

Adaptive Methods for Spatiotemporal Stream Data Mining

Reggio N. HARTONO

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

Supervisors:

Assoc. Prof. Russel PEARS

Prof. Nikola KASABOV

prepared at The Knowledge Engineering and Discovery Research
Institute, KEDRI

**School of Engineering, Computer and
Mathematical Sciences**

Abstract

The availability of temporal and spatiotemporal data is increasing, and the use of traditional statistical techniques to deal with such data is insufficient. Novel methods that are capable of adapting to changing patterns in time-variant spatiotemporal data need to be developed. To achieve this objective, the thesis proposes three different methods to deal with various types of data and employ distinct approaches to tackle common problems faced in spatiotemporal data mining.

The first method deals with multiple time series and presents a development of a generic framework to extract knowledge in the form of temporal rules. The main component is a modified association rule mining algorithm that also works with time dimension, producing rules in the form of $A \xrightarrow{T} B$. As part of the research, a discretisation technique inspired by concept drift detection is also proposed. The framework was then applied to a dataset that tracks the number of aphids caught in traps along with weather variables over almost twenty years in the Lincoln region in Canterbury, New Zealand.

The second method deals with building local models for a time-step ahead spatial prediction problem. Taking advantage of the locality-preserving property of the space-filling Hilbert curve, the method is able to work with existing concept drift detection algorithms to automatically determine where and when in the spatiotemporal landscape that patterns are changing. The framework was tested on the earthquake catalogue data around the Christchurch region. The empirical results reveal that the local models improved the prediction accuracy of up to 9% on one of the tests when compared to a standard incremental model building approach based on a fixed size sliding window scheme.

The third method employs a Spiking Neural Network (SNN)-based system

called NeuCube to build an early event prediction system. The system was trained to differentiate the seismicity readings obtained from spatially scattered seismograms around Canterbury both before large earthquakes happen and periods of low seismicity between 2010 and 2016. The system was tested to examine whether NeuCube could learn from complex data and demonstrate a capability of predicting large earthquakes with a reasonable window of time. The results from this scheme are promising as NeuCube could predict major seismic events with a much higher true positive rate (0.91) while keeping the false positive rate significantly lower (0.08) when compared to other prediction algorithms.

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 extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Auckland, 2016

Reggio N. Hartono

Acknowledgement

This PhD study has been a humbling and amazing journey both for my personal and academic life. I want to extend my utmost gratitude to the people and organisations who have given me support and made this life-changing experience possible.

Foremost, I want to express my deepest gratitude to my main supervisor Assoc. Prof. Russel Pears. He has been patient with me and has imparted so much knowledge which has really taught me a lot and changed the way I think about research. I really appreciate both the constructive criticisms and positive affirmations throughout the mentorship, which has enabled me to work better and given me the motivation I needed to persevere.

My sincere appreciation also goes to my second supervisor Prof. Nikola Kasabov, who has given me the chance to be a part of KEDRI and has mentored me all this time. I have learnt a lot of revolutionary ideas in the research field and also from his leadership, wisdom, and enthusiasm. I am also thankful for the invitations to his home and for the impromptu music jam sessions we played with the accordion and guitar.

I would also like to thank and acknowledge other researchers who have helped me. Thanks to Assoc. Prof. Sue Worner from Lincoln University who has provided support with the aphids data and has invited me to present my research. I am also indebted to Dr. Matt Gerstenberger from GNS Science who has helped me understand the earthquake datasets and has given me a new level of appreciation for Seismology. Special thanks to Dr. Mike Watts whose weekly visits to KEDRI have brought about feedbacks that have been really helpful.

My PhD study would not be possible without the scholarship and funding from The Ministry of Foreign Affairs and Trade under the NZ-ASEAN

Scholars award scheme. I am very thankful for this opportunity and the assistance that I have received, especially from my AUT scholarship officer Ms. Sacha Pointon.

My dearest appreciation goes to all the members and former members of KEDRI (a tremendous group, folks), who have given me the friendship and support that have made the lab a fun place to work at. Thanks for the discussions, inputs, and laughs that we all had. A very special thanks to Joyce D'Mello, who has been like a mother to all of us at KEDRI and has been a dear counsel for me.

Finally, I would like to extend a personal gratitude to all my friends and family who have given me the strength to continue and made my life meaningful. I want to especially thank my parents, who have supported and inspired me to push my boundaries. My heartfelt appreciation to my fiancé Chika Lim, whose love and support have been steadfast and made everything much more beautiful.

To all the people that have been a part of my journey, who I have and have not mentioned, thank you.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Objectives	3
1.3	Contributions	4
1.4	Thesis Structure	5
1.5	Publications	5
2	Rule Discovery from Multiple Time-series Stream	8
2.1	Introduction	8
2.2	Time-series Data Mining	9
2.3	Time-series Discretisation	9
2.3.1	Symbolic Aggregate approXimation (SAX)	11
2.4	Association Rule Mining	13
2.4.1	Definition	14
2.5	Proposed Framework	16
2.5.1	Sequential Pattern Mining	16
2.5.2	Problem Definition	17
2.5.3	Adaptive-window Discretisation	18
2.5.4	Itemset candidate generation	21
2.5.5	Rule Extraction	21
2.5.6	Rule Selection criteria	23
2.5.7	Rule Format Extension	24
2.6	Experimental Study	24
2.6.1	The Dataset	25
2.6.2	Experimental Configuration	26
2.6.3	Effects of Minimum Support Threshold	26
2.6.4	Effect of Window Size	27

2.6.5	Effects of Rule Antecedent Variable	29
2.6.6	Conclusions	32
2.7	Conclusions	33
3	Adaptive Local Models for Spatiotemporal Data Analysis	34
3.1	Introduction	34
3.2	Local Models	35
3.3	Concept-drift	38
3.3.1	Definition	39
3.3.2	Detection	40
3.3.3	The ADWIN Algorithm	42
3.4	The Hilbert Curve	43
3.4.1	Construction	43
3.4.2	Locality preservation	45
3.5	Proposed Methodology	46
3.5.1	Problem Definition	46
3.5.2	Adaptive Method	47
3.5.3	Performance measure	50
3.6	Conclusions	52
4	Earthquake Aftershocks Modelling Experiment	53
4.1	Introduction	53
4.2	The Dataset	53
4.2.1	The Seismicity of Christchurch	54
4.2.2	Data Acquisition and Preprocessing	54
4.2.3	Data Characteristics	56
4.3	Experimental Design	59
4.4	Results and Analysis	61
4.4.1	Visualisations	64
4.5	Conclusions	67

5	The NeuCube Architecture for Spatiotemporal Data Analysis	71
5.1	Introduction	71
5.2	The Neuron	72
5.3	Artificial Neural Networks	74
5.4	Spiking Neural Networks	75
5.4.1	The Hodgkin-Huxley Neuron Model	75
5.4.2	Leaky Integrate and Fire Model	78
5.5	Spike Encoding	79
5.5.1	Rate coding	80
5.5.2	Temporal coding	80
5.5.3	Population coding	82
5.6	SNN Learning Algorithm	83
5.6.1	STDP	83
5.7	Reservoir Computing	85
5.8	Evolving Connectionist System (ECOS)	87
5.9	The NeuCube Architecture	88
5.9.1	Input Data Encoding	91
5.9.2	SNN Reservoir	93
5.9.3	Evolving Output Classification	94
5.9.4	Applications	96
5.10	Conclusions	98
6	Seismogram Analysis for Event Prediction Experiment	99
6.1	Introduction	99
6.2	Research in Earthquake Prediction	100
6.2.1	Earthquake Precursors	100
6.2.2	Computational Intelligence Methods for Earthquake Prediction	101
6.3	NeuCube for Spatiotemporal Data	103

6.4	The Seismometer	104
6.4.1	The GeoNet Project & The New Zealand National Seismograph Network	105
6.5	Experiment: Very Large Earthquakes Prediction	106
6.5.1	Experiment Design	106
6.5.2	Data Acquisition and Preparation	109
6.5.3	Sampling Bias	112
6.5.4	Results	114
6.5.5	Analysis	118
6.5.6	Individual Earthquake Analysis	122
6.6	Visualisation	125
6.7	Conclusions	127
7	Conclusions	130
7.1	Summary	130
7.2	Rule Mining from Multiple Time-series	130
7.2.1	Future Research Directions	131
7.3	Adaptive Local Models	131
7.3.1	Future Research Directions	132
7.4	NeuCube for Seismic Data Analysis	133
7.4.1	Future Research Directions	134
	Bibliography	136

List of Figures

2.1	An illustration of PAA transformation (with time as X-axis) as an attempt to model a time series with a linear combination of box basis functions. In this case, a sequence of length 128 is reduced to eight dimensions [Lin 2007].	12
2.2	A time series is discretised by first obtaining a PAA approximation and then using predetermined breakpoints to map the PAA coefficients into SAX symbols. In the example above, with $n = 128$, $w = 8$ and $a = 3$, the time series is mapped to the word baabccbc [Lin 2007].	13
2.3	Discretisation result comparison. Values of 0.1 and 0.2 have been selected to signify which part of the time-series signal is discretised as a high-signal by SAX (box marks) and the proposed sliding window approach (circle marks).	19
2.4	A macro habitus photograph of a wingless adult <i>Rhopalosiphum padi</i> , one of the species of aphids that are collected in the dataset. Photo: Brendan Wray, AphID, USDA APHIS ITP, Bugwood.org	25
2.5	Number of rules generated with different <i>minconf</i> and <i>minsupp</i>	27
2.6	Number of rules generated with different window size of discretisation	28
2.7	Number of rules generated by different antecedent variables . .	29
2.8	Rule visualization: ddddaaa to b	30
2.9	Rule visualization: aaaabcbddd to b	31
2.10	Rule visualization: aaaaabbb, dddcb to b	32

3.1	Illustration of the difference between global (left) and local modelling (right) in trying to fit data in the input space. The local linear regression models are built based on clusters of sample data set in a 2-dimensional space.	36
3.2	Spatial data may demonstrate a pattern of positive spatial autocorrelation (left), negative spatial autocorrelation (right), or a pattern that is not spatially autocorrelated (center). [Radil 2011]	36
3.3	Hilbert curves with various iteration order number n	45
3.4	Projection of a 3-dimensional Hilbert curve	45
3.5	A visualisation of a second-order Hilbert curve with the scanning order derived from it.	48
4.1	The map of New Zealand and the studied region. Coordinates: North -43.0096, East 173.3741, South -44.1974, and West 171.3132.	55
4.2	Seismic shocks in the studied region and time period, visualised in 3D as point-process, displaying concept drifts both in space and time. Vertical axis is time and the horizontal axes are longitude and latitude.	57
4.3	The spatial point map of the earthquakes recorded 24 hours after the 2010 Darfield earthquake (left) and the discretised version as a grayscale heatmap (right). Each pixel represents a 4 km ² area (2 x 2 km resolution).	58
4.4	Time-ordered spatially-aggregated daily frequency of earthquakes from January 2010 to November 2015 occurring within the studied area. (Y-axis clipped at 90)	59
4.5	Illustration of raster scan-line.	61

4.6	Time-ordered daily frequency of earthquakes along the study period showing the resulting N-Test prediction. (Y-axis clipped at 150)	63
4.7	A graph displaying the N-Test error rate of both the local and global approach at every time frame.	64
4.8	The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Hilbert curve traversal (right).	65
4.9	The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Hilbert curve traversal (right).	65
4.10	This figure shows the local spaces made for time frame 248 (left) and 419 (right) with the aggregate number of shocks in each locality to illustrate the inter-group variance separated by the method.	66
4.11	The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Scanline fill traversal (right). It produced vertically separated local regions which do not preserve locality.	67
4.12	The actual data at time frame 253 (left) and the hotspot prediction created by LR-Local mode (right). PAI: 26.780. The predicted hotspots closely follow the true spatial distribution of aftershocks.	68
4.13	The actual data at time frame 272 (left) and the hotspot prediction created by LR-Local mode (right). PAI: 6.720. The decrease in score reflects how the hotspots following the true spatial distribution of aftershocks more loosely.	69

4.14	The actual data at time frame 306 (left) and the hotspot prediction created by LR-Local mode (right). Although the prediction can generally indicate the spatial distribution of aftershocks, the PAI score is 0 (complete miss). This demonstrates the importance of using N-test to complement PAI.	69
4.15	A slice projection of the 3-dimensional heatmap visualising the identified gradually evolving concepts in the dataset over space and time. The Vertical axis is time and every colour represents a concept.	70
5.1	Golgi staining works by soaking brain tissue in silver chromate solution. Using the stain, a small percentage of the neurons became darker, revealing their total structures.	72
5.2	Line art drawing of a Neuron. The numbers indicate the structures that constitute a Neuron: 1. Dendrites, 2. Cell Body, 3. Nucleus, 4. Axon	73
5.3	A diagram showing how the Perceptron works.	74
5.4	Hodgkin-Huxley type models represent the biophysical characteristics of cell membranes. The lipid bilayer is represented as a Capacitance (C_m). Voltage-gated and leak ion channels are represented as by nonlinear (g_n) and linear (g_L) conductances, respectively. The electrochemical gradients driving the flow of ions are represented by batteries (E), and ion pumps and exchanges are represented by current sources (I_p). Image: Behrang Amini.	76

5.5	Schematic diagram of the integrate-and-fire model. The basic circuit is the module inside the dashed circle on the right-hand side. A current $I(t)$ charges the RC circuit. The voltage $u(t)$ across the capacitance (points) is compared to a threshold ϑ . If $u(t) = \vartheta$ at time $t_i(f)$ an output pulse $\delta(t - t_i(f))$ is generated. Left part: A presynaptic spike $\delta(t - t_j(f))$ is low-pass filtered at the synapse and generates an input current pulse $\alpha(t - t_j(f))$.	78
5.6	The characteristic of LIF neurons where the membrane potential, increase with every input spike at a time t multiplied to the synaptic efficacy (strength) until it reaches a threshold and emit an output spike then the membrane potential is reset to initial state.	79
5.7	Rank Order Coding [Thorpe 1998]	81
5.8	Spike-Timing Dependent Plasticity (schematic): The STDP function shows the change of synaptic connections as a function of the relative timing of pre- and postsynaptic spikes . . .	84
5.9	The general schematic diagram of reservoir computing principle.	86
5.10	Architecture of an LSM. A function of time (time series) $u(\cdot)$ is injected as input into the liquid filter L^M , creating at time t the liquid state $x^M(t)$, which is transformed by a memoryless readout map f^M to generate an output $y(t)$ [Maass 2002]. . .	87
5.11	A schematic diagram of a NeuCube architecture for brain data modelling [Kasabov 2012].	89
5.12	The spatial distribution of the EEG nodes on an Emotiv EEG headset (left) and the location of the input neurons in NeuCube's reservoir network (right).	90
5.13	Thresholding method of encoding continuous time-series data into a spike train (left) and the reconstruction of the signal from the spike train (right)[Kasabov 2014b].	91

5.14	Sine wave converted using BSA. The original sine wave is plotted in dotted line, and the converted version in solid line. The lower plot visualises the spike train. [Schrauwen 2003].	92
5.15	Evolving SNN (eSNN) for classification [Kasabov 2007].	94
6.1	Simplified block diagram of the NeuCube Architecture	102
6.2	Schematic diagram of a seismometer. Ground motion causes the base and frame of the seismometer to oscillate. The mass, suspended by the spring and boom, tends to stay in one place because of inertia. The relative motion of the base as compared to the mass is recorded as the output of the seismometer. Image: Larry Braile, Purdue Uni.	104
6.3	The New Zealand National Seismograph Network consists of primary sites located at approximately 100 km spacing, with extra regional sites established at places of geophysical significance.	105
6.4	New Zealand National Seismograph Network with the 4 selected sites around Canterbury area grayed (McQueen's Valley: MQZ, Oxford: OXZ, Lake Taylor Station: LTZ, and Kahutara: KHZ)	109
6.5	This timeline illustrates the relationship between the actual time the earthquake starts, the offset, and the observation duration used in this experiment.	110
6.6	Raw seismogram (left), and after pre-processing (right). X-axis is the discrete time unit and Y-axis is the seismicity readings.	111
6.7	Preprocessed seismogram and the resulting spike train	113

-
- 6.8 SNN reservoir with input neurons and synapses after training. The line thickness and arrow signify the connection weight and direction respectively, which are obtained by applying STDP unsupervised learning with the training input data. The more spikes transmitted between two neurons the stronger they are connected to each other. 117
- 6.9 The effect of the prediction horizon to the accuracy of different classifiers. Performance deteriorates the further away readings are taken from the events. 118
- 6.10 Neuron proportion based on spike communication clustering for negative samples. 119
- 6.11 The new active fault sources of northeastern Canterbury developed in the new model by Russ Van Dissen of GNS Science. Overlapping sources are shown side-by-side to distinguish them from one-another [Stirling 2008]. 120
- 6.12 Total interaction between the input neurons for negative samples. LTZ is not temporally correlated with the other inputs in periods of low seismicity. 121
- 6.13 Total interaction between the input neurons for positive samples. 122
- 6.14 Neuron proportion based on spike communication clustering for positive samples. 123
- 6.15 Structural domains of the Canterbury region. The domains are as follows: DOM 1=Marlborough Fault Zone; DOM2,3 & 4=West Culverden Fault Zone, Porters Pass-Amberley Fault Zone & North Canterbury Fold & Thrust Belt; DOM 5&6=Mt Hutt-Mt Peel Fault Zone & South Canterbury Zone; DOM 7=Canterbury Plains Zone;DOM 8=Southern Alps Zone; and DOM 9=Alpine Fault Zone. [Stirling 2001]. 124

6.16 Spike raster plot of the activity in the reservoir 5 days before the September 2010 Darfield earthquake. Notice the increase of activity on the LTZ channel prior to the earthquake, and the spiking activity of the input neurons are becoming highly correlated to each other.	125
6.17 Spike raster plot of the activity in the reservoir 5 days before the earthquake in June 2011. Although there is no increase in seismic activity, LTZ channel became highly correlated with the other stations, spiking in sync with MQZ and KHZ. . . .	126
6.18 Spike raster plot taken from a sample of low-seismicity period.	127
6.19 NeuCube SNN reservoir rendered in a 3D Virtual Reality environment on top of a map of New Zealand, enabling users to immerse themselves and walk around the neurons and observe the connection building and spiking activity in time and space.	128
6.20 NeuCube SNN reservoir rendered in a 3D Virtual Reality environment on top of a map of New Zealand, showing the spiking activity of the cube.	129

List of Tables

2.1	Some rules produced by the method	30
4.1	The result of the predictive accuracy test on the whole dataset. Higher is better for PAI, lower is better for N-Test. The model count tracks the number of local models built by the system. .	62
6.1	Earthquakes within the Canterbury region used as positive samples	108
6.2	Classification accuracy result with varying prediction horizon length	115
6.3	Classification output results with stream data from 2010-2015	116

Introduction

1.1 Background and Motivation

The world is seeing revolutionary advances and price reduction of technologies for the collection of temporal, spatial, and spatiotemporal data. This includes but not limited to remote sensing satellites, smartphones, sensor networks, and multitudes of other devices that facilitate the collection of data that can be referenced in space and time. As the number, volume, and resolution of spatiotemporal datasets increase, researchers from various disciplines are starting to look into data mining as the tool of choice for analysing this breed of data as it is becoming more apparent that the use traditional statistical methods is sometimes insufficient.

This type of data often conceal interesting information and rich interactions between space-time dimensions which conventional systems and classical data mining techniques are unable to discover or simply treat just like any other variable or feature. The field of spatiotemporal data mining (STDM) emerged out of the need to create effective and efficient techniques that are able to deal with the intricacy and complexity of spatiotemporal datasets to discover meaningful analyses and knowledge [Yao 2003].

