Nuttapod Nuntalid and Giacomo Indiveri. *Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition.* Neural Networks, vol. 41, pages 188–201, 2013. 4, 57, 132

[Kasabov 2014a] Nikola Kasabov, Valery Feigin, Zeng-Guang Hou, Yixiong Chen, Linda Liang, Rita Krishnamurthi, Muhaini Othman and Priya Parmar. *Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke.* Neurocomputing, vol. 134, pages 269–279, 2014. xv, 62, 63, 66, 99

[Kasabov 2014b] Nikola K Kasabov. *NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data.* Neural Networks, vol. 52, pages 62–76, 2014. 65, 67, 76, 87

[Kempter 1999] Richard Kempter, Wulfram Gerstner and J Leo Van Hemmen. *Hebbian learning and spiking neurons.* Physical Review E, vol. 59, no. 4, page 4498, 1999. 40

[Khosla 2010] Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu and Honglak Lee. *An integrated machine learning approach to stroke prediction.* In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 183–192. ACM, 2010. 2

[Kiyan 2011] T'uba Kiyan and T'ulay Yildirim. *Breast cancer diagnosis using statistical neural networks.* IU-Journal of Electrical & Electronics Engineering, vol. 4, no. 2, pages 1149–1153, 2011. 47

[Klueken 2009] A. M. Klueken, B. Hau, B. Ulber and H.M. Poehling. *Forecasting migration of cereal aphids (Hemiptera: Aphididae) in autumn and spring.* Journal of Applied Entomology, vol. 133, no. 5, pages 328–344, 2009. 126

[Knoblauch 2005] Andreas Knoblauch. *Neural associative memory for brain modeling and information retrieval.* Information Processing Letters, vol. 95, no. 6, pages 537–544, 2005. 47

[Krogh 1994] Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander and David Haussler. *Hidden Markov models in computational biology: Applications to protein modeling.* Journal of Molecular Biology, vol. 235, no. 5, pages 1501–1531, 1994. 26

[Kyobutungi 2005] C Kyobutungi, A Grau, G Stieglbauer and H Becher. *Absolute temperature, temperature changes and stroke risk: a case-crossover study.* European Journal of Epidemiology, vol. 20, no. 8, pages 693–698, 2005. 96, 108

[Lange 2007] Matthew C Lange, Danielle G Lemay and J Bruce German. *A multi-ontology framework to guide agriculture and food towards diet and health.* Journal of the Science of Food and Agriculture, vol. 87, no. 8, pages 1427–1434, 2007. 2

[Lankin 2001] G Lankin, SP Worner, S Samarasinghe and DAJ Teulon. *Can artificial Neural Network Systems be used for forecasting aphid flight patterns?* In Proceedings of The New Zealand Plant Protection Conference, pages 188–192. New Zealand Plant Protection Society; 1998, 2001. 3, 124, 125, 127

[Lawrence 1997] Steve Lawrence, C Lee Giles and Ah Chung Tsoi. *Lessons in neural network training: Overfitting may be harder than expected.* In AAAI/IAAI, pages 540–545, 1997. 86

[Lee 2001] Yuh-Jye Lee and Olvi L Mangasarian. *SSVM: A smooth support vector machine for classification*. Computational Optimization and Applications, vol. 20, no. 1, pages 5–22, 2001. 16

[Legenstein 2005] Robert Legenstein, Christian Naeger and Wolfgang Maass. *What Can a Neuron Learn with Spike-Timing-Dependent Plasticity?* Neural Computation, vol. 17, pages 2337–2382, 2005. 26

[Levey 1999] Andrew S Levey, Juan P Bosch, Julia Breyer Lewis, Tom Greene, Nancy Rogers and David Roth. *A more accurate method to estimate glomerular filtration rate from serum creatinine: a new prediction equation*. Annals of Internal Medicine, vol. 130, no. 6, pages 461–470, 1999. 2, 13

[Li 2002] Bonan Li and Guoray Cai. *A general object-oriented spatial temporal data model*. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, vol. 34, no. 4, pages 100–105, 2002. 150