One of the domain areas in which STDM is gaining traction is in the fields of geospatial and ecological modelling, which increasingly have to deal with high-volume data collected over vast spatial range and over long periods of time [Han 2002]. Researchers in these fields mainly use statistical tech-

niques to analyse the relationships between the variables in a data set. These techniques are typically parametric and are unable to handle the complexities embedded in high-speed, high-volume, and highly-volatile spatiotemporal data. This is in contrast to data mining methods which have been noted as being capable to work mostly without any parametric assumption about the data and are flexible enough to be used in various ecological and geospatial applications [Hochachka 2007].

These emerging applications involve great data analysis challenges that also represent new opportunities for data mining research. The development and applications of STDM methods to mine biomedical, telecommunication, geospatial, and climate data are presently a very active research area. However, in light of the tremendous amount of fast growing and sophisticated types of data and comprehensive analysis tasks, the field of STDM may well be only in its infancy. Research is still needed to develop automated, scalable, integrated, and reliable data mining systems and tools.

This thesis is an exploration of various data mining paradigms and techniques to deal with problems commonly encountered in STDM. While the proposed methods encompass a wide range of approaches and tackle different sets of problems, there is a common theme and spirit that these methods have. They work in an adaptive manner and strive toward the interpretability of the models for the sake of knowledge discovery. The development of systems that are explainable adds to the ability of human experts to discover new knowledge. This transparency has the potential to make data mining the key tool for not only making predictions and forecasts, but also for a better understanding of the problem at hand.

The proposed methods were tested and applied on real-world geospatial and ecological modelling datasets. The experiments exemplify the various types of data with time and/or space referenced in it. The examples concerning earthquake include both numerical aftershock analysis helpful for

post-event disaster management and event prediction useful for pre-event warning systems. These problem domains are very suitable and opportune fields to run STDN experiments on, and are a treasure trove of knowledge and information that could lead to improving life quality and even saving lives. For instance, one of the case studies presented in this thesis was aimed to discover human-readable temporal patterns that precede pest outbreaks, which are a big threat to both the agriculture industry and also the fragile and highly-regarded biosecurity of New Zealand [Teulon 2002]. The other case studies revolve around the earthquakes of the city of Christchurch, where seismic related disasters since 2010 have killed over 180 people and up until 2016, have cost the nation about forty billion dollars to rebuild the city [Wood 2016].

1.2 Objectives

In line with the problems expounded in the previous section regarding STDN, this PhD study has the main objectives of:

1. Developing novel methods for data analysis which take into account temporal, spatial, and spatiotemporal dimensions and have the capability to identify, adapt, and learn as new observations or concepts become available;
2. Investigating further the utilisation of new computing paradigms based on Spiking Neural Networks (SNN) in working with spatiotemporal dataset and its potential to be used in hazardous event prediction.

1.3 Contributions

Throughout the thesis, several contributions have been made both to the field of machine learning and the applications in their respective scientific domains. These contributions can be listed as follows:

- A generic framework for extracting temporal association rules from multiple time-series is presented. The method works by first discretising the time-series data, generating the frequent itemsets, and then using a modified Apriori algorithm to mine the rules. Finally, some rules selection criteria are applied to obtain the best rules.
- As a part of the rule extraction research, a novel binary discretisation algorithm is also proposed. The algorithm takes into account changes that occur between two consecutive sliding windows in order to discretise real value into symbolic representation.
- A method to build local models that can be used with spatiotemporal data is proposed. The method detects concept drifts in space and time by combining the Hilbert curve and concept-drift detector. It identifies local regions that are homogeneous and this information is used to build local models.
- An experimental study of the adaptive local methods with aftershock modelling of the area around the city of Christchurch is discussed.
- A preliminary research on the application of an SNN-based spatiotemporal data machine called NeuCube to analyse seismic data for earthquake prediction is presented. To the best of my knowledge, this approach of analysing seismic waveforms has not been done before.

1.4 Thesis Structure

In departure from the usual manner of presenting theoretical frameworks and existing body of knowledge, this thesis does not have a dedicated chapter for literature reviews. Instead, each individual chapter starts with sections that lay out the relevant foundational concepts along with a review of previous and contemporary research pertinent to the subject matter. This thesis is organised into four major parts:

- Chapter 2 presents the framework that can be used to discover temporal association rules from multivariate time-series. It is presented along with a case study of the methods on an ecological dataset.
- Chapter 3 and 4 discuss the technique devised to building adaptive predictive local models for spatiotemporal data. An experiment with an interesting dataset of aftershocks which exhibits concept drifts in space and time is discussed in detail.
- Chapter 5 and 6 describe a new neuromorphic computing paradigm based on Spiking Neural Networks (SNN) called NeuCube which natively handles spatiotemporal data as well as a feasibility study of the application of NeuCube for hazardous event prediction through seismic data analysis.
- Chapter 7 concludes and summarises the thesis, discusses its findings and contributions, points out limitations of the current works, and also outlines directions for future research.

1.5 Publications

Throughout the completion of the PhD study, the following papers have been published:

- Hartono, R. N., Pears, R., Kasabov, N., & Worner, S. P. (2014, July). Extracting Temporal Knowledge from Time Series: A Case Study in Ecological Data. In *2014 International Joint Conference on Neural Networks (IJCNN)* (pp. 4237-4243). IEEE.
- Kasabov, N., Scott, N. M., Tu, E., Marks, S., Sengupta, N., Capecchi, E., Othman, M., Doborjeh M.G., Murli, N., Hartono, R., ... & Espinosa-Ramos, J. I. (2016). Evolving Spatio-temporal Data Machines Based on The NeuCube Neuromorphic Framework: Design Methodology and Selected Applications. *Neural Networks*, 78, 1-14.
- P. Bose, N. K. Kasabov, L. Bruzzone and R. N. Hartono (2016). Spiking Neural Networks for Crop Yield Estimation Based on Spatiotemporal Analysis of Image Time Series. In *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6563-6573

In addition, there were two poster papers:

- McNabb, C., Hartono, R., McIlwain, M., Anderson, V., Kasabov, N., Kydd, R., & Russell, B. (2014). Classification of People with Treatment-Resistant and Ultra-Treatment-Resistant Schizophrenia using Personalised Computer Modelling and EEG Data. *Schizophrenia Research*, 153, S197.
- Russell, B., Hartono, R., McIlwain, M., Anderson, V., Kasabov, N., Kydd, R., & McNabb, C. (2014). Using Personalised Computer Modelling to Classify People With Treatment-Resistant or Ultra-Treatment-Resistant Schizophrenia Based on Cognitive Measures. *Schizophrenia Research*, 153, S361.

And one book chapter:

- Hartono, R. N., Pears, R., Kasabov, N., & Worner, S. P. (2015). Sequential Rule Extraction from Ecological Time Series Data. In: Beresford RM, Froud KJ, Kean JM, Worner SP ed. The plant protection data tool box. (pp. 161-166). New Zealand Plant Protection Society.

Rule Discovery from Multiple Time-series Stream

2.1 Introduction

This chapter presents a generic framework and methods for mining temporal association rules from multiple time-series data along with its application to ecological data [Hartono 2014]. First, time-series inputs are transformed into symbolic representation by means of discretisation. Once discretised, itemset candidate can be generated. Rules are then generated with a modified apriori algorithm according to a specified minimum confidence and support value. Finally, rule selection criteria are further applied to acquire the best rules. As a part of this research, a method for discretisation of numerical data based on concept-drift detection was developed.

Our empirical study on the aphids dataset revealed that high confidence rules that predict an impending pest outbreak can be identified. The simple, actionable rules give growers an adequate window of time to take preventive action. The knowledge encapsulated by such rules could not be deduced by simple visualization methods due to the complex inter-relationships between the variables, thus reinforcing the need for application of rule mining methods.

This chapter is organised in the following manner:

- A review of the fundamental concepts related to the research is pre-

sented in Section 2.2, which includes the topics of Time-series discretisation and Association Rule Mining.

- Section 2.5 introduces the proposed methodology and describes the constituent algorithms and procedures in detail.
- In Section 2.6, the application of the technique on a real-world ecological dataset including the effects of the parameters and the extracted results is discussed.

2.2 Time-series Data Mining

Time-series data is ubiquitous in various scientific fields, and analysing time-series data is an active area of research [Fu 2011]. Although intensely researched from regression and prediction perspectives, relatively less research has focused on discovery of knowledge in the form of sequential associative rules. In an ecological context, knowledge of the environmental factors associated with a pest invasion assumes equal if not more importance than numerical predictions as it enables end users to make timely and informed decisions on when to put in place suitable pest control measures in a probabilistic manner.

2.3 Time-series Discretisation

In general, knowledge extraction from temporal data in the form of sequential rules from time series requires a pre-processing step to transform numerical data into symbolic form suitable for application of rule mining methods. One of the time-series in this study is especially interesting due to the highly skewed nature of species count time-series. This led to the development a

new discretisation method that uses the idea of concept-drift detection which does not assume that the underlying data follows a particular distribution.

The discretisation of time series data is a way to transform and efficiently represent sequence of numbers into symbolic format, which is easier to work with because of the bounded dimensionality. As with most problems in computer science, the suitable choice of information representation greatly affects the general performance of time series data mining algorithms. This data transformation phase has been a very important procedure in data mining simply because a lot of machine learning algorithms work only or much better with discrete and finite problem space [Dougherty 1995].

In order to give a general context about time-series discretisation, a definition is appropriate [Hartemink 2001]: a discretisation of a real-valued vector $v = (v_1, \dots, v_N)$ is an integer-valued vector $\mathbf{d} = (d_1, \dots, d_N)$ with the following properties:

1. Each element of \mathbf{d} is in the set $\{0, 1, \dots, D - 1\}$ for some (usually small) positive integer D , called the degree of the discretisation.
2. For all $1 \leq i, j \leq N$, we have $d_i \leq d_j$ if $v_i \leq v_j$.

There is an enormous wealth of existing algorithms and data structures that allow the efficient manipulations of strings from the fields of natural language processing, information retrieval and bioinformatics. With the help of an effective symbolic representation of numerical data, these methods can also work with time-series research and has promising potential for scientists to explore new ways of time-series data mining.

There are multiple approaches that one can take to discretise time-series data. In [Das 1998], a clustering technique is used to transform the time series into symbols representing the geometrical shape of the time-series. In [Wan 2007], a clustering method based on the concept of Partial K-Completeness and Interestingness is used on Hydrological data. Mörchen and

Ultsch proposed a new quality score to measure unsupervised discretisation of time series, by taking the temporal information into account and searching for persistence, argued as more suitable for knowledge discovery purposes, and offered a discretisation algorithm called Persist [Mörchen 2005].

2.3.1 Symbolic Aggregate approxIimation (SAX)

One of the most widely used discretisation methods for time series is the Symbolic Aggregate approxIimation (SAX). The SAX method uses the Gaussian distribution to discretise data into bins that contain values that occur with equal probability [Lin 2003].

SAX allows a time series of length n to be reduced to a string of arbitrary length w , ($w < n$, typically $w \ll n$). The alphabet size is also an arbitrary integer a , where $a > 2$. This alphabet size equals to the degree of discretisation, i.e. how many bins to put the numbers into. The SAX approach also uses an intermediate representation between the raw time series and the symbolic strings. It first transforms the data into the Piecewise Aggregate Approximation (PAA) representation and then symbolize the PAA representation into a discrete string.

A time series C of length n can be represented in a w -dimensional space by a vector $\bar{C} = \bar{c}_1, \dots, \bar{c}_w$. The i -th element of \bar{C} is calculated by the following equation:

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \quad (2.1)$$

This step reduces the time series from n -dimensions to w -dimensions by partitioning the data into w equal sized pieces. The mean value of every piece is calculated and a vector of these values becomes the data-reduced representation. This dimensionality reduction step is depicted in Figure 2.1.

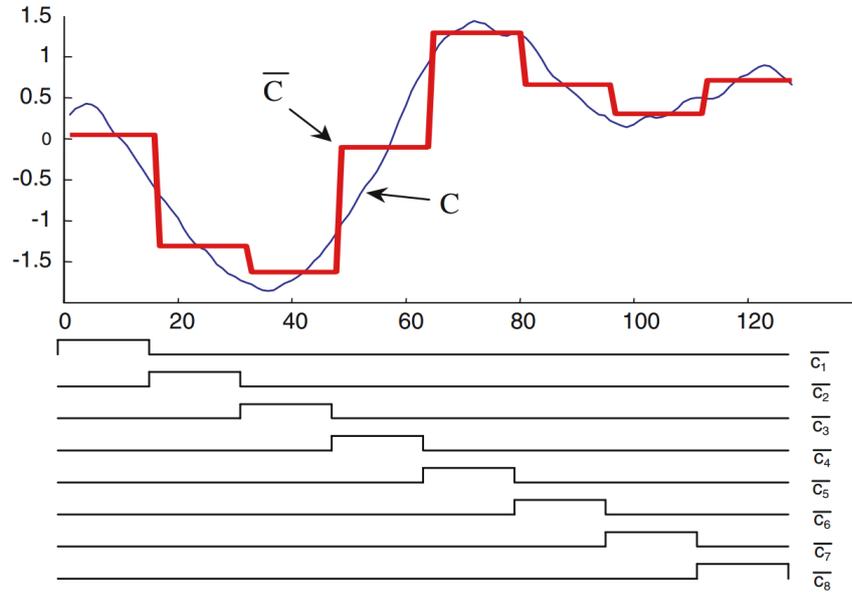


Figure 2.1: An illustration of PAA transformation (with time as X-axis) as an attempt to model a time series with a linear combination of box basis functions. In this case, a sequence of length 128 is reduced to eight dimensions [Lin 2007].

Having transformed a time series database into the PAA, further transformation is applied to obtain a discrete representation. What SAX aims to produce is symbols with equiprobability, which means that the alphabet letters are equally likely to occur. The authors of SAX assumed that normalised time series have a Gaussian distribution, based on [Larsen 1986]. If this assumption is true, then it is straightforward to determine the cut points that will produce a equal-sized areas under Gaussian curve as can be seen from Figure 2.2. However it has to be noted that this assertion has been contested by [Balzanella 2010].

These cut points are a sorted list of numbers $B = \beta_1, \dots, \beta_{a-1}$ such that the area under a $N(0, 1)$ Gaussian curve from β_i to $\beta_{i+1} = 1/a$ (β_0 and β_a are defined as $-\infty$ and ∞ , respectively). These breakpoints may be determined by looking them up in a statistical table. Once the breakpoints have been

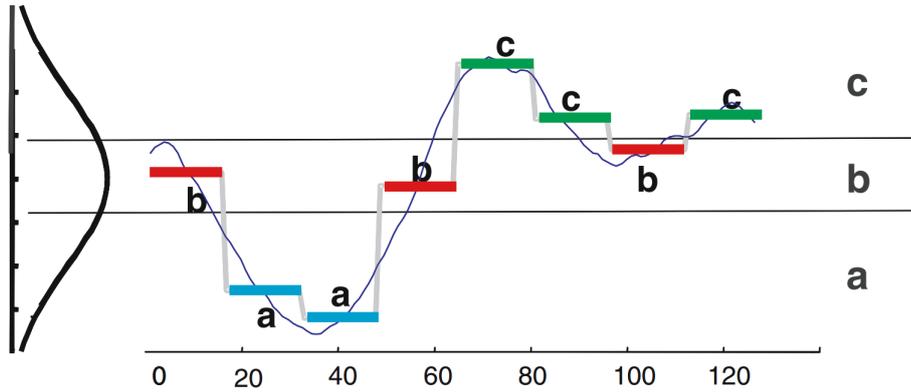


Figure 2.2: A time series is discretised by first obtaining a PAA approximation and then using predetermined breakpoints to map the PAA coefficients into SAX symbols. In the example above, with $n = 128$, $w = 8$ and $a = 3$, the time series is mapped to the word *baabccbc* [Lin 2007].

obtained the values in the PAA can be converted into a letter. For example, those that are below the smallest breakpoint (β_1) are mapped to the symbol 'a'.

2.4 Association Rule Mining

Association rule mining is a term to describe methods that aim to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. First introduced by IBM researchers in 1993 [Agrawal 1993], it has grown to be one of the most important and well researched techniques in data mining [Kotsiantis 2006].

One of the first applications of rule mining was on a large database of customer transactions trying to figure out what items that are often bought together and has seen a great acceptance in the retail industry. This has been touted as one of the greatest successes of Business Intelligence and Data Mining endeavours that has brought the approach into mainstream publications

with the story of the beer and diaper finding. The legend has it that using association rule mining algorithms, retailers in the United States have found that on Friday nights, men between 30-40 years of age who purchased diapers were most likely to also have beer in their carts. Using this previously inconceivable information, retailers rearranged the aisles and pair these items together and consequently the sales for both items increased.

2.4.1 Definition

Let $I = I_1, I_2, \dots, I_m$ be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented as a binary vector, with $t[k] = 1$ if t bought the item I_k , and $t[k] = 0$ otherwise. There is one tuple in the database for each transaction. Let X be a set of some items in I . We say that a transaction t satisfies X if for all items I_k in X , $t[k] = 1$. Association Rule is the implication in the form $X \Rightarrow I_j$, where X is a set of some items in I , and I_j is a single item in I that is not present in X . The rule $X \Rightarrow I_j$ is satisfied in the set of transactions T with the confidence factor $0 \leq c \leq 1 \iff$ at least $c\%$ of transactions in T that satisfy X also satisfy I_j . The notation $X \Rightarrow I_j | c$ can be used to specify that the rule $X \Rightarrow I_j$ has a confidence factor of c .

Given the set of transactions T , association rule mining generates all rules that satisfy these following constraints of different form:

1. *Syntactic Constraints*: These constraints involve restrictions on items that can appear in a rule. For example, one may be interested only in rules that have a specific item I_x appearing in the consequent, or rules that have a specific item I_y appearing in the antecedent.
2. *Support Constraints*: These constraints concern the number of transactions in T that support a rule. The support for a rule is defined to be

the fraction of transactions in T that satisfy the union of items in the consequent and antecedent of the rule. Support should not be confused with confidence. While confidence is a measure of the rule's strength, support corresponds to statistical significance. Besides statistical significance, another motivation for support constraints comes from the fact that people are usually interested only in rules with support above some minimum threshold for practical reasons.

It becomes obvious why these constraints are of utter importance in large dataset rule mining: without defining a set of items that we are interested in and without defining a minimum threshold of support and confidence, the system will produce copious amount of rules that are not of interest or significance, and ultimately meaningless. In this context, the problem of discovering all association rules can be decomposed into two subproblems:

1. Generate all combinations of items (itemsets) by finding all those that have fractional transaction support above a certain threshold, called *minsupport*. The support for an itemset is the number of transactions that contain the itemset. This combined with the syntactic constraints further constrain the admissible combinations. For example, if only rules involving an item I_x in the antecedent are of interest, then it is sufficient to generate only those combinations that contain I_x .
2. Use the itemsets obtained from the previous step to generate the desired rules. For a given itemset $Y = I_1, I_2, \dots, I_k, k \geq 2$, generate all rules (at the most k rules) that use items from the set Y . The antecedent of each of these rules will be a subset X of Y such that X has $k - 1$ items, and the consequent will be the item $Y - X$.

In the following sections, a framework for sequential rule extraction from multiple time series data is described that incorporates discretisation; pattern specification for rule generation, sequential rule mining, and finally rule

evaluation methods. Fundamental concepts that the proposed technique is built on will be presented first, followed by the description of the proposed methodology and its constituent components.

2.5 Proposed Framework

This section will elaborate in detail on the framework that we used to mine temporal rules. Three major steps are involved, namely discretisation of time series variables, rule extraction and finally, rule evaluation.

2.5.1 Sequential Pattern Mining

One of the very first studies on extracting rules from time-series data used a clustering method to discretise the time-series based on geometrical shape and proposed a modified Apriori algorithm to discover rules from a set of discretised time-series [Das 1998]. A general methodology for knowledge discovery from time-series which produce fuzzified rules based on an information-theoretic and connectionist approach was also proposed [Last 2001].