[Liang 2011] Wen Liang, Yingjie Hu, Nikola Kasabov and Valery Feigin. *Exploring associations between changes in ambient temperature and stroke occurrence: comparative analysis using global and personalised modelling approaches*. In Neural Information Processing, pages 129–137. Springer, 2011. 99

[Liang 2014] Wen Liang, Rita Krishnamurthi, Nikola Kasabov and Valery Feigin. *Information Methods for Predicting Risk and Outcome of Stroke*. In Springer Handbook of Bio-/Neuroinformatics, pages 993–1001. Springer Berlin Heidelberg, 2014. 47

[Lim 2013] Youn-Hee Lim, Ho Kim and Yun-Chul Hong. *Variation in mortality of ischemic and hemorrhagic strokes in relation to high temperature*. International Journal of Biometeorology, vol. 57, no. 1, pages 145–153, 2013. 96, 97

[Lladós 2001] Josep Lladós, Enric Martí and Juan J. Villanueva. *Symbol recognition by error-tolerant subgraph matching between region adjacency graphs*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 10, pages 1137–1143, 2001. 88

[Loiselle 2005] Stéphane Loiselle, Jean Rouat, Daniel Pressnitzer and Simon Thorpe. *Exploration of rank order coding with spiking neural networks for speech recognition.* In Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, volume 4, pages 2076–2080. IEEE, 2005. 38

[Low 2006] Ronald B Low, Leonard Bielory, Adnan I Qureshi, Van Dunn, David FE Stuhlmiller and David A Dickey. *The relation of stroke admissions to recent weather, airborne allergens, air pollution, seasons, upper respiratory infections, and asthma incidence, September 11, 2001, and day of the week.* Stroke, vol. 37, no. 4, pages 951–957, 2006. 3, 96, 97, 111

[Lumley 2002] Thomas Lumley, Richard A Kronmal, Mary Cushman, Teri A Manolio and Steven Goldstein. *A stroke prediction score in the elderly: validation and Web-based application.* Journal of Clinical Epidemiology, vol. 55, no. 2, pages 129–136, 2002. 2

[Maass 2001] Wolfgang Maass and Christopher M Bishop. Pulsed neural networks. MIT press, 2001. 27, 28

[Maass 2002] Wolfgang Maass, Thomas Natschl´ager and Henry Markram. *Real-time computing without stable states: A new framework for neural computation based on perturbations.* Neural Computation, vol. 14, no. 11, pages 2531–2560, 2002. 4, 45, 46, 53, 59

[Maass 2010] Wolfgang Maass. *Liquid state machines: motivation, theory, and applications.* In Computability in Context: Computation and Logic in the Real World, pages 275–296, 2010. xiv, 46, 53

[Maassa 2002] Wolfgang Maassa and Henry Markramb. *Synapses as dynamic memory buffers.* Neural Networks, vol. 15, pages 155–161, 2002. 26

[Markram 1997] Henry Markram, Joachim Lübke, Michael Frotscher and Bert Sakmann. *Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs.* Science, vol. 275, no. 5297, pages 213–215, 1997. 39, 67

[McArthur 2010] Kate McArthur, Jesse Dawson and Matthew Walters. *What is it with the weather and stroke?* 2010. 93, 96

[McCulloch 1943] Warren S McCulloch and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity.* The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pages 115–133, 1943. 27

[Mennis 2000] Jeremy L Mennis, Donna J Peuquet and Liujian Qian. *A conceptual framework for incorporating cognitive principles into geographical database representation.* International Journal of Geographical Information Science, vol. 14, no. 6, pages 501–520, 2000. 150

[Mohammadi 2015] Kasra Mohammadi, Shahaboddin Shamshirband, Chong Wen Tong, Muhammad Arif, Dalibor Petković and Sudheer Ch. *A new hybrid support vector machine wavelet transform approach for estimation of horizontal global solar radiation.* Energy Conversion and Management, vol. 9, no. 0, pages 162 – 171, 2015. 16

[Mohemmed 2011] Ammar Mohemmed, Stefan Schliebs and Nikola Kasabov. *SPAN: a neuron for precise-time spike pattern association.* In Neural Information Processing, pages 718–725. Springer, 2011. 4, 26, 59, 68

[Nault 1997] LR Nault. *Arthropod transmission of plant viruses: a new synthesis.* Annals of the Entomological Society of America, vol. 90, no. 5, pages 521–541, 1997. 124

[Nelson 2004] ME Nelson. *Databasing the Brain: From Data to Knowledge*, 2004. 31

[Newman 2003] J. A. Newman, D. J. Gibson, A. J. Parsons and J. H. M. Thornley. *How Predictable Are Aphid Population Responses to Elevated CO2?* Journal of Animal Ecology, vol. 72, no. 4, pages pp. 556–566, 2003. 125

[Ng 2004] James CK Ng and Keith L Perry. *Transmission of plant viruses by aphid vectors.* Molecular Plant Pathology, vol. 5, no. 5, pages 505–511, 2004. 124

[Norval 2001] Mary Norval. *Effects of solar radiation on the human immune system.* Journal of Photochemistry and Photobiology B: Biology, vol. 63, no. 1, pages 28–40, 2001. 103

[Nuntalid 2011] Nuttapod Nuntalid, Kshitij Dhoble and Nikola Kasabov. *EEG classification with BSA spike encoding algorithm and evolving probabilistic spiking neural network.* In Neural Information Processing, pages 451–460. Springer, 2011. xv, 66, 67

[O'Donnell 2010] Martin J O'Donnell, Denis Xavier, Lisheng Liu, Hongye Zhang, Siu Lim Chin, Purnima Rao-Melacini, Sumathy Rangarajan, Shofiqul Islam, Prem Pais, Matthew J McQueen*et al. Risk factors for ischaemic and intracerebral haemorrhagic stroke in 22 countries (the INTERSTROKE study): a case-control study.* The Lancet, vol. 376, no. 9735, pages 112–123, 2010. 95

[Organization 2005] World Health Organization*et al. Preventing chronic diseases: a vital investment: WHO global report.* 2005. 2, 93

[Othman 2014] Muhaini Othman, Nikola Kasabov, Enmei Tu, Valery Feigin, Rita Krishnamurthi, Zhengguang Hou, Yixiong Chen and Jin Hu. *Improved predictive personalized modelling with the use of Spiking Neural Network system and a case study on stroke occurrences data.* In Neural Networks (IJCNN), 2014 International Joint Conference on, pages 3197–3204. IEEE, 2014. xvi, 76, 99, 101, 102, 103, 104

[Pfurtscheller 2006] Gert Pfurtscheller, Robert Leeb, Claudia Keinrath, Doron Friedman, Christa Neuper, Christoph Guger and Mel Slater. *Walking from thought.* Brain Research, vol. 1071, no. 1, pages 145–152, 2006. 47

[Pinto 1999] H Sofia Pinto, Asunción Gómez-Pérez and João P Martins. *Some issues on ontology integration.* IJCAI and the Scandinavian AI Societies. CEUR Workshop Proceedings, 1999. 149

[Platel 2009] Micha´el Defoin Platel, Stefan Schliebs and Nikola Kasabov. *Quantum-inspired evolutionary algorithm: a multimodel EDA*. Evolutionary Computation, IEEE Transactions on, vol. 13, no. 6, pages 1218–1232, 2009. 146

[Ponulak 2005] Filip Ponulak. *ReSuMe-new supervised learning method for spiking neural networks*. Institute of Control and Information Engineering, Poznan University of Technology, 2005. 42

[Rabiner 1989] Lawrence Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, vol. 77, no. 2, pages 257–286, 1989. 26, 63

[Ritter 2015] Leslie Ritter and Bruce Coull. *Lowering the Risks of Stroke in Women (and Men)*. 2015. xvi, 95

[Rumelhart 1985] David E Rumelhart, Geoffrey E Hinton and Ronald J Williams. *Learning internal representations by error propagation*. Rapport technique, DTIC Document, 1985. 27