Aside from these studies, research in time-series rule mining has revolved mostly around improvement to certain aspects of the aforementioned techniques, such as the discretisation step, and different applications of it. Useful rules were mined from multi-attribute medical data, specifically from multiple surface electromyogram (EMG) data to analyse muscle movement behaviors with sequential apriori algorithm [Pradhan 2009]. Rule mining as a part of an integrated time-series data mining of medical therapy data as part of a hospital information system has also been explored [Abe 2005]. Temporal rule mining was also used for financial data analysis in conjunction with the usage of symbolic aggregate approximation (SAX) as the discretisation technique [Warasup 2006].

2.5.2 Problem Definition

The temporal rule extraction problem in general can be stated as the discovery of rules that associate the occurrence of an event of interest B within a given time period T of the occurrence of another event A . The events A and B are represented by items or sets of items (henceforth referred to as itemsets). We will first formally define the notion of itemsets in the context of temporal rule extraction.

Given a set of n time series variables: X_1, X_2, \dots, X_n that are considered to be predictors of another time series Y , we first obtain the discretised versions of the predictor variables as sets $D_1, D_2, \dots, D_i, \dots, D_n$ respectively, where each D_i is itself a set of symbols obtained by discretising variable X_i . An itemset I can now be defined as:

$$I \subseteq DS \quad (2.2)$$

where $DS = \bigcup_{s=1, n} D_s I$.

Thus an itemset is essentially a set of co-occurring items as in classical association rule mining, but with the added constraint that their occurrence is sequential in nature.

A temporal rule spanning a time period T is denoted by $(A \xrightarrow{T} B)$ where A, B are itemsets, $supp(A) > minsup$, $supp(B) > minsup$ represent the support of itemsets A and B respectively, while $minsup$ is a user defined minimum support threshold; confidence of the rule $c(A \xrightarrow{T} B) = \frac{supp(A, B, T)}{supp(A)} > minconf$, a user defined threshold on confidence, and T is a user defined time horizon that specifies that itemset B occurs at most T units of time after the occurrence of itemset A . This definition is an expanded version of the non-sequential rule mining described in Section 2.4.1 by the addition of a previously nonexistent or ignored temporal variable.

In the context of the aphids dataset we restrict the itemset B to strings containing the symbol that denotes high occurrence of aphid count as the

focus of the research is to discover events that lead to high levels of aphid infestation.

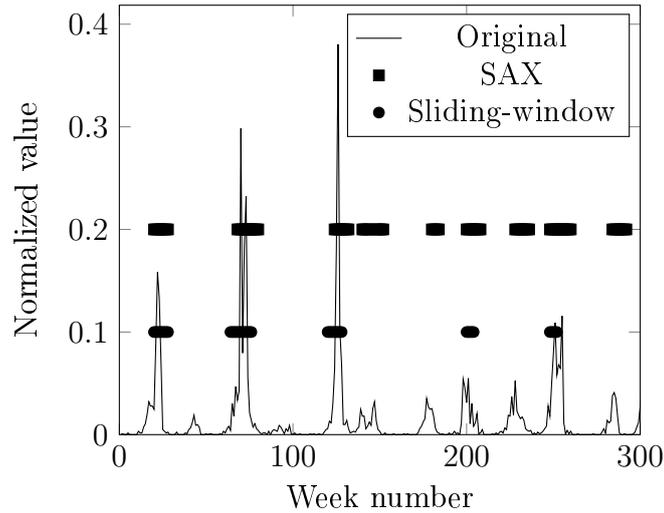
2.5.3 Adaptive-window Discretisation

Since most of sequential rule mining algorithms work on data in the form of strings of symbols, one of the most crucial steps in this framework is to find the most appropriate way to transform numerical values of the time-series into symbolic strings. Different strategies have been proposed by various researchers to suit the characteristics of the data.

In this research, SAX can be applied to all but one of the time series variables. The result with SAX on ecological species count observation series, which follows a Poisson distribution, is not satisfactory. Because SAX works on the assumption that the values follow Gaussian distribution, there were a lot of miscategorised small peaks. Figure 2.3 shows that discretising the time-series into a low-high two-symbol string with SAX will incorrectly assign many low-valued peaks to the high symbol. When faced with such a problem the usual workaround is to log-transform the numerical value to fit the Gaussian distribution, but such transformation with ecological count data for the purpose of satisfying the parametric assumption should be avoided [O'hara 2010]. This became a problem because ecological species count data is very often sparse, containing many zero values and as such a log transformation is often applied.

The proposed sliding window algorithm works by having two segments of a particular size starting from the beginning of the series which slide across the series incrementally. A model is built by using training data in the left segment, and the model is then deployed on new unseen data arriving in the right segment. If the root mean square error of the model on the test (right) segment exceeds a certain threshold, then the data element that defines the

Figure 2.3: *Discretisation result comparison. Values of 0.1 and 0.2 have been selected to signify which part of the time-series signal is discretised as a high-signal by SAX (box marks) and the proposed sliding window approach (circle marks).*



boundary between the left and right segments is considered to be a cut point. A cut point represents a transition from either a low signal state to a high signal state or vice-versa. In order to distinguish between the two cases we record the average signal value between the left and right segments. If the average of the right segment is larger, then a transition from a low state to a high state is indicated, else the transition occurs in the opposite direction, from high to low [Hartono 2014].

The comparison made in Figure 2.3 shows that the proposed algorithm is more selective in indicating which peaks are considered to represent high occurrences of aphids. Moreover, the change-detecting nature of the algorithm means that the segmentation is not made according to the absolute value in the time series, but the changes in the value which indicate concept changes. This explains why the high symbol generated by the sliding window algorithm seems to be segmenting a little bit ahead of the actual peak. Once

the right window sees the peak, the cut point which is ahead of it, is made. This behaviour is expected to be useful in detecting a pest outbreak, where the identification of when a concept change happens is more important than identifying when a peak occurs. Although there is no rigid objective criterion as to which discretisation is better, the effectiveness of the discretisation can be indirectly evaluated by looking at the rules produced in the consequent steps.

Algorithm 2.1 Sliding-window model based discretisation

```

INPUT:  $D$ , block-size,  $\varepsilon$  (threshold for change detection)
 $a \leftarrow 1$ 
 $i \leftarrow 1$ 
while not finished scanning the time-series do
   $b \leftarrow a + \text{block-size}$ 
  model  $\leftarrow$  build-model( $D[a:b]$ )
  error  $\leftarrow$  test-model( $D[b:b + \text{block-size}]$ )
  if error  $> \varepsilon \wedge \text{avg}(D[a:b]) < \text{avg}(D[b:b + \text{block-size}])$  then
    highpoint[ $i$ ]  $\leftarrow b$ 
     $i \leftarrow i + 1$ 
  end if
   $a \leftarrow a + 1$ 
end while
return highpoint

```

The pseudo-code for this sliding window based discretisation algorithm is laid out in Algorithm 5.2. The algorithm is intuitive and has been explored before as a way to segment time-series along with variations in the selection of the regression model (linear vs non-linear) to fit the data and ways to measure the error [Keogh 2004]. This generic algorithm can be implemented by utilizing any type of learning scheme. In this research, we have chosen

to use a multiple linear regression model in conjunction with a lagged data, 1-step ahead prediction training/testing regime. Regression was a natural choice as the underlying data is numeric in nature and the linear variant is efficient while being reasonably robust in terms of predictive accuracy. In this algorithm, one important parameter that has to be tuned is the ε , which will signify the sensitivity of the change detection. This parameter can be optimized by incrementing the value until there is no change in the discretisation result, which suggests that the detected changes are significant enough.

2.5.4 Itemset candidate generation

Given a sequence $D(S) = (a_1, a_2, \dots, a_n)$, the itemset candidate is:

$$I = \left\{ i \mid i \subseteq D(S) \wedge |i| < T \right\} \quad (2.3)$$

where T is a predefined maximum time period. This step will essentially generate a set of all the subsequences possible from a sequence where the length of every subsequence is not larger than T .

2.5.5 Rule Extraction

As introduced in section 2.5.2, we used an association rule mining algorithm which produces rules in the form:

if A occurs, then B occurs within time T.

where A,B are itemsets. Instead of representing symbols like in [Das 1998], the A and B represent subsequences. If the above rule is denoted as $A \xrightarrow{T} B$, then we calculate the confidence of each rule as:

$$c(A \xrightarrow{T} B) = \frac{\text{supp}(A, B, T)}{\text{supp}(A)}, \quad (2.4)$$

where

$$\text{supp}(A, B, T) = \left| \left\{ i \mid a_i = A \wedge B \in \{a_{i+1}, \dots, a_{i+T-1}\} \right\} \right| \quad (2.5)$$

and a_i is the symbol that occurs in the i^{th} time step.

Equation (2.5) represents the number of occurrences of A that are followed by B within a given time period T . The pseudocode of a generic implementation of this technique is presented in Algorithm 2.2 and can be optimized or modified in various ways to suit any need.

Algorithm 2.2 Rule-mining algorithm

INPUT: $D_{1,\dots,n}$, minsupp, minconf, T

$I \leftarrow \text{generate-itemset}(D)$

counter $\leftarrow 0$

for $i=1$ to n **do**

for all a in I_i **do**

for all b in I_i that occurs in T time steps after a **do**

 confidence $\leftarrow \text{support}(a,b,T) / \text{support}(a)$

if confidence $>$ minconf \wedge support(a) $>$ minsupp **then**

 rules[counter] $\leftarrow a \xrightarrow{T} b$

 counter \leftarrow counter + 1

end if

end for

end for

end for

return rules

2.5.6 Rule Selection criteria

One issue with association rule mining in general and sequential rule mining in particular is that a large amount of rules may be generated, most of which may be trivial and/or uninteresting. Thus, the selection of those rules that are significant and interesting is a challenging task. Adopting the idea of support and confidence from associative rule mining could be useful. Confidence and support are the two most commonly used metrics for measuring rule quality, but a number of researchers have devised other measurements of interestingness for association rules such as the J-measure and Mutual Information, and have subjected these measures to validation and testing [Lallich 2007][Tan 2002]. The J-measure [Smyth 1992] could be used here and is defined as:

$$j(A \xrightarrow{T} B) = p(A) * \left(p(B|A) \cdot \log\left(\frac{p(B|A)}{p(B)}\right) + (1 - p(B|A)) \cdot \log\left(\frac{1 - p(B|A)}{1 - p(B)}\right) \right) \quad (2.6)$$

In this context, $p(A)$ is the probability of pattern A occurring among all itemsets of the same length generated from the sequence, while $p(B|A)$ is the probability of pattern B occurring within T time period after the pattern A . The left-hand term gives weight to the frequency of the pattern A , and the right-hand term is the cross-entropy or the information gain. Cross-entropy is well known as a goodness-of-fit measure between two distributions. In this sense, the J-measure is a measure of how dissimilar the *a priori* and *a posteriori* beliefs are about a rule. Since there are many rules with high confidence with low support, and vice versa, J-measure is useful because it combines and gives a balanced measurement between the support and confidence. Practically, it can be used as a sound method to create an additional criterion to rank rules.

In this research T is considered as an additional constraint measuring the usefulness of the rules generated. There is obviously little benefit in mining rules with high confidence and support but which span over a long period of time. That is the real world equivalent of saying a plane crash will happen within the next decade. The statement carries a very high level of confidence, but is not particularly useful because of the excessive length of the prediction period.

2.5.7 Rule Format Extension

We extend the rule format and algorithm to accommodate multiple antecedents from different time-series in the form:

if A_1 and A_2 and ... and A_h occur within V units of time, then B occurs within T time units.

The above rule can be denoted by $A_1 \wedge \dots \wedge A_h \xrightarrow{V,T} B$. This opens up the possibility of mining from multiple time-series and extracting interactions between the variables involved.

2.6 Experimental Study

The methods described in this chapter was empirically tested on a dataset which comprises of aphid trap catches recorded by Crop & Food Research, Lincoln, Canterbury, and weekly weather data consisting of 12 weather variables recorded at the Canterbury Agricultural Research Center, Lincoln, New Zealand, spanning over 19 years as described in [Worner 2002].

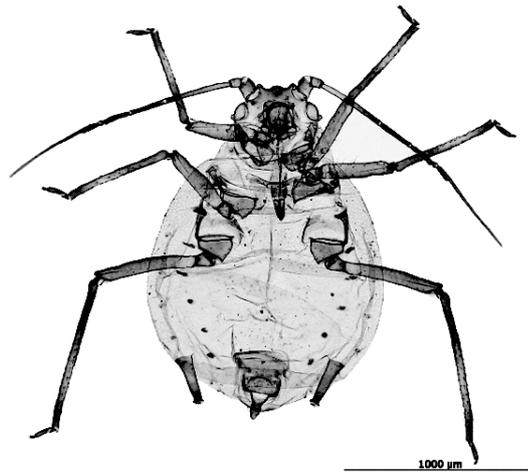


Figure 2.4: A macro habitus photograph of a wingless adult *Rhopalosiphum padi*, one of the species of aphids that are collected in the dataset. Photo: Brendan Wray, AphID, USDA APHIS ITP, Bugwood.org

2.6.1 The Dataset

The dataset used in this experiment comes from nineteen years of aphid trap catches recorded by Crop & Food Research, Lincoln, Canterbury, and weekly weather data consisting of 12 variables recorded at the Canterbury Agricultural Research Centre, Lincoln. The data collection for all variables started from week 27 of 1981 and extended to week 26 of 2000.

The following independent variables were available: cumulative weekly rainfall (mm); maximum, minimum and mean air temperature ($^{\circ}\text{C}$); grass minimum temperature ($^{\circ}\text{C}$); solar radiation (MJ/m^2); wind run (km/day); vapour pressure (hPa); Penman potential evaporation (mm); potential deficit (accumulated excess of Penman over rainfall); average weekly degree days (above a threshold of 0°C); and soil temperature at 100 cm below ground ($^{\circ}\text{C}$).

Weather conditions influence aphid population dynamics either directly, by affecting development and survival, or indirectly by affecting natural ene-

mies or the aphid host plant. Several studies have shown aphid migrations to be correlated with the weather conditions prevailing prior to migratory flights and have used these relationships to predict them [Worner 1995]. Several aphid species are vectors of barley yellow dwarf virus (BYDV), an important cause of yield loss in wheat crops in New Zealand. In Canterbury, the most dangerous aphid vector is *Rhopalosiphum padi*. Any prior information that can indicate the imminence of an outbreak of this species would benefit growers by better informing their pest management decisions.

2.6.2 Experimental Configuration

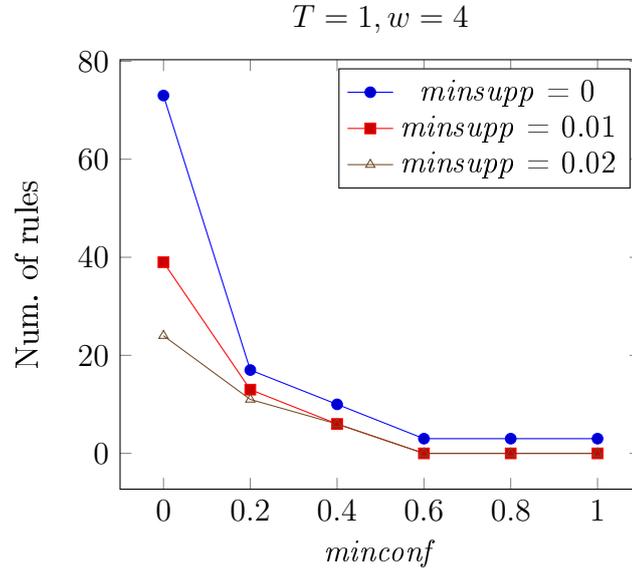
In this research, the number of variables used are limited to: Cumulative weekly rainfall, wind run, average air temperature, potential deficit, Penman potential evaporation, and solar radiation.

The experiment focuses on whether the proposed framework and methods could discover interesting rules from the multiple time series that can act as a precursor of aphid infestation, and also to see the effect and the sensitivity of parameter values on the results obtained. Since the focus of interest is prediction of high incidence of aphid infestation the rules extracted are restricted to those that feature high aphid count occurrence on rule consequents. We also conduct a sensitivity analysis on key parameters such as window size (w), prediction time horizon (T), minimum support threshold ($minsupp$) and minimum confidence threshold ($minconf$).

All experiments were run on a Core i7 processor configuration running under Windows 7 with MATLAB as the main programming tool.

2.6.3 Effects of Minimum Support Threshold

Figure 2.5 shows the effect on the number of rules generated as the $minsupp$ threshold is increased. As expected, when $minsupp$ increases from 0 a steady

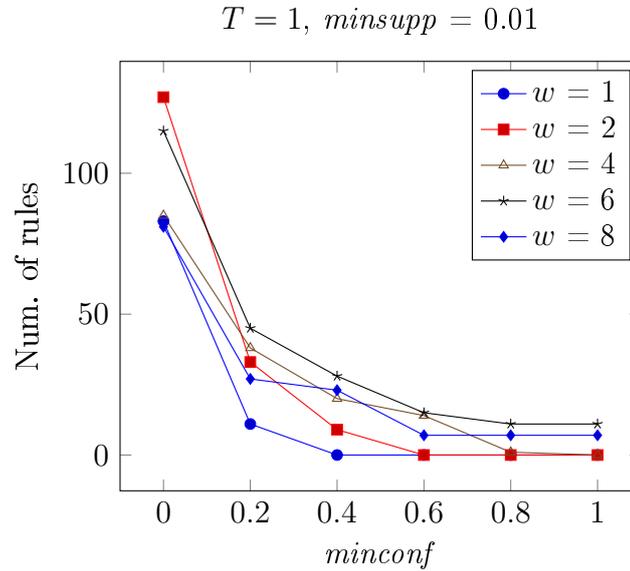
Figure 2.5: Number of rules generated with different *minconf* and *minsupp*

decrease in the number of high confidence rules (with confidence ≥ 0.8) is observed. Interestingly, with no constraint on support we observe that a substantial number of rules (numbering 10) with confidence of 1 predict high occurrence of aphid infestation within a week of the triggering event firing on the rule antecedent. The time step resolution used is weekly, as the aphids count data was collected on a weekly basis. The same trend is observed for higher *minsupp* threshold values, although the number of high confidence rules generated reduces by a factor of 2 or more.

2.6.4 Effect of Window Size

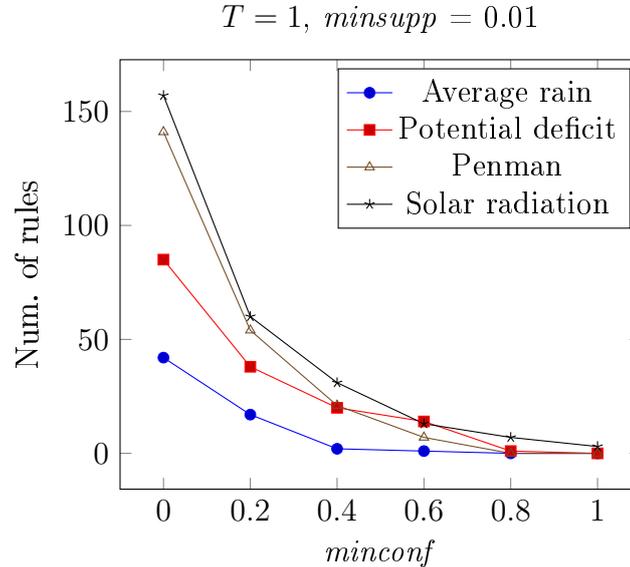
Figure 2.6 shows that as the window size w increases from 1 a higher number of high confidence rules can be obtained. Basically, the reason is that smaller window sizes are vulnerable to the effects of noise. Small window sizes capture smaller transitions between states when compared to larger window sizes, thus effectively identifying smaller peaks. The problem with smaller peaks is that they are associated with random behaviour, thus trigger

Figure 2.6: Number of rules generated with different window size of discretisation



conditions on rule antecedents are rendered ineffective, giving rise to low confidence rules. As the window size increases from 2 the noise level decreases and the number of high confidence rules increases.

However, increasing the window size from 6 to 8 results in a reduction in the number of high confidence rules. Beyond a certain threshold on window size, dependent on the nature of the underlying dataset, some high valued peaks will not be detected and thus some rules, including some with high confidence rules will not be generated. Moreover, too large a window size is undesirable because the period of time that T represents corresponds to the window size. Having a wide window size for T means that the rules produced have a longer window of prediction, rendering them less useful. We decided to use $w = 4$ as the window size, which is what we thought as a practical trade-off between the number of rules generated and the resolution of prediction.

Figure 2.7: Number of rules generated by different antecedent variables

2.6.5 Effects of Rule Antecedent Variable

A visualization of the difference in the ability of the variables to produce rules with varying levels of confidence is shown in Fig2.7. The number of high confidence rules which a variable can produce is an indicator of its relative importance in influencing a high aphid count outcome. In this context, we can infer that the cumulative rainfall is relatively less important than the other variables, and on the other hand, the Penman potential evaporation seems to be a very strong feature, being able to produce rules with very high confidence.

Table 2.1 shows some of the rules that are discovered using the algorithm. These rules are visualized in Figure 2.8, 2.9, and 2.10, which show the occurrence of high aphid count following the triggering events which are captured by the antecedent of the rule. The string symbols used as the antecedent and the consequence of the rules are from the discretised time series with 4 levels of intensity, the character a and d for the lowest and the highest values respectively. Thus Figure 2.8 shows that 4 consecutive occurrences of high

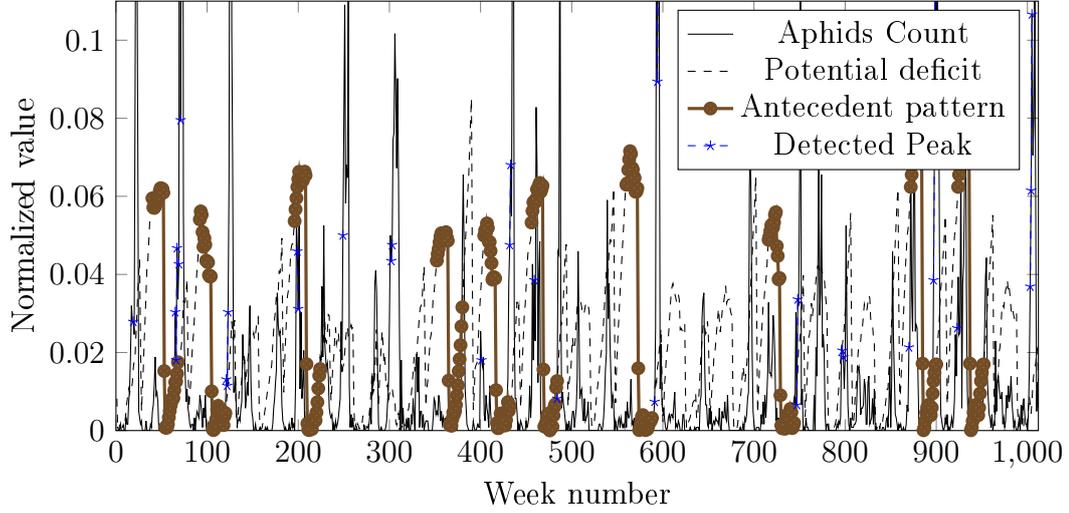
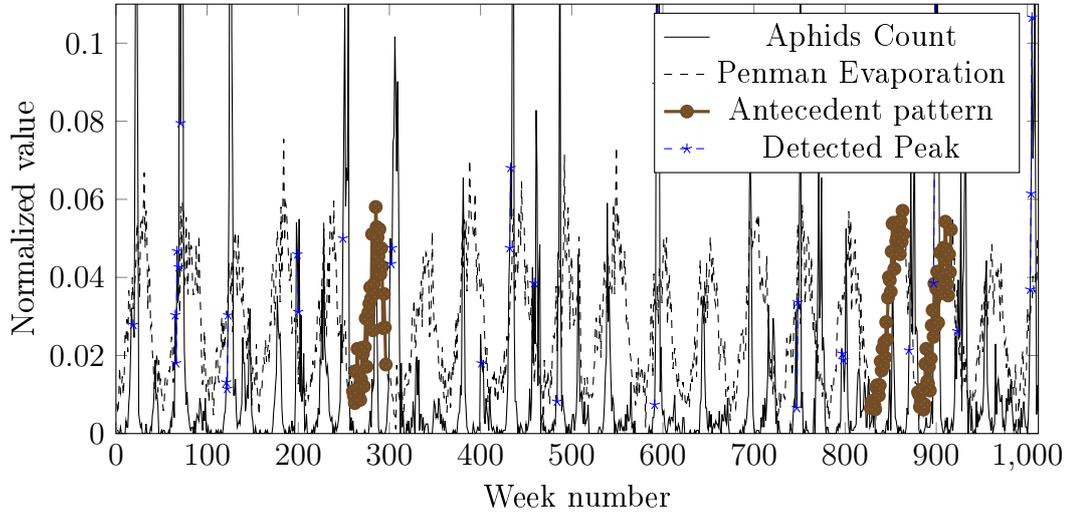
Figure 2.8: Rule visualization: $ddddaaa \rightarrow b$ 

Table 2.1: Some rules produced by the method

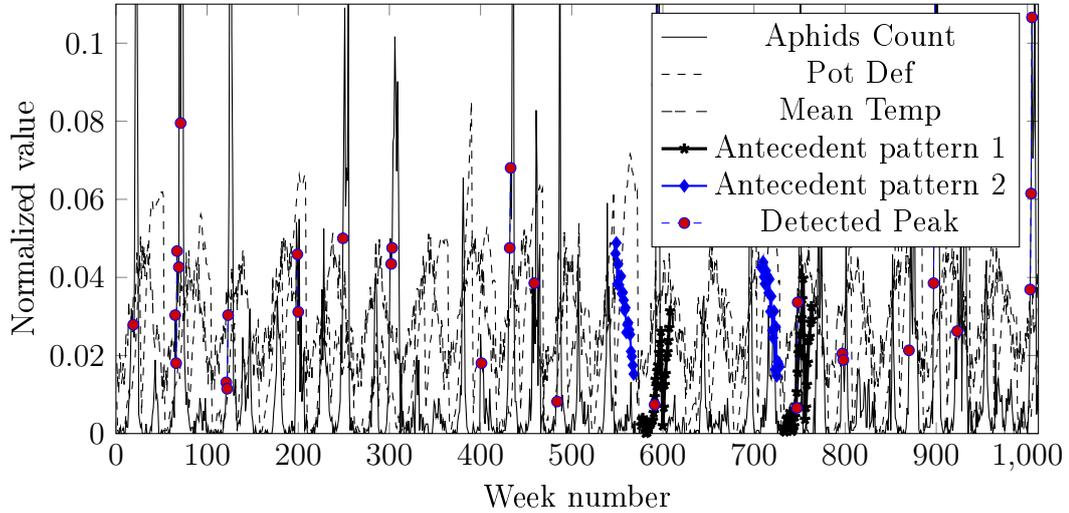
w	Antecedent	Rule	Supp. (%)	Conf. (%)	J-Measure	Fig.
4	Potential deficit	$ddddaaa \xrightarrow{1} b$	8.86	70	0.0036	2.8
4	Penman evaporation	$aaaabcddd \xrightarrow{2} b$	3.52	75	0.0024	2.9
4	Potential deficit & Mean temperature	$aaaaabbb, dddcb \xrightarrow{4,1} b$	0.8	100	0.0009	2.10

potential deficit values (denoted by symbol "d") followed by 3 consecutive low level occurrences of potential deficit (denoted by symbol "a") triggers a high aphid count. This can be interpreted as the occurrence of a rapid drop

Figure 2.9: Rule visualization: $aaaabcddd \rightarrow b$ 

after a period of high values in the antecedent. The high confidence of the rule is evident from the visualization that shows that the antecedent pattern is followed almost always by a peak in the aphid count value. The support of the rule is also evident from the number of co-occurrences of the antecedent and consequent patterns - i.e. the number of times that the string $dddaaab$ occurs in the data.

Likewise, Figure 2.9, and 2.10 visually show the confidence of rules: $aaaabcddd \xrightarrow{2} b$ and $aaaaabbb, dddcb \xrightarrow{4,1} b$ respectively. The latter rule clearly shows the effect of having multiple variables in the rule antecedent. The inclusion of Mean temperature with Potential deficit in the rule antecedent results in an increase in confidence of 30% over the use of Potential deficit alone. However, the trade-off caused by the inclusion of the additional variable has caused the support and J-measure metrics to decrease substantially.

Figure 2.10: Rule visualization: *aaaaabb, dddcb to b*

2.6.6 Conclusions

Overall, it is evident that the rules generated in this experiment are useful in identifying high levels of aphid infestation. For example, rules 1 and 2 in Table 2.1 identify trends in Potential deficit and Penman evaporation variables respectively that lead to high aphid count with reasonably high levels of confidence (70% and 75% respectively), while giving growers an adequate time periods (1 week and 2 weeks, respectively) to implement suitable pest control measures. This usefulness also comes from the interpretability of the rules that can be expressed in a form that humans can understand with relative ease. For instance, rule 1 (shorthand form $ddddaaa \xrightarrow{1} b \mid 70$) can be reformulated in natural language form as:

If very high values of Potential deficit occur for 4 weeks and then followed by 3 weeks of very low value, then 70% of the time, a surge in aphids count will occur within 1 week.

These temporal rules are not only human-readable and actionable, but also

descriptive of the relationships between variables that also take time information into account. These rules can potentially discover a previously unknown direct or indirect causality or correlation between itemsets and can also be verified and cross-checked with the existing body of literature in the problem domain. In this case with the Potential deficit, it has been noted by several ecological studies that humidity and evaporation play an important role in the reproductive cycle and migration of aphids [Lowe 1966, Dill 1990, Hand 1983].

2.7 Conclusions

In this chapter, a generic framework for mining temporal rules from multiple time-series has been described and a case study on an ecological dataset has been demonstrated. It works by firstly transforming the time-series data into symbolic representations. This is achieved by using discretisation algorithms. A way to do binary discretisation was also described, which uses simple sliding windows. The next step is itemset candidate generation and rule extraction by using a modified apriori algorithm which was made to consider temporal dimension and are able to work with multiple time series. Finally, several rule selection criteria are further applied to obtain the best rules.

A case study with multiple time-series data was also presented and discussed. We used a real-world ecological data that tracks the weekly count of aphids captured in suction traps along with the weather variables for almost 20 years. The methods described have been shown to be able to extract some useful temporal rules and have the potential to be applied in many other fields in which such rules could be used to improve the ability of humans to predict the likelihood of an incident happening based on currently available observations.

Adaptive Local Models for Spatiotemporal Data Analysis

3.1 Introduction

This chapter introduces a method to build local models for spatiotemporal data stream analysis. As opposed to building a single global model to learn all the data in the input space, multiple models can be employed to handle the data in a localised manner. To construct these local spaces of concepts, we treat the problem as an exercise of detecting concept drifts in space and time. By combining the locality-preserving behaviour of the Hilbert curve and the versatility of concept-drift detection algorithms, the system can continuously and indefinitely adapt to concept drifts and unseen changes in incoming data stream.

The proposed system was tested on a dataset of the occurrence of aftershocks for years around the city of Christchurch after the great 2010 Darfield earthquake, discussed in detail in Chapter 4. It exhibits superior performance and accuracy over a global model approach, and additional visualisations can be extracted from the resulting localised regions.

The following sections of this chapter are structured as follows:

- The theoretical foundations of this research are discussed in Sections 3.2, 3.4, and 3.3. The rationale for local models and the use of Hilbert curve along with concept-drift detectors is explained.

- Section 3.5 introduces the proposed methodology, which traverses the data over the spatial dimension with Hilbert curve and feed it to a concept-drift detector algorithm to determine regions belonging to the same concepts. Some performance measures for spatiotemporal data are also discussed.

3.2 Local Models

A fundamental concern that one has to take into account in dealing with data that is referenced in space is with the differences between places. Traditionally, global models have been employed in quantitative analyses of such data with the implicit assumption that there is no variability of data characteristics as a function of space. Global methods make use of all available data, whereas local methods are often defined as those that make use of some subset of the data. Therefore, it may be the case that a global model does not represent well the variations that exist at any individual location [Lloyd 2010].

The main principle behind the motivation to using local methods is the concept of spatial dependence. That is, objects that are near each other in space tend to be more similar than to objects which are farther apart [Tobler 1970]. This principle was called the Tobler's First Law of Geography. The term spatial autocorrelation refers to the correlation of a variable with itself and where neighbouring values tend to be similar this is termed positive spatial autocorrelation. Where neighbouring values tend to be dissimilar this is termed negative spatial autocorrelation. Figure 3.2 illustrates the visual difference between these different cases.

In the field of statistics, the problem of spatial nonstationarity has been a subject of great interest. Early research efforts on spatiotemporal forecasting focused on adapting existing statistical regression models from the fields

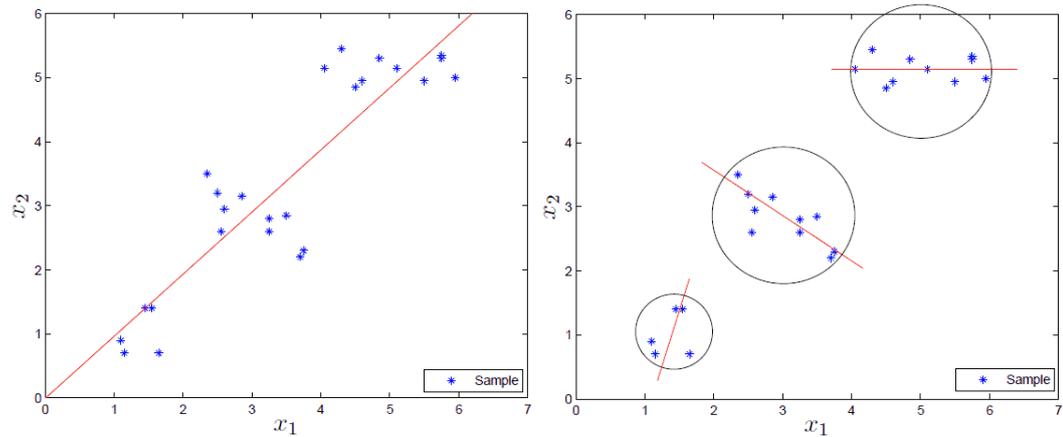


Figure 3.1: *Illustration of the difference between global (left) and local modelling (right) in trying to fit data in the input space. The local linear regression models are built based on clusters of sample data set in a 2-dimensional space.*

of time series analysis, spatial analysis and econometrics to deal with spatiotemporal data. Such models are typically geared towards teasing scarce information from homogenous datasets and have been overwhelmed by the increasing volume and diversity of spatiotemporal data that is now being collected. Some of these models also need parametric assumptions of the data distribution. Researchers and practitioners are increasingly turning towards less conventional techniques, often drawing inspiration from the fields of ma-

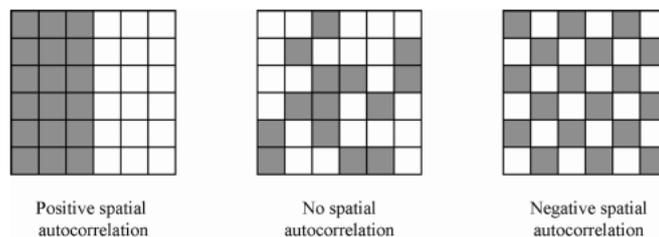


Figure 3.2: *Spatial data may demonstrate a pattern of positive spatial autocorrelation (left), negative spatial autocorrelation (right), or a pattern that is not spatially autocorrelated (center). [Radil 2011]*

chine learning and data mining, that are better equipped to deal with the heterogeneous, nonlinear and multi-scale properties of large scale spatiotemporal datasets.

It has been demonstrated that machine learning algorithms can improve their performance significantly if a specific adaptation to the given problem is employed. Vapnik argued that when the data is clearly non-evenly distributed, one can significantly improve the prediction result by building multiple local models instead of one global model [Bottou 1992]. Vapnik later introduces the concept of transductive reasoning or transductive inference and defined it as a method that is used to estimate the value of a potential model only for a single point of space by utilising additional information related to that vector [Gammerman 1998]. In [Widiputra 2011], multiple time-series prediction can be greatly improved by identifying local models based on the trends found by clustering. Local models for time-series prediction problem using concept-drift detection has also been explored in [Barddal 2015].

The principle of transductive reasoning can be applied to enhance the capability of Support Vector-based learning algorithms to deal with the problem of spatial dependence [Gilardi 2000]. It works by building one model for each point to be estimated, taking into account only a subset of the training points. This subset is chosen on the basis of the Euclidean distance between the testing point and the training point in the input space. For each testing point, a new model is learned using only training points lying inside a user defined radius which centre is the current testing point. Besides the computational power needed to build a model for each point, deciding this radius is not a trivial problem. It needs *a priori* knowledge about the dataset. If the selected radius is too small, the data points might not be enough to train the model. If the radius is too big, it will impact the training time.

In the following sections, concepts that are applied and became part of the

methodology are briefly laid out to explain the underlying theoretical considerations of employing them in tackling the problem of spatial dependence. In this research, we treat the nonstationarity problem as a case of handling concept drifts in space and time by answering the question on how to identify these local regions of homogeneous data points. This highly-automated method allows for minimal human involvement. A key component of this methodology is the application of the Hilbert curve to transform a signal in two dimensions to a single dimension in order to enable a concept drift detector that normally operates in one dimension to be used in a spatiotemporal context.

3.3 Concept-drift

The nonstationarity of data is a common challenge when mining data streams, where the concept (i.e., the underlying distribution or statistical properties) of incoming data unpredictably drifts over time. A typical example is weather prediction rules that may vary radically between seasons. Another example is the patterns of customer buying preferences that may change with time, depending on the current day of the week, availability of alternatives, inflation rate, etc. Often the cause of change is a hidden context, not known a priori, making the learning task more complicated [Widmer 1996]. An effective learner should be able to track such changes and to quickly adapt to them. Over the last decade, research related to learning with concept drift has been increasingly growing and many drift-aware adaptive learning algorithms have been developed [Gama 2014]

In a supervised learning context, there are two general approaches to concept detection. One way is to have a multivariate test keeping track of incoming samples and see whether a change in statistical properties has taken place. Another way is to feed the classification error to a change detector,

which will detect if a statistically significant shift has occurred in the error rate. A change in the error rate indicates the arrival of a new concept which has not been learned by the classifier. This latter approach is used in this thesis.

3.3.1 Definition

In machine learning the problem of supervised learning is formally defined as follows. We aim to predict a target variable $y \in \mathbb{R}$ in regression tasks (or y categorical in classification tasks) given a set of input features $X \in \mathbb{R}$. An example is one pair of (X, y) . During training, both X and y are known and used for model building. The predictive model is then applied on new examples where X is known, but y is not known.

A classification can be described by the prior probabilities of the classes $p(y)$ and the class conditional probability density functions $p(X|y)$ for all classes $y = 1, \dots, c$, where c is the number of classes. In regression, the target value takes continuous values. The classification decision is made according to the posterior probabilities of the classes, which for class y can be represented as:

$$p(y|X) = \frac{p(y)p(X|y)}{p(X)} \tag{3.1}$$

where $p(X) = \sum_{y=1}^c p(y)p(X|y)$. In dynamically changing environment, the underlying distribution of data can change over time. Formally concept drift between time point t_0 and time point t_1 can be defined as:

$$\exists X : p_{t_0}(X, y) \neq p_{t_1}(X, y) \tag{3.2}$$

where p_{t_0} denotes the joint distribution at time t_0 between the set of input variables X and the target variable y . Changes in the components of this relation characterises changes in data [Gao 2007]:

- Prior probabilities $p(y)$ may change,
- Class conditional probabilities $p(X|y)$ may change.
- Posterior probabilities $p(y|X)$ may change and thus affect the prediction performance.