[Russell 1995] Stuart Russell, Peter Norvig and Artificial Intelligence. *A modern approach*. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, vol. 25, 1995. 27

[Ryan 2014] Geraldine D Ryan, Lisa Emiljanowicz, Simone A Haerri and Jonathan A Newman. *Aphid and host-plant genotype× genotype interactions under elevated CO2*. Ecological Entomology, vol. 39, no. 3, pages 309–315, 2014. 125

[Saïghi 2008] Sylvain Saïghi, Laure Buhry, Yannick Bornat, Gilles N'Kaoua, Jean Tomas and Sylvie Renaud. *Adjusting the neurons models in neuromimetic ICs using the voltage-clamp technique*. In Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pages 1564–1567. IEEE, 2008. 33

[Sanfeliu 1983] Alberto Sanfeliu and King-Sun Fu. *A distance measure between attributed relational graphs for pattern recognition*. Systems, Man and Cybernetics, IEEE Transactions on, no. 3, pages 353–362, 1983. 88

[Schliebs 2005] Reinhard Schliebs. *Basal forebrain cholinergic dysfunction in Alzheimer's disease-interrelationship with β-amyloid, inflammation and neurotrophin signaling*. Neurochemical Research, vol. 30, no. 6-7, pages 895–908, 2005. 47

[Schliebs 2009a] Stefan Schliebs, Michaël Defoin-Platel and Nikola Kasabov. *Integrated feature and parameter optimization for an evolving spiking neural network.* In Advances in Neuro-Information Processing, pages 1229–1236. Springer, 2009. xiii, 39, 42

[Schliebs 2009b] Stefan Schliebs, Michaël Defoin-Platel, Sue Worner and Nikola Kasabov. *Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models.* Neural Networks, vol. 22, no. 5, pages 623–632, 2009. 42

[Schliebs 2010] Stefan Schliebs, Nikola Kasabov and Michael Defoin-Platel. *On the probabilistic optimization of spiking neural networks.* International Journal of Neural Systems, vol. 20, no. 06, pages 481–500, 2010. 4, 26

[Schliebs 2011] Stefan Schliebs, Haza Nuzly Abdull Hamed and Nikola Kasabov. *Reservoir-based evolving spiking neural network for spatio-temporal pattern recognition.* In Neural Information Processing, pages 160–168. Springer, 2011. xiv, 26, 47, 52, 53, 59

[Schliebs 2014] Stefan Schliebs and Nikola Kasabov. *Computational Modeling with Spiking Neural Networks.* In Springer Handbook of Bio-/Neuroinformatics, pages 625–646. Springer, 2014. 49

[Schliep 2003] Alexander Schliep, Alexander Schönhuth and Christine Steinhoff. *Using hidden Markov models to analyze gene expression time course data.* Bioinformatics, vol. 19, no. suppl 1, pages i255–i263, 2003. 26

[Schneider 2008] Nathan C Schneider and Daniel Graupe. *A modified LAMSTAR neural network and its applications.* International Journal of Neural Systems, vol. 18, no. 04, pages 331–337, 2008. 47

[Schrauwen 2003] Benjamin Schrauwen and Jan Van Campenhout. *BSA, a fast and accurate spike train encoding scheme.* In Proceedings of the International Joint Conference on Neural Networks, volume 4, pages 2825–2830. IEEE Piscataway, NJ, 2003. 66

[Schreiber 1995] Guus Schreiber, Bob Wielinga and Wouter Jansweijer. *The KACTUS view on the'O' word.* In IJCAI workshop on basic ontological issues in knowledge sharing, pages 159–168, 1995. 148

[Sèguier 2002] Renaud Sèguier and David Mercier. *Audio-visual speech recognition one pass learning with spiking neurons.* In Artificial Neural Networks-ICANN 2002, pages 1207–1212. Springer, 2002. 42, 48

[Shabo 2007] Amnon Shabo. *Health record banks: integrating clinical and genomic data into patient-centric longitudinal and cross-institutional health records.* 2007. 2