3.3.2 Detection

Dealing with data whose nature changes over time (as defined in Section 3.3.1) is one of the core problems in data mining and machine learning. To mine or learn such data, one of the necessary tasks is to detect when such changes occurs.

Let $S_1 = (x_1, c_2, \dots, x_m)$ and $S_2 = (x_{m+1}, \dots, x_n)$ with $0 < m < n$ representing two samples of instances from a stream with population means μ_1 and μ_2 respectively. Change detection problem can be described as testing the null hypothesis $H_0 : \mu_1 = \mu_2$ against the alternate hypothesis $H_1 : \mu_1 \neq \mu_2$. If H_0 is rejected, concept drift is taken to have occurred.

The standard tools for drift detection are methods from statistical decision theory. These methods usually compute a statistic from the available data, which is sensitive to changes between the two sets of examples [Dries 2009]. The measured values of the statistic are then compared to the expected value under the null hypothesis that both samples are from the same distribution.

Concept drift detectors are used in this framework to analyse the data stream and label similar or homogeneous data points together. Several methods have already been proposed, one of which is the Drift Detection Method (DDM) [Gama 2004]. It uses a base learner to classify incoming instances and the classification result is used to compute the online error rate of the base learner. If the base learner correctly classifies the actual instance, the

error rate decreases. DDM considers that, when the concept changes, the base learner will incorrectly classify the arriving instances that are created based on a different data distribution. Thus, if the error-rate increases, it is an indication of a concept drift. The average error rate p_i and the standard deviation s_i are computed and then used to determine between *Warning* and *Change* states when $p_i + s_i$ is over a threshold.

The Early Drift Detection Method (EDDM) is similar to DDM but instead of using the error rate, it uses the distance-error-rate of the base learner to identify whether a drift has occurred [Baena-Garcia 2006]. EWMA for Concept Drift Detection (ECDD) is a proposal for a drift detection method based on Exponentially Weighted Moving Average (EWMA), used for identifying an increase in the mean of a sequence of random variables [Ross 2012].

Different test statistics have been proposed to deal with change detection problem. The Kolmogorov-Smirnoff (KS) test computed through the use of KS-Tree was used to determine whether the samples came from the same distribution [Kifer 2004]. A semi-parametric log-likelihood change detector was proposed based on Kullback-Leibler statistics [Kuncheva 2013]. Another based on Martingale tests has also been proposed [Ho 2005].

Adjusting the windowing or sampling scheme and modifying test statistics are among the ways to improve the accuracy and performance of change detection algorithm. For example, the algorithm SeqDrift1 [Sakthithasan 2013] was able to cut execution time dramatically and improve the false positive rate compared to ADWIN by using sequential windowing and the use of Bernstein bound. Additional improvement in detection delay can be achieved in the next iteration of the algorithm, SeqDrift2, by using reservoir sampling and a different derivation of the Bernstein bound [Pears 2014].

Theoretically, various kinds of existing change detectors are usable with this research. For the purpose of this research, we chose ADWIN as it does not need too many parameter tuning and provides a rigorous guarantees on

false positive and negative rates.

3.3.3 The ADWIN Algorithm

Short for ADaptive-WINdow, ADWIN works on the principle of statistical hypothesis testing [Bifet 2007]. It works by using a window consisting of all instances that have arrived since the last detected change point. On the arrival of each new instance, the current window is split into two sub-windows, left and right. The sample means of the data in the two sub-windows are compared under a null hypothesis H_0 that the means across the sub-windows are not significantly different from each other. If H_0 is rejected, concept drift is taken to have occurred and ADWIN shrinks the window to only include instances in the right sub-window, thus removing instances in the left window representing the old concept.

Algorithm 3.1 Adaptive Windowing Algorithm

INPUT: Stream data X of indefinite length t

Initialise Window W

for each $t > 0$ **do**

$W \leftarrow W \cup \{x_t\}$

repeat

Drop elements from the tail of W

until $|\hat{\mu}_{w_0} - \hat{\mu}_{w_1}| \geq \varepsilon_{cut}$ holds for every split of W into $W = W_0 \cdot W_1$

output $\hat{\mu}_w$

end for

The algorithm is presented in Algorithm 3.1. It keeps a sliding window W with the most recently read x_i . Let n denote the length of W , $\hat{\mu}_W$ the average of the elements in W , and μ_W the average of μ_t for $t \in W$. Whenever two large enough subwindows of W exhibit distinct enough averages, one can conclude that the corresponding expected values are different, and the older

portion of the window is dropped. The value of ε_{cut} for a partition $W_0 \cdot W_1$ of W is computed as follows: Let n_0 and n_1 be the lengths of W_0 and W_1 and n be the length of W , so $n = n_0 + n_1$. Let $\hat{\mu}_{W_0}$ and $\hat{\mu}_{W_1}$ be the averages of the values in W_0 and W_1 , and μ_{W_0} and μ_{W_1} their expected values:

$$\varepsilon_{cut} = \sqrt{\frac{1}{2m} \cdot \ln \frac{4}{\delta'}} \tag{3.3}$$

where $m = \frac{1}{1/n_0 + 1/n_1}$ and $\delta' = \frac{\delta}{n}$.

The statistical test simply checks whether the observed average in both subwindows differs more than the threshold ε_{cut} . ADWIN’s only user-given parameter is a confidence bound δ . We use all the default values set in an implementation of ADWIN found in the open-source stream data mining tool MOA.

3.4 The Hilbert Curve

First described by the German mathematician David Hilbert, The Hilbert curve is a continuous fractal curve which can be represented as a Lindenmayer system whose limit is a plane-filling function [Hilbert 1891]. It is a space-filling curve, meaning that the ranges pass through every point in a 2-dimensional unit square¹ and never crosses itself. It was introduced as a variant of the Peano curves, which was the first space-filling curve ever published [Peano 1890].

3.4.1 Construction

According to Jordan’s precise notion of continuous curves, a space-filling curve is a continuous mapping of the closed unit interval $I = [0; 1]$ into

¹Although the space-filling curves are discussed mostly in 2-dimensional terms in this thesis, most of them can also be generalised into n -dimensional unit hypercube.

the closed unit square $S = [0; 1]^2$ [Simmons 1963]. If the interval I can be mapped continuously onto the square S , then after partitioning I into four congruent subintervals and S into four congruent subsquares, each subinterval can be mapped continuously onto one of the subsquares. If this is carried on ad infinitum, I and S are partitioned into 2^{2n} congruent replicas for $n = \{1, 2, 3, \dots\}$. Alternatively, the construction of Hilbert curve can also be described geometrically. The curve is initially defined on a 2×2 lattice with a \square shape. Given an order k curve defined on a $2^k \times 2^k$ lattice we define the curve on a $2^{k+1} \times 2^{k+1}$ lattice as follows:

1. Place a copy of the curve, rotated 90° counter clockwise, in the lower right cell
2. Place a copy of the curve, rotated 90° clockwise, in the lower left cell.
3. Place a copy of the curve in each of the upper cells.
4. Connect these four disjoint curves in the obvious manner.

It was demonstrated that the subsquares can be arranged so that the inclusion relationships are preserved, that is, if a square corresponds to an interval, then its subsquares correspond to the subintervals of that interval. Figure 3.3 illustrates how Hilbert curves are expressed for the first five iterations.

An analytical generalization of Hilbert curves for higher dimensional space was given in [Butz 1969]. A method to generate 3-dimensional Hilbert curve (Figure 3.4) was described in [Sagan 1993], and further into 4-dimensions in [Haverkort 2011]. Moreover, an efficient implementation of 2-dimensional non-square quadrilateral Euclidean space is also possible [Hamilton 2008], which is of interest in this research.

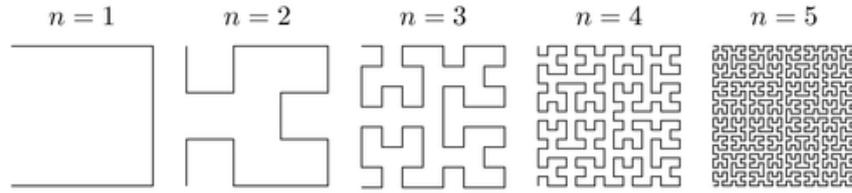


Figure 3.3: *Hilbert curves with various iteration order number n .*

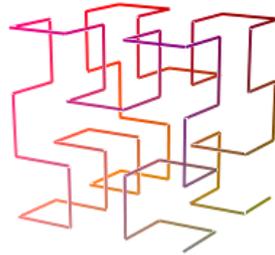


Figure 3.4: *Projection of a 3-dimensional Hilbert curve*

3.4.2 Locality preservation

Both the true Hilbert curve and its discrete approximations are useful because they provide a method to map between 1D and 2D space that preserves locality measurably well [Moon 2001]. Because of this property, Hilbert curve has found many use in computer science. For example, it has been used to compress and accelerate R-tree indexes, which can also be used to help compress data warehouses. [Kamel 1993]. In the fields of image processing and computer vision, the curve has been utilised to get better results in dithering [Zhang 1998] and compression [Biswas 2000, Liang 2008].

The locality-preserving property of space-filling curves in general and Hilbert curve in particular has been very well studied and attested by multiple analyses in different applications and purposes [Jagadish 1990, Abel 1990, Moon 2001]. As the linear ordering used in this research to transform the data, this property is deemed very important to make sure that the 2-dimensional data is presented as 1-dimensional with as little spatial infor-

mation loss as possible.

3.5 Proposed Methodology

This section presents a methodology to build predictive local models that are adaptive to changing environment. It works by treating spatiotemporal nonstationarity as a concept-drift problem. It achieves so by utilising the Hilbert curve and change detection algorithm to identify regions with similar concepts together in space and time. Local models are then built to learn these regions.

3.5.1 Problem Definition

The proposed methodology works on a series of data points listed both in time and space order. This can also be described as an expansion of time-series problem which includes also spatial dependence in addition to the temporal dependence. When a variable Z is observed over time at two or more locations, it is both a spatial series and a time series and can be referred to as a space-time series $z = \{z(m, n, t) \mid m, n \in S, t \in T\}$ in spatial domain S of size $m \times n$ and temporal interval T . A spatiotemporal series may exhibit spatiotemporal dependence which describes its evolution over space and time. If the spatiotemporal dependence in a dataset can be modeled then one essentially has predictive information.

Although sharing many commonalities in techniques and concepts, the fields of time series analysis and spatial analysis have largely developed separately from one another. The behavior of a variable over space differs from its behavior in time. Time has a clear ordering of past, present and future while space does not.

3.5.2 Adaptive Method

The method proposed works by untangling the spatiotemporal series as a 1-dimensional problem utilising the Hilbert curve as interdimensional mapping so that it can be readily fed into a concept-drift detection algorithm, which will give indications of suspected drift points. These cut points mark the start and end of spatial regions with similar statistical properties.

Based on the problem definition laid out in the previous section (3.5.1), the procedure can be explained as follows:

- For every $t \in T$ of space-time series z build a matrix $\mathbf{A}_t^{m \times n}$ that will store the model identifier for every corresponding $m, n \in S$
- The model identifiers can be implemented in the form of integer numbers that is monotonically incremented every time the change detection algorithm detects a change point.
- As new data comes in time, build and populate an error matrix $\mathbf{E}_t = (e_{ij}) \in \mathbb{R}^{m \times n}$ that stores the error (i.e. the difference between the actual and predicted value) for every corresponding $m, n \in S$
- Feed all the entries in the error matrix to the change detection algorithm to detect drifts in concept by transforming the 2-dimensional matrix into a 1-dimensional vector by scanning and ordering the error matrix according to the steps taken by the Hilbert curve.

To illustrate this point, a 4×4 matrix $\mathbf{H} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$

becomes vector $hilbert(\mathbf{H}) = [a b f e i m n j k o p l h g c d]$.

This ordering can be compared with the second-order Hilbert curve depicted in Figure 3.5.

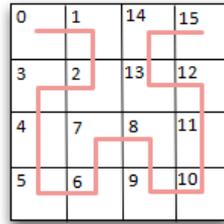


Figure 3.5: *A visualisation of a second-order Hilbert curve with the scanning order derived from it.*

- Every time a change is detected, the last concept is learned by a regressor and saved in a repository of model. As an optional step, a multivariate statistics test based on kNN [Schilling 1986] can be used to check the repository of an existing similar concept that the system has seen and learned before and if one is found, it can be recalled and reused instead of learning a new one. This model reuse means that the system is able to work with recurring concepts both in space and time.
- To make the step-ahead prediction, information in the matrix \mathbf{A} can be utilised to select which model(s) to use for every spatial grid.

The approach laid out in this section is generic and can be implemented in slightly different ways. Algorithm 3.2 describes an implementation of this methodology in pseudo-code form. Variations also include using various other techniques in place of, for example, the Hilbert curve by using other space-filling curve. In this research, we used the MTSKNN package for R [Chen 2010] as the statistical method to compare models in the repository. This test works by appending the instances of both samples together, adding one attribute to indicate from which sample each instance comes from [Henze 1988, Schilling 1986]. Then, the k closest instances of each instance are computed. Next, for each instance, it is verified how many of these k instances come from the same sample. If both samples are drawn from the

Algorithm 3.2 Adaptive Method for Local Models

INPUT: Z (Spatiotemporal stream dataset), t (Start time)

$a \leftarrow 1$

base-model \leftarrow build-model($Z[1:t]$)

model $_a \leftarrow$ base-model

prediction $_{t+1} \leftarrow$ model(z_t)

while there are new incoming data points z_{t+1} **do**

$\mathbf{E}_{t+1} \leftarrow z_{t+1} - \text{prediction}_{t+1}$

$\hat{E}_t = \text{hilbert}(\mathbf{E}_{t+1})$

for all elements $\mathbf{E}_{m,n}$ in \hat{E}_t **do**

add $\mathbf{E}_{m,n}$ to buffer

if ADWIN($\mathbf{E}_{m,n}$) indicates change **then**

$a \leftarrow a + 1$

model $_a \leftarrow$ get-model-from-repository(buffer)

if model $_a ==$ null **then**

model $_a \leftarrow$ build-model(buffer)

save-model(model $_a$)

end if

flush buffer

end if

$\mathbf{a}_{m,n,t+1} \leftarrow a$

end for

for all elements $\mathbf{a}_{m,n}$ in \mathbf{A}_t **do**

prediction $_{m,n,t+1} \leftarrow$ model $_a(z_{m,n,t})$

end for

$t \leftarrow t + 1$

end while

return highpoint

same distribution, the k closest instances might, in average, be equally divided between the two samples. For numerical attributes, their values are used in the computation; for categorical attributes, each category is assigned a sequential positive integer and these values are used in the computation. MTSKNN has been used in another study for recurring context detection with satisfactory results [Gonçalves Jr 2013].

3.5.3 Performance measure

As the experiment works with spatiotemporal data, classical ways of measuring regression prediction performance and accuracy might not be sufficient to account for the spatial aspect of the prediction. Evaluating the predictive performance of models which works with discretised point-process data poses a unique challenge, as the same sparseness prevents the direct use of popular measures such as the root mean squared error [Adepeju 2016]. The following sections will describe the two performance measurements that we use to evaluate the system.

3.5.3.1 N-Test

Adopted from the field of seismology, The N(umber)-test measures the consistency of a forecast with the observed number of earthquakes [Schorlemmer 2010]. The results of the N-test indicate whether a forecast has predicted too many earthquakes, too few earthquakes, or a number of earthquakes that may be considered consistent with the observed number. The N-test does not consider the forecasted or observed spatial distribution of earthquakes. The error at time t is calculated by:

$$e_t = \left\| \sum_{i=1}^m \sum_{j=1}^n z_{i,j,t} - \sum_{i=1}^m \sum_{j=1}^n p_{i,j,t} \right\| \tag{3.4}$$

Which means that all the values in the spatial grids of both the actual

value and prediction matrix a time frame are summed, and then the difference is calculated.

This difference values then will be used to calculate the Mean Average Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^T e_i \tag{3.5}$$

and the Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^T e_i^2}{n}} \tag{3.6}$$

3.5.3.2 Predictive Accuracy Index (PAI)

Commonly used in criminology to predict 'hotspots' where crime might happen often, PAI was proposed as ratio of hit-rate to area coverage [Chainey 2008]:

$$PAI = \left(\frac{n}{N}\right) / \left(\frac{a}{A}\right) \tag{3.7}$$

where:

- n : number of crimes in areas where crimes are predicted to occur (e.g. hotspots)
- N : number of crimes in study area
- a : area (e.g. km²) of areas where crimes are predicted to occur (e.g. area of hotspots)
- A : area (e.g. km²) of study area

This index has been devised to consider the hit rate against the areas where crimes are predicted to occur with respect to the size of the study area. The PAI is calculated by dividing the hit rate percentage (the percentage of crime events for a measurement data time period falling into the areas where

crimes are predicted to occur determined from input data i.e. the crime hotspots) by the Area Percentage (the percentage area of the predicted areas (the hotspots) in relation to the whole study area.

Although the PAI is a considerably good measurement for hotspot-type predictions, it has been noted that it may not be too appropriate for methods that are not specifically made to produce hotspots as their output [Levine 2008]. For regression methods, usually a threshold value is selected as a cut-off between 'hotspots' and not. To avoid this parametrisation, we modified the calculation by making a a constant where $1 \leq a \leq A$. In turn, the n is obtained by counting the number of crimes in areas where the models predict the a highest predicted value in that particular time frame.

3.6 Conclusions

In this chapter, the theoretical foundation and a description of a method for building an adaptive spatiotemporal predictive model using concept-drift detection is presented. It works by identifying regions in space and time which are homogeneous. This is achieved by treating the data stream as a concept-drift detection problem. It is possible to use existing change detection algorithms by mapping the data according to the traversal of Hilbert curve. Finally, this information is then used to determine the local models to be learned. Several performance metrics that can be used to evaluate such system are also discussed. In the next chapter, the method is applied to a real-world dataset concerning aftershocks.

Earthquake Aftershocks

Modelling Experiment

4.1 Introduction

In order to test the effectiveness of the local predictive models obtained by the method described in Chapter 3, a real-world spatiotemporal dataset derived from the earthquake catalogue of the Christchurch region was curated. This area is selected for its seismicity, which has been remarkably very active since the famous 7.1 moment magnitude earthquake in September 2010 hit [Stramondo 2011]. This event has triggered a chain of activities that presumably exhibits a continuously changing flow of processes and concept drifts in both space and time, which is of interest in this research.

4.2 The Dataset

This section will explain briefly about the data being used in the experiment to familiarise the readers to the context regarding the dataset in the problem domain of seismology and geostatistics and the historical events surrounding it.

4.2.1 The Seismicity of Christchurch

In the first 80 years of the city's history since its modern establishment in 1848, four large earthquakes significantly damaged the growing settlement. A 1991 report by the EQC stated that if any one of these four events would happen again, it would cost the city millions of dollars in direct damage and could result in major disruption to the local economy [Elder 1991]. This unfortunately came true at 4:35 am on September 4, 2010. This principal event, sometimes referred to as the Darfield earthquake, struck with an astounding 7.1 magnitude and triggered series of tremors that occurred within and near the city of Christchurch and the Canterbury Plains region from early September 2010 to late December 2011. The earthquake's epicentre was 10 km beneath the surface and located 40 km west of Christchurch near the town of Darfield. It was caused by movement along a previously unknown regional strike-slip fault later called the Greendale Fault, the displacement of which created a 29.5 km long surface rupture [Nicol 2010]. Thousands of smaller aftershocks along with several big events occurred in the months that followed.

4.2.2 Data Acquisition and Preprocessing

The earthquake catalogue was downloaded from the GeoNet online service provided by GNS Science and The Earthquake Commission of New Zealand that can be accessed through the website <http://www.geonet.org.nz>. Attributes supplied by the GeoNet for every earthquake includes the origin time, the latitude and longitude coordinates of the epicentrum, the magnitude, the depth, and so forth.

An arbitrary spatial domain around the city of Christchurch was determined as the region of interest for this study. Most of what are considered aftershocks are located over the full area of fault rupture and either occur

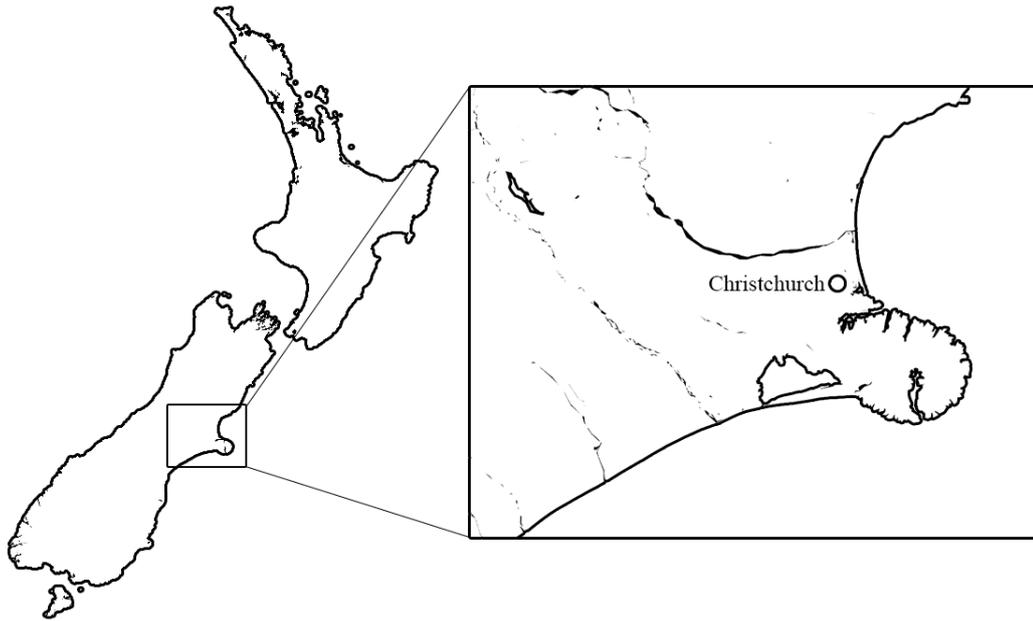


Figure 4.1: *The map of New Zealand and the studied region. Coordinates: North -43.0096, East 173.3741, South -44.1974, and West 171.3132.*

along the fault plane itself or along other faults within the volume affected by the strain associated with the main shock [Wells 1994]. This region is deliberately selected to encompass the major faults that were thought to be involved in the seismicity of Christchurch, namely the Christchurch Fault, Greendale Fault, and the Port Hills Fault [Bannister 2012]. The total area of the region of interest spans approximately 22140 km^2 (167.6 km width and 132.1 km height), depicted by Figure 4.1.

The spatial point data is discretised into bins of the total count of shocks in grids the granularity of which depends on the specified spatial and temporal resolution. As an example, Figure 4.3 visualises this transformation step of the earthquakes recorded in the study region for the first 24 hours after the 2010 Darfield earthquake as a grayscale heatmap, where colour intensity corresponds to the count value at the discrete grids. For the purpose of this experiment, the studied time period starts from 4 September 2010 (two days

before the 2010 Darfield earthquake) up until 27 September 2012, a total of 754 days. This corresponds to point 246 to 1000 on the X-axis of Figure 4.4. This time window is selected because of the noticeable flow of changes in seismic processes and activities both in space and time caused by the occurrence of several big events. The spatial discretisation resolution is 2 x 2 km, which resulted in a 66 x 84 (5544) data points per time slice.

It has to be noted that this dataset might introduce some bias because it contains events which are specific to a region and time period. However, this thesis maintains that the results from this dataset is indicative of the method's general performance based on the generalisability of aftershocks behaviour as described in the following section (4.2.3). The selection of the geographical and time spans of the dataset was also driven by pragmatic concerns. While it was possible to enlarge both the time and spatial boundaries the aim of this thesis was to make a contribution in the Computational Intelligence field and not to solve the Earthquake prediction problem per se. Earthquake prediction remains as one of the most significant open problems in Science and will likely remain so in the coming decades.

4.2.3 Data Characteristics

The data used in this experiment is a real-world dataset the property of which has been extensively studied in the field of seismology. In this section the governing empirical laws regarding the statistical characteristics of aftershocks will be discussed.

4.2.3.1 Omori-Utsu's Law

One of the earliest attempts to find the empirical relationship between The frequency of aftershocks and time after the main shock was known as the Omori's law. It is expressed as:

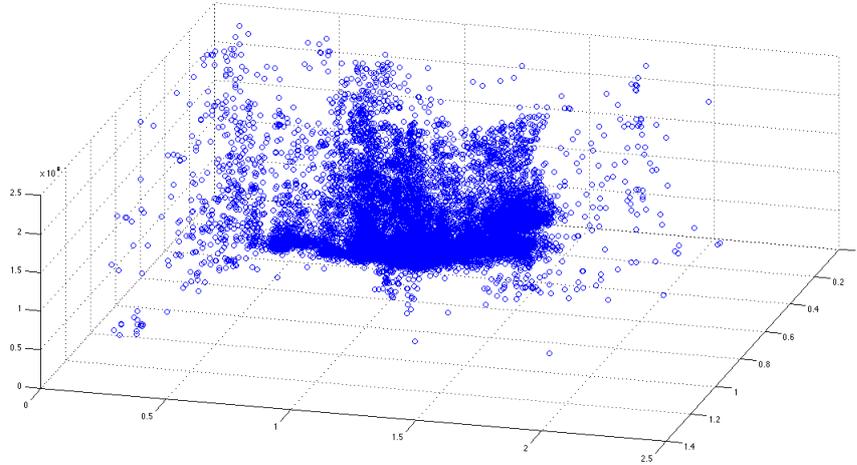


Figure 4.2: *Seismic shocks in the studied region and time period, visualised in 3D as point-process, displaying concept drifts both in space and time. Vertical axis is time and the horizontal axes are longitude and latitude.*

$$n(t) = \frac{k}{c + t} \quad (4.1)$$

where k and c are constants, which are different between earthquake sequences and has to be calculated or estimated after the sequence subside [Omori 1894]. Nowadays, the commonly used form of Omori's law is the modified version, commonly called Omori-Utsu's law:

$$n(t) = \frac{k}{(c + t)^p} \quad (4.2)$$

where p is a third constant which acts as the decay modifier and typically falls in the range of 0.8 to 1.2, depending on the earthquake sequence [Utsu 1961].

It can be inferred from these equations that the rate of aftershocks decreases quickly with time. This swift post-event decay is visible by examining Figure 4.4. The rate of aftershocks is proportional to the inverse of time since the mainshock and this relationship can be used to estimate the probability

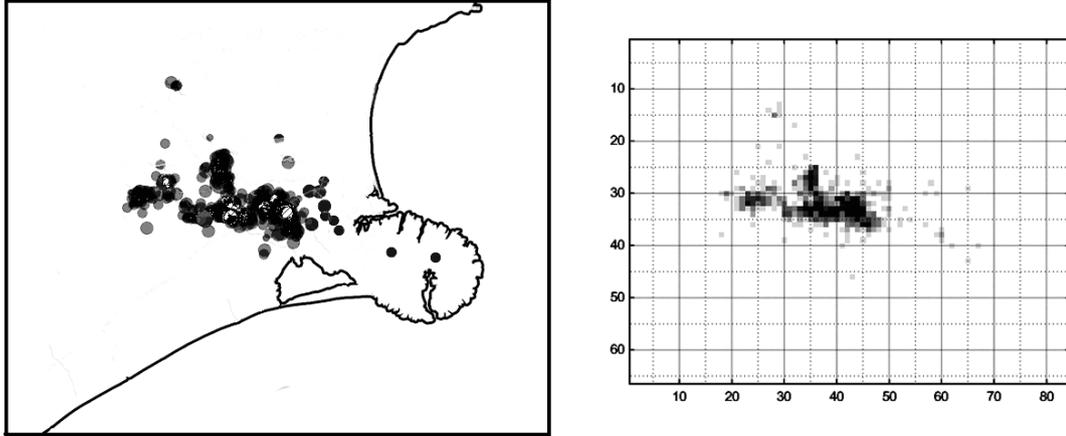


Figure 4.3: *The spatial point map of the earthquakes recorded 24 hours after the 2010 Darfield earthquake (left) and the discretised version as a grayscale heatmap (right). Each pixel represents a 4 km^2 area ($2 \times 2 \text{ km}$ resolution).*

of future aftershock occurrence. These patterns describe only the statistical behavior of aftershocks. The actual times, numbers and locations of the aftershocks are stochastic, while tending to follow these patterns.

4.2.3.2 Gutenberg-Richter Law

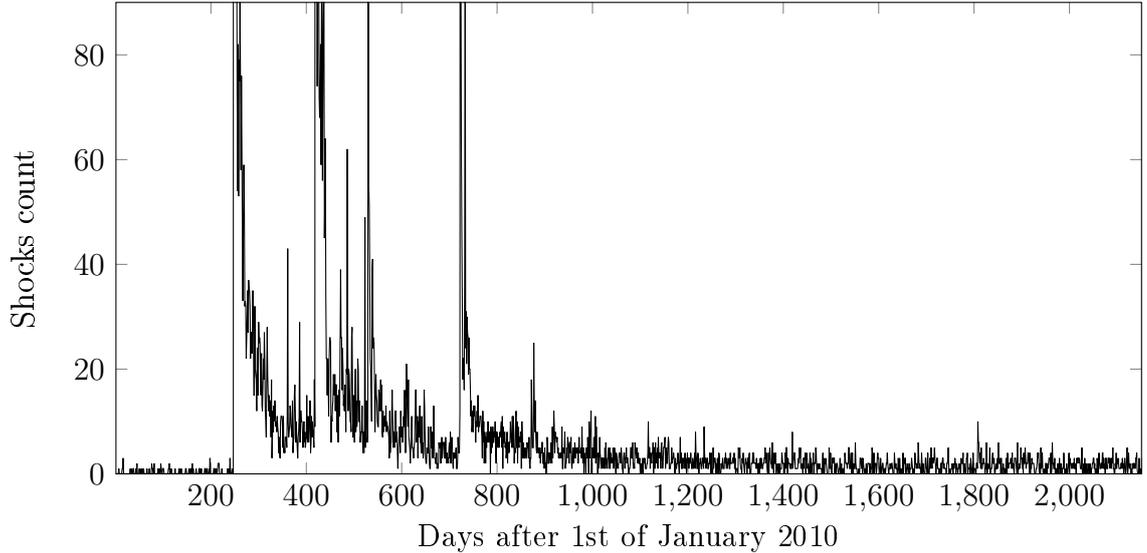
The Gutenberg-Richter Law is an empirical relation between the magnitude x of some seismic event, and $N(x)$, the number of events with magnitudes higher than x [Gutenberg 1956]. This can be written as:

$$N = 10^{a-bM} \quad (4.3)$$

Where N is the number of events greater or equal to M (magnitude), and a and b are constants. According to the equation, there are more small aftershocks and fewer large aftershocks.

While the Omori and Gutenberg-Richter laws are useful in describing general trends of aftershocks they cannot be used for predictive purposes, in the sense of forecasting the number of aftershocks in a given geographical

Figure 4.4: *Time-ordered spatially-aggregated daily frequency of earthquakes from January 2010 to November 2015 occurring within the studied area. (Y-axis clipped at 90)*



area and time ahead of the actual activity. Thus they are no substitute for a genuine spatiotemporal forecasting scheme.

4.3 Experimental Design

The main objective of the experiments that are conducted in this chapter is trying to verify if the proposed adaptive framework build local models whose predictive capability, when taken as a whole, is superior to a global model approach.

For the purpose of this experiment, a simple dataset is built from the earthquake catalog that has the 1-step ahead value in time for every spatial grid $z_{m,n,t+1}$ as the dependent variable. The features includes the spatial coordinates (m, n) and the current actual value $z_{m,n,t}$. The learning scheme is incremental; for every new incoming data frame Z_{t+1} , new models are

trained based on the latest data, constrained to just the samples from the current frame Z_t for simplicity. The prediction error for every time step is recorded and aggregated over the whole testing period for performance evaluation. Put differently, the experiment is basically an exercise of building autoregressive models of the aftershock count in an temporally incremental manner.

In this experiment, there are four modes of testing:

- *Adaptive Global*: The performance of which used as a reference point, this mode learns new incoming data without taking into account spatial variance. All data points in a time frame are learned by a single global model that encompasses the whole spatial input. This model is then used to predict the next value Z_{t+1} for every $m, n \in S$.
- *Adaptive Local with Hilbert Curve*: Employing the methodology laid out in Section 3.5.2, this mode traverses new incoming data according to the ranges of the Hilbert curve and feed the data into a concept-drift detector, carving out local regions over the spatial input space that are considered homogeneous. Models then are built to learn according to these cuts.
- *Adaptive Local with Scanline*: This mode is very similar to the previous mode, except that the inter-dimensional mapping that it uses is a simple scanline fill (Figure 4.5) instead of the Hilbert curve. This is to investigate if space-filling curves offer a measurable advantage compared to standard rasters.
- *Adaptive Local with Hilbert Curve and Model Recall*: In addition to the localisation process, data and the models that has been learned by the system are stored in a repository and every time a new region is identified, the system will try to find a model that has previously been learned

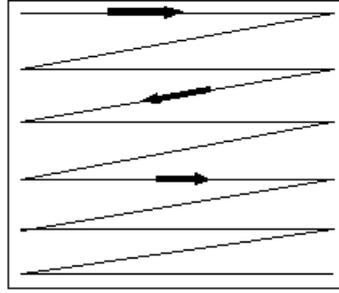


Figure 4.5: *Illustration of raster scan-line.*

with similar statistical properties. Instead of building and learning a new one, old models are recalled and updated every time a recurring concept is detected. If the concept is not found in the repository, the system will simply build a new model. In this experiment, the model recall is only enabled after 450 time points to give the system enough models to populate the repository.

The different modes are designed to truly understand the behaviour of the proposed system. Furthermore, learning algorithms whose characteristics are well-understood were chosen. Linear Regression (LR) and Multi-layer Perceptron (MLP) are chosen to observe how both linear and non-linear models perform when used as local learners.

4.4 Results and Analysis

The result of the predictive accuracy test on the whole dataset is listed in Table 4.1. Figure 4.6 visualises the N-Test in a graph form. The N-Test error rate of both approach is compared in Figure 4.7. As we hypothesised, the local models almost always give better performance in some way compared to the global model. For example, with LR as the regressor, introducing the localisation technique gives us a 9% increase in performance in terms of the PAI, although only a meager 1.1% improvement with the RMSE. This is also

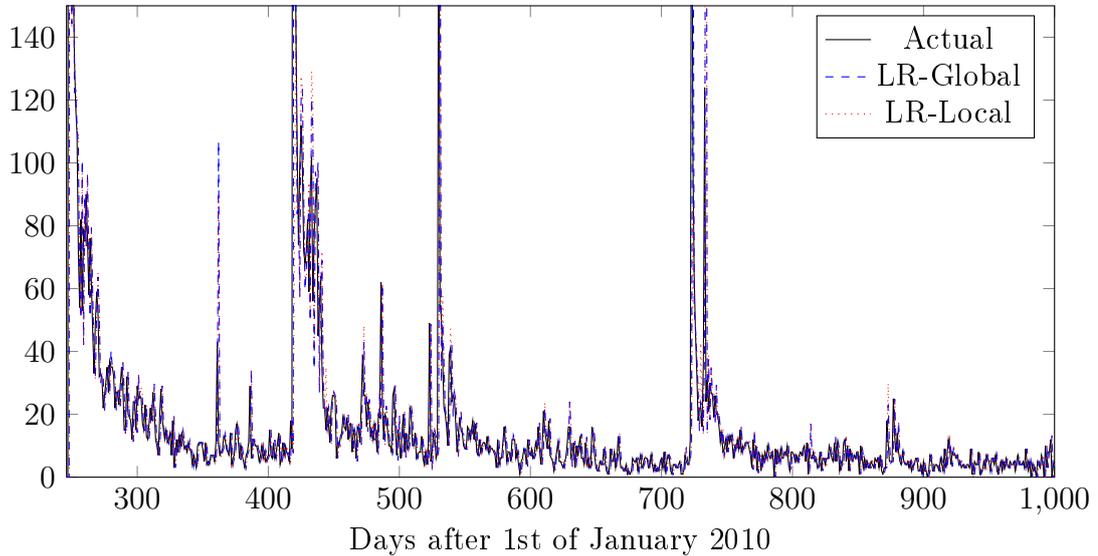
Table 4.1: *The result of the predictive accuracy test on the whole dataset. Higher is better for PAI, lower is better for N-Test. The model count tracks the number of local models built by the system.*

		PAI		N-Test		Mdl Count
		Avg	Stdev	MAE	RMSE	
LR	Global	4.773	9.410	10.226	51.773	754
	Local,Hilbert	5.206	9.965	9.973	51.190	1279
	Local,Scan	5.553	10.080	10.804	55.275	1755
	Local+Recall	4.676	9.289	13.524	58.307	894
MLP	Global	4.487	9.030	14.284	59.837	754
	Local,Hilbert	4.954	9.510	13.889	55.815	1095
	Local,Scan	5.106	9.559	15.331	65.248	1458
	Local+Recall	4.004	8.405	16.220	71.796	734

the case with MLP, with 10.4% better PAI and 6.7% smaller RMSE. This gain in performance is at the cost of increasing the amount of models that were trained and used. MLP performed generally worse presumably due to over-fitting since the data is linear.

Surprisingly, the scanline fill gives better result in terms of the PAI compared to the Hilbert curve. With 16.3% and 13.8% gain using LR and MLP respectively, it offers quite a respectable increase in performance. However, it suffers a significant penalty in regards to the RMSE performance: it increases the error up to 9% with MLP even compared to the global technique. The gain in PAI and loss in RMSE with scanline are due to the fact that this method is prone to identifying uninteresting low-activity hotspots which has low absolute error rate, inflating its PAI performance. This is why it is essential that multiple ways of measuring performance are used. It also produces a less compact representation than the Hilbert space, needing up

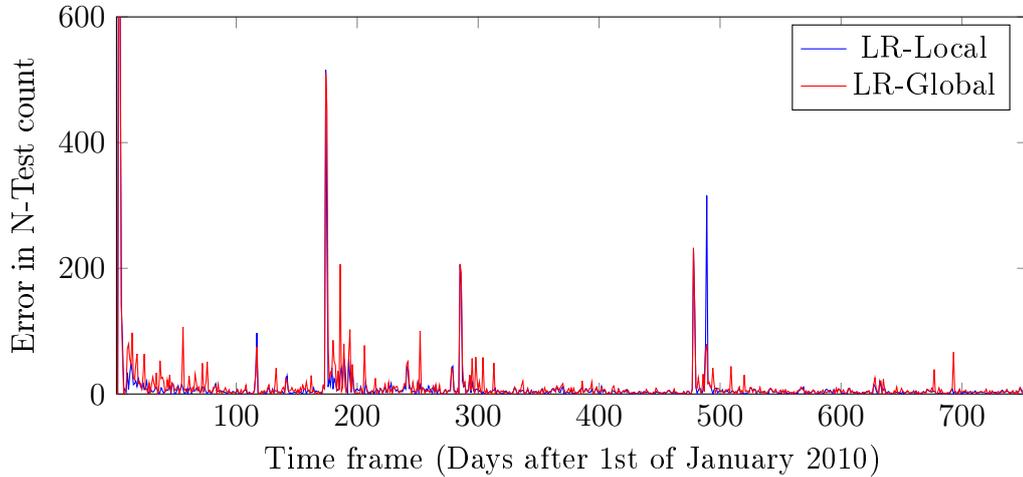
Figure 4.6: *Time-ordered daily frequency of earthquakes along the study period showing the resulting N-Test prediction. (Y-axis clipped at 150)*



to almost 500 extra models built when used with LR. This finding seems to further confirm the locality-preserving property of the Hilbert curve.