[Shaposhnikov 2014] Dmitry Shaposhnikov, Boris Revich, Yuri Gurfinkel and Elena Naumova. *The influence of meteorological and geomagnetic factors on acute myocardial infarction and brain stroke in Moscow, Russia.* International journal of biometeorology, vol. 58, no. 5, pages 799–808, 2014. 96, 97

[Shrager 1987] Jeff Shrager, Tad Hogg and Bernardo A Huberman. *Observation of phase transitions in spreading activation networks.* Science, vol. 236, no. 4805, pages 1092–1094, 1987. 91, 138

[Sobel 1987] Eugene Sobel, ZX Zhang, Milton Alter, SM Lai, Zoreh Davanipour, Gary Friday, Robert McCoy, Tish Isack and Lawrence Levitt. *Stroke in the Lehigh Valley: seasonal variation in incidence rates.* Stroke, vol. 18, no. 1, pages 38–42, 1987. 109

[Solomon 1999] S Solomon, RW Portmann, RW Sanders, JS Daniel, W Madsen, B Bartram and EG Dutton. *On the role of nitrogen dioxide in the absorption of solar radiation.* Journal of Geophysical Research: Atmospheres (1984–2012), vol. 104, no. D10, pages 12047–12058, 1999. 114

[Soltic 2008] Snjezana Soltic, Simei Gomes Wysoski and Nikola K Kasabov. *Evolving spiking neural networks for taste recognition.* In Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pages 2091–2097. IEEE, 2008. 42

[Soltic 2010] Snjezana Soltic and Nikola Kasabov. *Knowledge extraction from evolving spiking neural networks with rank order population coding*. International Journal of Neural Systems, vol. 20, no. 06, pages 437–445, 2010. 26

[Soltic 2011] S Soltic and N Kasabov. *A Biologically Inspired Evolving Spiking Neural Model with Rank-Order Population Coding and a Taste Recognition System Case Study*. System and Circuit Design for Biologically-inspired Intelligent Learning, page 136, 2011. 26

[Song 2004] Qun Song and Nikola Kasabov. *TWRBF–Transductive RBF Neural Network with Weighted Data Normalization*. In Neural Information Processing, pages 633–640. Springer, 2004. 14

[Song 2006] Qun Song and Nikola Kasabov. *TWNFI-a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling*. Neural Networks, vol. 19, no. 10, pages 1591–1596, 2006. 15, 22

[Stern 2008] David L Stern. *Aphids*. Current Biology, vol. 18, no. 12, pages R504–R505, 2008. 122, 123

[Stufflebeam 2008] Robert Stufflebeam. *Neurons, synapses, action potentials, and neurotransmission*. Consortium on Cognitive Science Instruction, 2008. xiii, 28, 29, 30

[Stufkens 2000] MAW Stufkens, DAJ Teulon, D Nicol and JD Fletcher. *Implications of aphid flight patterns for pest management of potatoes*. In Proceedings of The New Zealand Plant Protection Conference, pages 78–82. New Zealand Plant Protection Society; 1998, 2000. 121

[Suissa 2013] Laurent Suissa, Mikael Fortier, Sylvain Lachaud, Pascal Staccini and Marie-Hélène Mahagne. *Ozone air pollution and ischaemic stroke occurrence: a case-crossover study in Nice, France*. BMJ Open, vol. 3, no. 12, page e004060, 2013. 105

[Sun 2011] Yucheng Sun and Feng Ge. *How do aphids respond to elevated CO 2?* Journal of Asia-Pacific Entomology, vol. 14, no. 2, pages 217–220, 2011. 125

[Suykens 1999] Johan AK Suykens and Joos Vandewalle. *Least squares support vector machine classifiers.* Neural Processing Letters, vol. 9, no. 3, pages 293–300, 1999. 16

[Szatmáry 2010] Botond Szatmáry and Eugene M Izhikevich. *Spike-timing theory of working memory.* PLoS Computational Biology, vol. 6, no. 8, page e1000879, 2010. xiv, 44, 45