With the recurring concept detection and model recall enabled, performance generally suffer compared to without. This was expected, as it reuse old concepts and also the accuracy relies heavily on the appropriateness of the model selection in the repository. Although it achieved the worst performance overall, it demonstrated the potential to produce a more compact representation of the data than without recall, needing hundreds fewer models over the course of the test period. This is indicated by the model count, i.e. the number of both model creation and update. Since the experiment run on incremental learning scheme, 754 model updates are made for the global methods as there are 754 time frame. The performance in this mode also depends on how many models the system is allowed to learn initially, which was arbitrarily constrained to $t > 450$ in this experiment.

Figure 4.7: A graph displaying the *N-Test* error rate of both the local and global approach at every time frame.



4.4.1 Visualisations

In this section, some selected visualisations that can be obtained from the experiments are displayed and discussed.

4.4.1.1 Local spaces

Figure 4.8 and 4.9 shows the local partitions that are created with the Hilbert curve in conjunction with ADWIN for two particular time slices where the initial aftershocks are unfolding from two different large earthquakes, exhibiting differing concepts in space. In the partition figures, the colour scheme indicates different regions without implying any ordinality. On the other hand, Figure 4.10 shows the polygons of local models with the number of shocks inside each locality. This illustrates the significant variance between regions carved out by the system. It can be seen from the figures that because of the nature of Hilbert curve's traversal, the partitions could be very fragmented (Figure 4.8) or relatively unfragmented (Figure 4.9) depending on the spatial distribution of the data. We reckon that time frame 248 seems

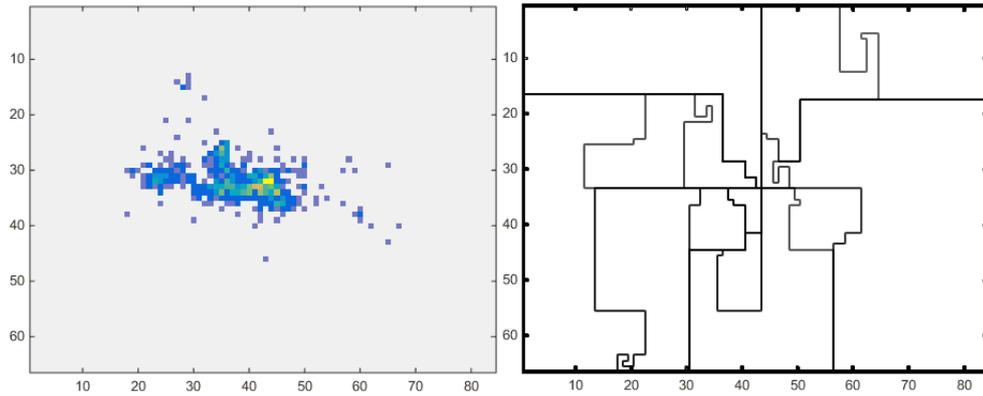


Figure 4.8: *The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Hilbert curve traversal (right).*

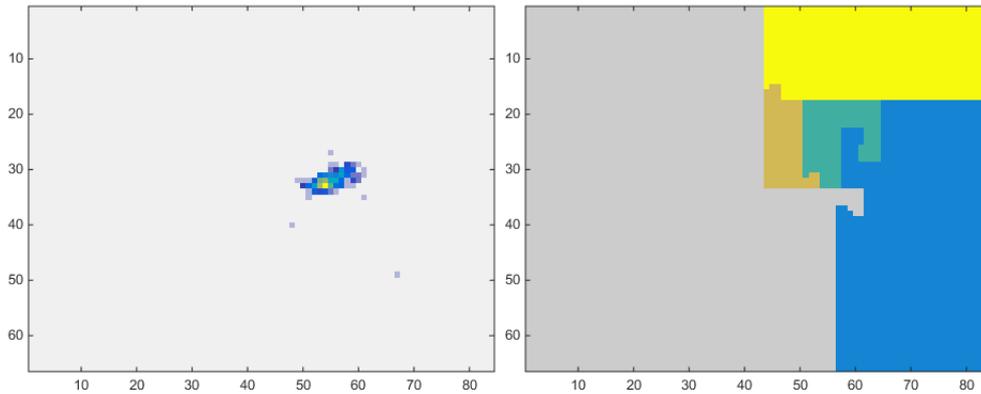


Figure 4.9: *The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Hilbert curve traversal (right).*

to be an example of a worst-case scenario for a Hilbert traversal because the curve passes through the boundaries of the non-zero data multiple times, forcing the concept-drift detector to divide the area in a very fragmented way. On the other hand, time frame 419 shows a case where the Hilbert curve can isolate the region quite well. Despite the seemingly very fragmented partitioning, it visually produces a much better outcome compared to using scanline fill as depicted by Figure 4.11.

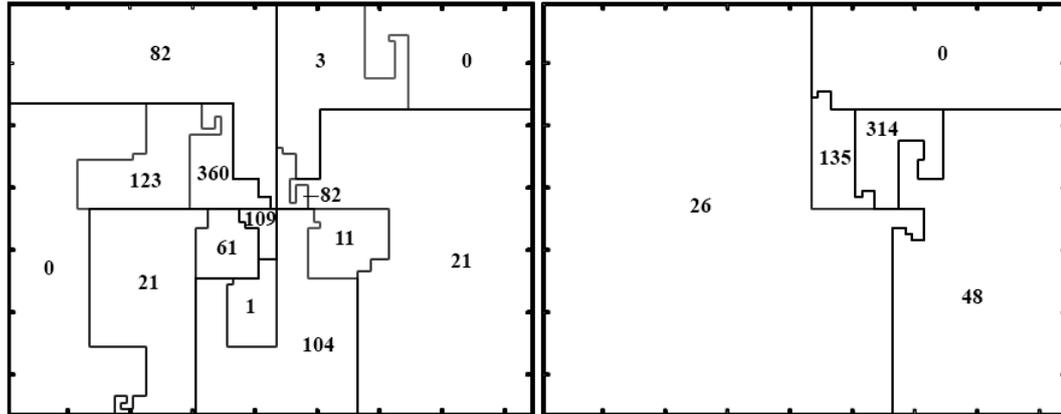


Figure 4.10: *This figure shows the local spaces made for time frame 248 (left) and 419 (right) with the aggregate number of shocks in each locality to illustrate the inter-group variance separated by the method.*

4.4.1.2 Hotspot Maps

As explained in Section 3.5.3.2, the PAI is a performance index used to evaluate the effectiveness of hotspot outputs made by a predictive model. In this section, select hotspot maps produced by the experiments are displayed to give a better sense of what different values of PAI entails visually.

Figure 4.12, 4.13, and 4.14 exhibit example snapshots at time frame 253, 272, and 306 respectively. Figure 4.12 is an example where the predictive models produced a very good hit rate between the actual spatial distribution of aftershocks and the predicted hotspot map, scoring very high on the PAI at 26.78. Figure 4.13 shows a prediction with decent hit rate, but with a considerable amount of misses as well.

Of interest is Figure 4.14, where the PAI is 0, meaning that at the predicted hotspots, no aftershocks came forth. However, although it is considered a total miss by the way PAI is calculated, the map still visually show useful indications. This demonstrates the imperfection of PAI as a performance measure, which is why in evaluating the spatiotemporal data mining problems, multiple measures have to be used to fully capture the intricacies

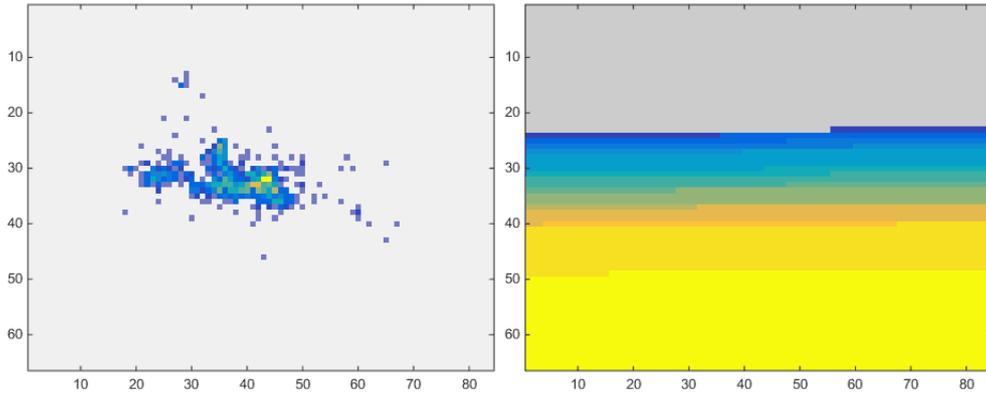


Figure 4.11: *The actual data at time frame 248 (left) and the local partitions created by LR-Local mode with Scanline fill traversal (right). It produced vertically separated local regions which do not preserve locality.*

of error in spatial and temporal sense.

An additional visualisation of interest is the system’s ability to obtain a projection of concept drifts in the dataset in space and time that is recorded by the system as it process the data stream continuously adapting to incoming data, which can be seen in Figure 4.15. A series of such slices taken over time would provide a visualisation of the change dynamics of the system.

4.5 Conclusions

In this chapter, an experiment of modelling spatiotemporal aftershock sequence has been presented. As an exploration of the idea of building adaptive local models, the experiment’s results have shown that the proposed methodology (Chapter 3) generally performed better and have a higher predictive power than global models approach. Further analysis of the results have uncovered interesting observations and visualisations regarding the behaviour of both the model and the data. This study also further confirms the locality-preserving property of the space-filling Hilbert curve by comparing

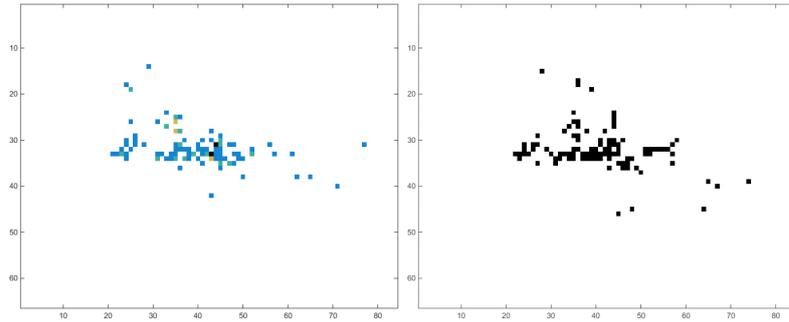


Figure 4.12: *The actual data at time frame 253 (left) and the hotspot prediction created by LR-Local mode (right). PAI: 26.780. The predicted hotspots closely follow the true spatial distribution of aftershocks.*

it with the simple scanline fill.

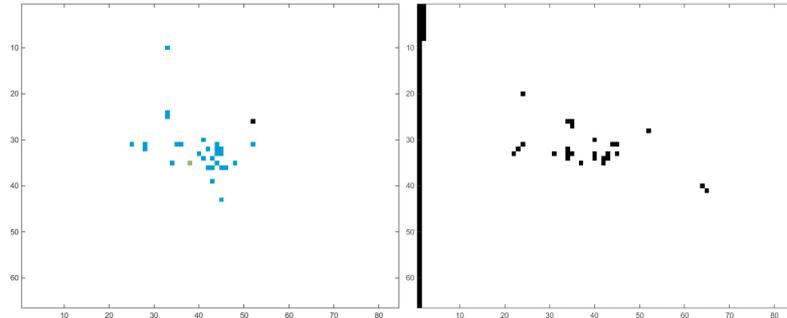


Figure 4.13: *The actual data at time frame 272 (left) and the hotspot prediction created by LR-Local mode (right). PAI: 6.720. The decrease in score reflects how the hotspots following the true spatial distribution of aftershocks more loosely.*

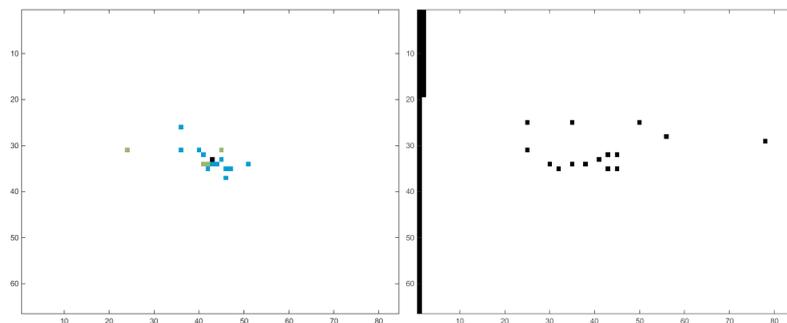


Figure 4.14: *The actual data at time frame 306 (left) and the hotspot prediction created by LR-Local mode (right). Although the prediction can generally indicate the spatial distribution of aftershocks, the PAI score is 0 (complete miss). This demonstrates the importance of using N-test to complement PAI.*

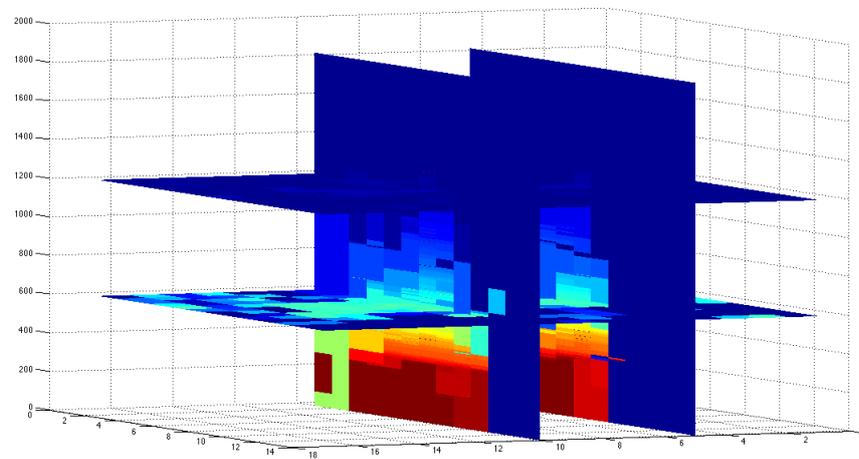


Figure 4.15: *A slice projection of the 3-dimensional heatmap visualising the identified gradually evolving concepts in the dataset over space and time. The Vertical axis is time and every colour represents a concept.*

The NeuCube Architecture for Spatiotemporal Data Analysis

5.1 Introduction

This chapter presents a Spiking Neural Networks (SNN)-based Spatiotemporal Data Machine (STDM) architecture named NeuCube, first proposed in [Kasabov 2012] and developed at KEDRI by Prof. Nikola Kasabov and other people in the group. Designed initially to map and model brain signals captured in the form of EEG streams, the architecture has found its way as well into several non-brain data case studies such as the prediction of stroke occurrence with weather variables [Kasabov 2014b]. The organisation of this chapter is as follows:

- In Sections 5.2 through to 5.8, the theoretical background and the fundamental principles that serve as the basis of NeuCube and the conceptual framework of the STDM will be presented. This includes a primer on biological neurons (Section 5.2) and a brief discussion on how artificial neural networks developed as a mathematical model of the biological neuron (Section 5.3). It is then followed by an introduction to SNN (Section 5.4), its data encoding method (Section 5.5), and its training algorithms (Section 5.6) which includes the Spike-timing-dependent plasticity (STDP). Finally, the principles of Reservoir Com-

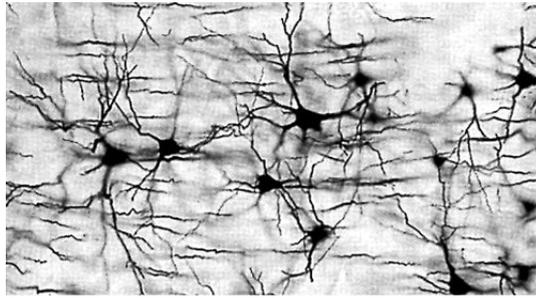


Figure 5.1: *Golgi staining works by soaking brain tissue in silver chromate solution. Using the stain, a small percentage of the neurons became darker, revealing their total structures.*

puting and the Evolving Connectionist System (ECOS) are laid out in Section 5.7 and 5.8 respectively.

- Section 5.9 explains the overall architecture of NeuCube, including its contemporary implementations and applications.

5.2 The Neuron

To better understand how SNN works, it is imperative to have an overview of the real-world object that it is built or modeled upon, namely the Neural cells of the brain. The human brain, or more generally the Central Nervous System (CNS) of animals, are composed primarily of two broad classes of cells: neurons and glial cells. Neurons, however, are usually considered the most important cells in the brain since most of information processing seem to happen in the neural structures.

Humanity has long speculated that the head (and thus the brain) has something to do with mentality, as attested by the records of symptoms of brain damage and evidence of cranial surgery performed by the Egyptian, Mesopotamian, Chinese, Indian, Greek, and Roman civilisations [Finger 2001]. However, not until after the invention of the microscope and the development

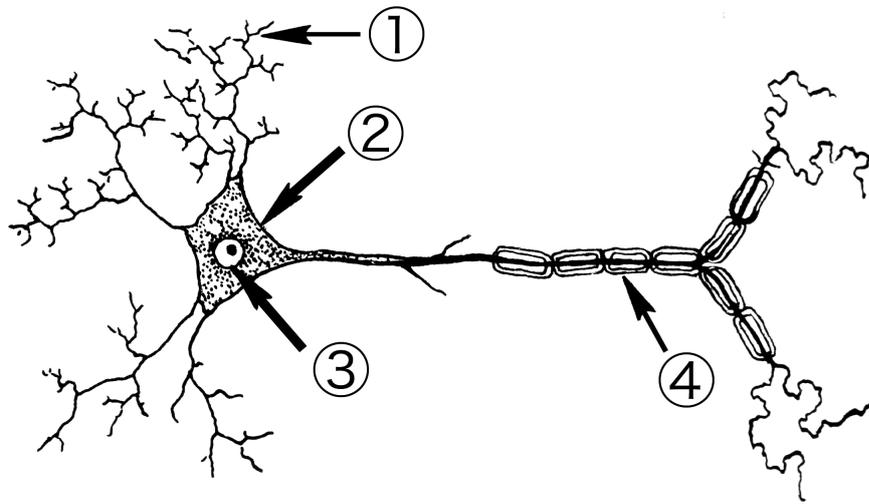


Figure 5.2: *Line art drawing of a Neuron. The numbers indicate the structures that constitute a Neuron: 1. Dendrites, 2. Cell Body, 3. Nucleus, 4. Axon*

of a staining procedure by Camillo Golgi during the late 1890s that studies of the brain became more sophisticated. The procedure was able to reveal the intricate structures of individual neurons by using a silver chromate salt staining (Figure 5.1). This is a breakthrough that led to the conception of The Neuron Doctrine by Santiago Ramón y Cajal, which is considered to be the cornerstone of the field of Neuroscience [López-Muñoz 2006].

Unlike other types of cells, a neuron's morphology can consist of additional structures outside the cell body, like dendrites and axons (Figure 5.2). In the strictest sense, a neuron is a cell that processes and transmits information through electrical and chemical signals. This transmission and processing of information is possible because neurons are electrically excitable, maintaining voltage gradients across their membranes by means of metabolically driven ion pumps, which combine with ion channels embedded in the membrane to generate intracellular-versus-extracellular concentration differences of ions such as sodium, potassium, chloride, and calcium. This force

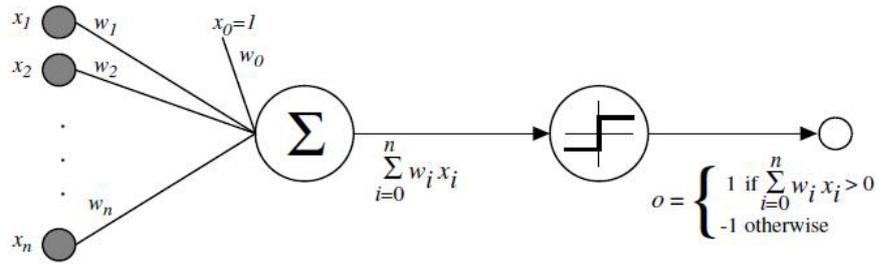


Figure 5.3: A diagram showing how the Perceptron works.

that generates neural activity is called action potential, and the excitation manifest itself in the form of electrical spikes [Hodgkin 1952b].