[Teulon 1999] DAJ Teulon, MAW Stufkens, D Nicol and SJ Harcourt. *Forecasting barley yellow dwarf virus in autumn-sown cereals in 1998.* In Proceedings of the New Zealand Plant Protection Conference, pages 187–191. New Zealand Plant Protection Society; 1998, 1999. 125

[Teulon 2004a] DAJ Teulon, GO Lankin, MAW Stufkens, J Lee, GR Travis*et al. Local variation in cereal aphid flight activity in Canterbury.* New Zealand Plant Protection, vol. 57, page 221, 2004. 127

[Teulon 2004b] DAJ Teulon, MAW Stufkens, JD Fletcher*et al. Crop infestation by aphids is related to flight activity detected with 7.5 metre high suction traps.* New Zealand Plant Protection, vol. 57, page 227, 2004. 127

[Teulon 2008] DAJ Teulon, CM Till, RF van Toor*et al. Conditions surrounding the outbreak of yellow dwarf virus in autumnwinter-sown cereals in Canterbury during 2005.* New Zealand Plant Protection, vol. 61, pages 270–276, 2008. 121

[Thorpe 1997] Simon J Thorpe. *How can the human visual system process a natural scene in under 150ms? Experiments and neural network models.* In ESANN, 1997. 48

[Thorpe 1998] Simon Thorpe and Jacques Gautrais. *Rank order coding.* In Computational Neuroscience, pages 113–118. Springer, 1998. xiii, 38, 66

[Tobias 2002] Martin Tobias, Jit Cheung and Harry McNaughton. Modelling stroke: a multi-state life table model. Ministry of Health, 2002. 93

[Tsai 2003] Shang-Shyue Tsai, William B Goggins, Hui-Fen Chiu and Chun-Yuh Yang. *Evidence for an association between air pollution and daily stroke*

admissions in Kaohsiung, Taiwan. Stroke, vol. 34, no. 11, pages 2612–2616, 2003. 96, 111

[Tsapatsoulis 2007] Nicolas Tsapatsoulis, Konstantinos Rapantzikos and Constantinos Pattichis. *An embedded saliency map estimator scheme: Application to video encoding.* International Journal of Neural Systems, vol. 17, no. 04, pages 289–304, 2007. 47

[Tu 2014] Enmei Tu, Nikola Kasabov, Muhaini Othman, Yuxiao Li, Susan Worner, Jie Yang and Zhenghong Jia. *NeuCube (ST) for spatio-temporal data predictive modelling with a case study on ecological data.* In Neural Networks (IJCNN), 2014 International Joint Conference on, pages 638–645. IEEE, 2014. 62, 76, 122

[Turaga 2008] Pavan Turaga, Rama Chellappa, VS Subrahmanian and Octavian Udrea. *Machine Recognition of Human Activities: A Survey.* IEEE Transaction on Circuits and Systems for Video Technology, vol. 18, no. 11, page 1473, 2008. 26

[Vapnik 1963] Vladimir Vapnik. *Pattern recognition using generalized portrait method.* Automation and Remote Control, vol. 24, pages 774–780, 1963. 3, 15

[Vapnik 1998] Vladimir Naumovich Vapnik and Vlamimir Vapnik. Statistical learning theory, volume 2. Wiley New York, 1998. 14, 20

[Verma 2009] Anju Verma, Nikola Kasabov, Elaine Rush and Qun Song. *Ontology based personalized modeling for chronic disease risk analysis: An integrated approach.* In Advances in Neuro-Information Processing, pages 1204–1210. Springer, 2009. 150

[Verstraeten 2005] David Verstraeten, Benjamin Schrauwen and Dirk Stroobandt. *Isolated word recognition using a Liquid State Machine.* In ESANN, volume 5, pages 435–440, 2005. 47

[von der Malsburg 1988] Christoph von der Malsburg. *Pattern recognition by labeled graph matching.* Neural Networks, vol. 1, no. 2, pages 141–148, 1988. 88

[Wang 2008] Xiuqing Wang, Zeng-Guang Hou, Anmin Zou, Min Tan and Long Cheng. *A behavior controller based on spiking neural networks for mobile robots.* Neurocomputing, vol. 71, no. 4, pages 655–666, 2008. 47

[Watts 2007a] Michael J Watts and Sue P Worner. Comparison of multi-layer perceptrons and simple evolving connectionist systems over the lincoln aphid data set. Lincoln University. Bio-Protection & Ecology Division, 2007. 127, 130

[Watts 2007b] Michael J Watts and Sue P Worner. Further sensitivity analysis of simple evolving connectionist systems applied to the lincoln aphid data set. Lincoln University. Bio-Protection & Ecology Division, 2007. 127

[Wolf 1991] Philip A Wolf, Ralph B D'Agostino, Albert J Belanger and William B Kannel. *Probability of stroke: a risk profile from the Framingham Study.* Stroke, vol. 22, no. 3, pages 312–318, 1991. 2

[Worner 2002] SP Worner, GO Lankin, S Samarasinghe, DAJ Teulon, SM Zydenbos*et al. Improving prediction of aphid flights by temporal analysis of input data for an artificial neural network.* New Zealand Plant Protection, pages 312–316, 2002. 125, 127, 128, 130, 139

[Wysoski 2006] Simei Gomes Wysoski, Lubica Benuskova and Nikola Kasabov. *Online learning with structural adaptation in a network of spiking neurons for visual pattern recognition.* In Artificial Neural Networks–ICANN 2006, pages 61–70. Springer, 2006. 4, 38, 42, 47, 48, 68

[Wysoski 2007] Simei Gomes Wysoski, Lubica Benuskova and Nikola Kasabov. *Text-independent speaker authentication with spiking neural networks.* In Artificial Neural Networks–ICANN 2007, pages 758–767. Springer, 2007. 38, 42, 48

[Wysoski 2008a] Simei Gomes Wysoski. *Evolving spiking neural networks for adaptive audiovisual pattern recognition.* PhD thesis, Auckland University of Technology, 2008. xiv, 49

[Wysoski 2008b] Simei Gomes Wysoski, Lubica Benuskova and Nikola Kasabov. *Fast and adaptive network of spiking neurons for multi-view visual pattern recognition*. Neurocomputing, vol. 71, no. 13, pages 2563–2575, 2008. 48

[Wysoski 2010] Simei Gomes Wysoski, Lubica Benuskova and Nikola Kasabov. *Evolving spiking neural networks for audiovisual information processing*. Neural Networks, vol. 23, no. 7, pages 819–835, 2010. 69

[Yang 1995] L Yang, BK Widjaja and R Prasad. *Application of hidden Markov models for signature verification*. Pattern Recognition, vol. 28, no. 2, pages 161–170, 1995. 26

[Yang 2014] Wan-Shui Yang, Xin Wang, Qin Deng, Wen-Yan Fan and Wei-Ye Wang. *An evidence-based appraisal of global association between air pollution and risk of stroke*. International Journal of Cardiology, vol. 175, no. 2, pages 307–313, 2014. 96, 111

[Yau 2007] Wai Chee Yau, Dinesh Kant Kumar and Sridhar Poosapadi Arjunan. *Visual recognition of speech consonants using facial movement features*. Integrated Computer-Aided Engineering, vol. 14, no. 1, pages 49–61, 2007. 47

[Yusuf 1998] Hussain R Yusuf, Wayne H Giles, Janet B Croft, Robert F Anda and Michele L Casper. *Impact of multiple risk factor profiles on determining cardiovascular disease risk*. Preventive Medicine, vol. 27, no. 1, pages 1–9, 1998. 2

[Zhang 2004] Li Zhang, Weida Zhou, Licheng Jiao*et al.* *Wavelet support vector machine*. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 1, pages 34–39, 2004. 16

[Zhou 2004] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston and Bernhard Sch'olkopf. *Learning with local and global consistency*. Advances in Neural Information Processing Systems, vol. 16, no. 16, pages 321–328, 2004. 91, 110, 138

[Zhou 2012] Feng Zhou and Fernando De la Torre. *Factorized graph matching.* In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 127–134. IEEE, 2012. 88