5.3 Artificial Neural Networks

As our knowledge of the nervous system improve, scientists from various backgrounds have come up with mathematical models that tries to simulate the workings of neurons. For example, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper in 1943 on how neurons might work by modelling a simple neural network using electrical circuits. [McCulloch 1943]. However, the McCulloch-Pitts model lacked a mechanism for learning, which was crucial for it to be usable for AI.

Another breakthrough came in 1949, when the Canadian psychologist Donald O. Hebb published the theory of Hebbian learning, which he introduced in his classic work *The Organization of Behavior* [Hebb 1949]. This then in turn inspired the Perceptron; in 1958, the American psychologist Frank Rosenblatt came up with a way to make such artificial neurons learn [Rosenblatt 1958].

The Perceptron (Figure 5.3) did not follow Hebb’s idea exactly, but having weights on the inputs allowed for a very simple and intuitive learning scheme: given a training set of input-output examples the Perceptron should ‘learn’ a

function from, for each example increase the weights if the Perceptron output for that example's input is too low compared to the example, and otherwise decrease the weights if the output is too high. This procedure is simple, and produces a simple result: an input linear function (the weighted sum), just as with linear regression, 'squashed' by a non-linear activation function (the thresholding of the sum).

Perceptron's simple training algorithm imposed so many limitation on the learning capability of perceptron. It could not learn multi-class patterns and because it can only train a single computation unit, it was practically only a linear function which could not even learn the XOR [Minsky 1969]. After an AI winter of little progress and unfulfilled expectations, people came to realise that multiple layers of perceptrons can be trained using Backpropagation algorithm [Rumelhart 1988]. It is such a powerful combination that this configuration has been mostly the way ANNs are built until today.

Although biologically inspired, all these methods worked with real values and thus ignore the inherent temporal aspects of biological neurons, which works by processing electrical signals travelling through its synapses in the form of spike trains. In the following sections, more biologically plausible models belonging to a next-generation class of artificial neurons called SNN will be discussed.

5.4 Spiking Neural Networks

5.4.1 The Hodgkin-Huxley Neuron Model

The model described by Alan Lloyd Hodgkin and Andrew Huxley in 1952 is considered the first scientific model of how action potentials are initiated and propagated in a spiking neuron [Hodgkin 1952b]. The paper was the finale of a series of five papers in the same journal describing their experiment with

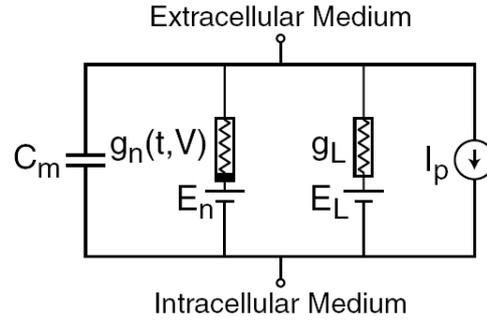


Figure 5.4: *Hodgkin-Huxley type models represent the biophysical characteristics of cell membranes. The lipid nilayer is represented as a Capacitance (C_m). Voltage-gated and leak ion channels are represented as by nonlinear (g_n) and linear (g_L conductances, respectively. The electrochemical gradients driving the flow of ions are represented by batteries (E), and ion pumps and exchanges are represented by current sources (I_p). Image: Behrang Amini.*

an axon from a Giant Squid [Hodgkin 1952a, Hodgkin 1952c, Hodgkin 1952d, Hodgkin 1952e].

Figure 5.4 describes the famous Hodgkin-Huxley neuron model in the form of an electrical circuit schematic diagram. Capacitors are used to model the charge storage capacity of the cell membrane, and batteries are used to represent the electrochemical potentials established by differing intra- and extracellular ion concentrations [Nelson 2004].

The model can be mathematically described as:

$$I_{Inward} = C_m \frac{dV}{dt} + I_{Na} + I_k + I_l \quad (5.1)$$

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 is derived using Ohm's law ($V = IR$)

$$I_{Na} = g_{Na}(E - E_{Na}) \quad (5.2)$$

$$I_k = g_k(E - E_k) \quad (5.3)$$

$$I_l = g_l(E - E_l) \quad (5.4)$$

where g_{Na} and g_k are both functions of time and membrane potential. While E_{Na} , E_k , E_l , C_m and g_l are all constants that are determined via experimentation.

The model can further be expand by adding the following relationships:

$$I_{Na} = g_{Na}(V - V_{Na}) = g_{Na}(E_{Na} - E_R) \quad (5.5)$$

$$I_k = g_{Na}(V - V_k) = g_k(E_k - E_R) \quad (5.6)$$

$$I_l = g_{Na}(V - V_{Na}) = g_{Na}(E_l - E_R) \quad (5.7)$$

$$V = E - E_R \quad (5.8)$$

where E_R is the resting potential. The simplifications 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:

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \quad (5.9)$$

$$\frac{dm}{dt} = \alpha_m(1 - n) - \beta_m m \quad (5.10)$$

$$\frac{dh}{dt} = \alpha_h(1 - n) - \beta_h h \quad (5.11)$$

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.

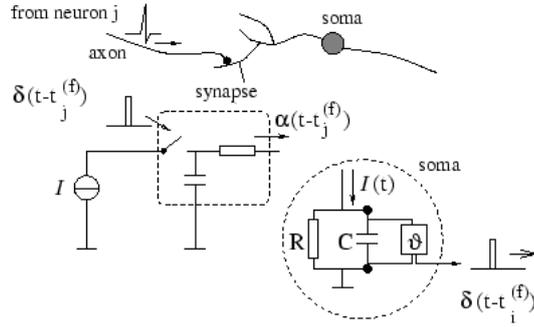


Figure 5.5: Schematic diagram of the integrate-and-fire model. The basic circuit is the module inside the dashed circle on the right-hand side. A current $I(t)$ charges the RC circuit. The voltage $u(t)$ across the capacitance (points) is compared to a threshold ϑ . If $u(t) = \vartheta$ at time $t_i(f)$ an output pulse $\delta(t - t_i(f))$ is generated. Left part: A presynaptic spike $\delta(t - t_j(f))$ is low-pass filtered at the synapse and generates an input current pulse $\alpha(t - t_j(f))$.

5.4.2 Leaky Integrate and Fire Model

The Hodgkin-Huxley model can model and replicate electrophysiological measurements very accurately and its parameter can be easily evaluated from the experiments. However, the model is highly complex and makes it too computationally expensive for large network of spiking neurons. This section discusses the Leaky Integrate-and-Fire (LIF) model, which is a type of spiking neurons popular and widely used because it preserves a large part of the biological realism in a computationally inexpensive way [Gerstner 2002].

The basic circuit of an integrate-and-fire model consists of a capacitor C in parallel with a resistor R driven by a current $I(t)$ (Figure 5.5); a LIF neuron is a simplified Hodgkin-Huxley model where all the ion channels are represented with a single current [Stein 1967]. Considering the Ohm's law and the Equation 5.1, we can derive:

$$I_t - \frac{V_m(t)}{R_m} = C_m \frac{dV_m(t)}{dt} \quad (5.12)$$

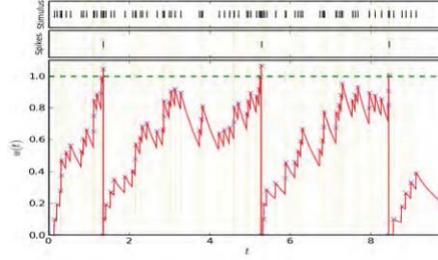


Figure 5.6: *The characteristic of LIF neurons where the membrane potential, increase with every input spike at a time t multiplied to the synaptic efficacy (strength) until it reaches a threshold and emit an output spike then the membrane potential is reset to initial state.*

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. We multiply Equation 5.12 by R and introduce the time constant $\tau_m = RC$ as the 'leaky integrator'. This yields the standard form:

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \quad (5.13)$$

We refer to u as the membrane potential and to τ_m as the membrane time constant of the neuron. These types of artificial neuron has exhibited a behaviour very close to the biological counterparts; Figure 5.6 showed the characteristic of the membrane potential of LIF neurons.

5.5 Spike Encoding

Although it is known that neural cells use action potentials to encode and transmit information, there has been no consensus as to the exact way biological neuron encodes and decodes information from and into conceptual element [Bishop 1999]. As the information we are working with computationally are mostly represented in real value, the information must be encoded

into trains of spikes to work with SNN. The number of possible encoding schemes are theoretically numerous. These methods can be grouped into two main approaches, which are rate and temporal coding. Whether neurons use rate coding or temporal coding is still a topic of intense debate.

5.5.1 Rate coding

This encoding method counts the average number of spikes within a specific time window without taking into account the exact timing of the spikes. Rate coding has been shown to be the mechanism by which sensory neurons encode how much force is applied on muscle tissues [Adrian 1926]. The rate is generally calculated by dividing the number of spikes emitted during a time window. There are special forms of this general formula for calculating the average rate over multiple neurons, differing time-windows, and experimentations [Gerstner 2002].

Rate coding is often considered a traditional coding scheme due to its simplistic assumption that most, if not all, information about the stimulus is contained in the firing rate of the neuron. It is also relatively easy to measure experimentally. For decades after it was discovered, measurement of firing rates became a standard tool for describing the properties of sensory or cortical neurons. However, this approach neglects all the information possibly contained in the temporal structure of the spikes. Experimental evidence has come about suggesting that a straightforward rate coding may be too simplistic to describe more advanced neuronal activities, especially in the brain [Stein 2005].

5.5.2 Temporal coding

Neural codes are often said to be temporal when precise spike timing or high-frequency firing-rate fluctuations are found to carry information [Dayan 2003].

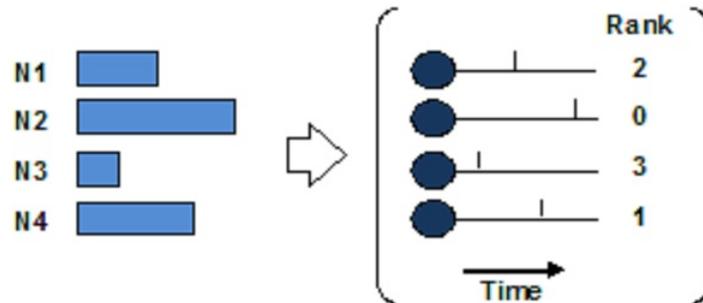


Figure 5.7: Rank Order Coding [Thorpe 1998]

Temporal coding considers the precise spike timing as important, and that the fluctuations of the spiking order carry information.

Temporal codes make use of the time-related features of the spiking activity that are not described by the firing rate. Examples of temporal codes include time to first spike after the stimulus onset, spike randomness, or precisely timed groups of spikes [Kostal 2007].

An example of method based on temporal coding is the Rank Order Coding (ROC). ROC encodes information by using the order of the firing time [Thorpe 1998]. ROC considers only on the order in which the spikes arrive, not the precise timing of information. For example, for four neurons, where $N3 < N1 < N4 < N2$, the rank assigned to each neuron is therefore Rank 0 = N2, Rank 1 = N4, Rank 2 = N1 and Rank 3 = N3 as illustrated in Figure 5.7. 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].

Another example of encoding scheme that make use of the temporal properties of spike trains to encode information is the Phase-of-firing Coding. This code supplements the rate code with a time reference based on oscillations. This means that neuronal spike trains can encode information in the phase of a pulse with respect to the background oscillation [Gerstner 2002]. The biological basis for this scheme is that it has been shown that neurons in some

cortical sensory areas encode stimuli in terms of their spike times relative to the phase of ongoing network fluctuations, rather than only in terms of their spike count [Montemurro 2008].

5.5.3 Population coding

Population coding is a method to represent information regarding stimuli by aggregating the joint activities of a number of neurons. In population coding, each neuron has a distribution of responses over some set of inputs, and the responses of many neurons may be combined to determine some value about the inputs. It accommodates the essential features of neural coding and is well-formulated for theoretic analysis [Wu 2006a].

An example of this approach is called the Position Coding, which involves neurons with a Gaussian tuning curve whose means vary linearly with the stimulus intensity, meaning that the neuron responds most strongly (in terms of spikes per second) to a stimulus near the mean [Bohte 2002]. This Gaussian activation function or receptive field is shown in Eq. 5.14.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (5.14)$$

This function is used to calculate the firing time for the input neuron. The mean μ_i and the width σ are calculated with Eq. 5.15 and Eq. 5.16, respectively.

$$\mu_i = I_{min} = +\frac{2i-3}{2} \cdot \frac{I_{max} - I_{min}}{M-2} \quad (5.15)$$

$$\sigma = \frac{1}{\beta} \cdot \frac{I_{max} - I_{min}}{M-2} \quad (5.16)$$

I_{min} and I_{max} are the minimum and maximum range of input variable and β coefficient controls the width of each Gaussian receptive field, taking any real value between 1.0 and 2.0.

5.6 SNN Learning Algorithm

There exist various ways of training an SNN for different learning modes (supervised, reinforcement, unsupervised, etc.). In this research, we are interested in the unsupervised learning algorithm called Spike-Time Dependent Plasticity (STDP).

5.6.1 STDP

As an algorithm that operate according to the Hebbian type of learning (as mentioned in Section 5.3), it works by modifying the synaptic activity based on global competition (i.e. between neurons' pre- and post-synaptic action potentials) [Hebb 1949]. With STDP, repeated presynaptic spike arrival a few milliseconds before postsynaptic action potentials leads in many synapse types to long-term potentiation (LTP) of the synapses, whereas repeated spike arrival after postsynaptic spikes leads to long-term depression (LTD) of the same synapse [Markram 1997].

It has been shown that STDP was observed in more than 20 different types of synapses from insects to mammals, and from striatum to neocortex [Feldman 2012]. Its cellular basis is increasingly understood, and it has already been widely utilised in computational models of neural network plasticity and learning.

STDP can be mathematically described as calculating the weight change Δw_j of a synapse from a presynaptic neuron j depends on the relative timing between presynaptic spike arrivals and postsynaptic spikes. Thus the total weight change is described as [Kempster 1999, Gerstner 1996]:

$$\Delta w_j = \sum_{f=1}^N \sum_{n=1}^N W(t_i^n - t_j^f) \quad (5.17)$$

where $W(x)$ denotes one of the STDP functions, the pre-synaptic spike

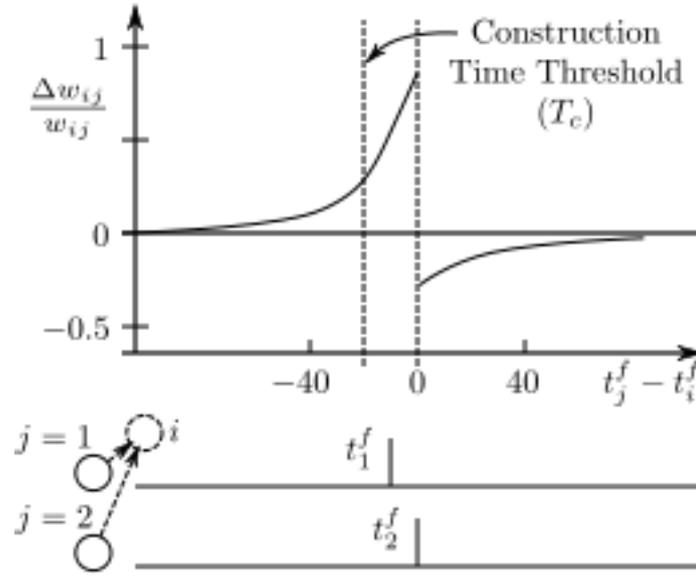


Figure 5.8: *Spike-Timing Dependent Plasticity (schematic): The STDP function shows the change of synaptic connections as a function of the relative timing of pre- and postsynaptic spikes*

arrival times at synapse j by t_j^f where $f = 1, 2, 3, \dots$ counts the pre-synaptic spikes and the post-synaptic spike arrival times at synapse i by t_i^n with $n = 1, 2, 3, \dots$ count the post-synaptic spikes.

An example of STDP function is:

$$W(x) = \begin{cases} A_+ \exp(-x/\tau_+) & , \text{ for } x > 0 \\ -A_- \exp(x/\tau_-) & , \text{ for } x < 0 \end{cases} \quad (5.18)$$

which has been used in fits to experimental data [Zhang 1998]. 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 $\tau_+ = 10\text{ms}$ and $\tau_- = 10\text{ms}$. Pseudo-code for STDP learning is as follows:

Algorithm 5.1 The STDP weight updating algorithm

INPUT: Weight vectors of presynaptic neuron i

for j in input_size **do**

while there are post-spikes for neuron j **do**

 Increment Δw_{ij} using the pre-synaptic spikes at time t_j^{post}

end while

end for

while there are pre-spikes for neuron i **do**

for j in input_size **do**

 Decrement Δw_{ij} using the pre-synaptic spikes at time t_i^{pre}

end for

end while

for j in input_size **do**

 Apply Δw_{ij} to the synapse

end for

5.7 Reservoir Computing

Reservoir computing is a neural network based computational framework for computation where the input signals are fed into a dynamical system called the reservoir, resulting in mapping of the input to a higher dimension. Then the readout function is used to read the states / dynamics of the reservoir for imposing an input output mapping [Schrauwen 2007]. Figure 5.9 provides an illustration of this general principle.

Reservoir computing is an umbrella term which includes a number of independently found approaches based on this fundamental idea, namely Liquid State Machines, Echo State Networks, Backpropagation Decorrelation and Temporal Recurrent Networks. The reservoir comprises a group of recurrently connected neurons. The connectivity is generally random, and the units are typically nonlinear. On the whole, the activity in the reservoir is

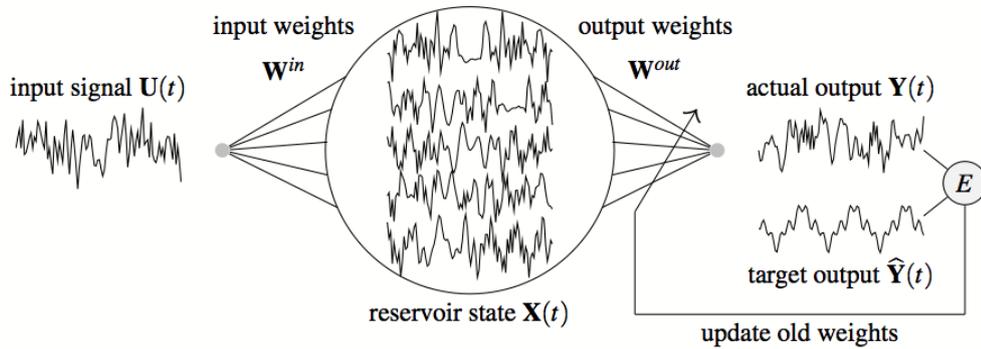


Figure 5.9: *The general schematic diagram of reservoir computing principle.*

driven by the input and is also influenced by the past.

Figure 5.10 illustrates the structure of a Liquid State Machine (LSM), which transforms input streams $u(\cdot)$ into output streams $y(t)$ through a liquid filter L^M . The liquid state $x^M(t)$ is a part of the internal state of the reservoir at time t which is visible to an external observer. During LSM simulation, the synaptic weights, neurons connectivity and parameters are predefined. The continuous stream of input $u(\cdot)$ will cause neurons to respond and generate the liquid spiking activities. The state of the Liquid $x(t)$ can be recorded at different time points, which is simply the result of mapping the input into a higher-dimensional space. This state is then passed to the memoryless readout f^M that will transform into output $v(t)$.

LSM offers an improvement over traditional artificial neural networks in regards to its biological plausibility. It is argued to be more similar to actual neurons because the neural circuits are not hard coded to perform a specific task, and continuous time inputs are handled naturally [Hazan 2012]. Its executions can run on various time scales and multiple computations can be performed on the same network.

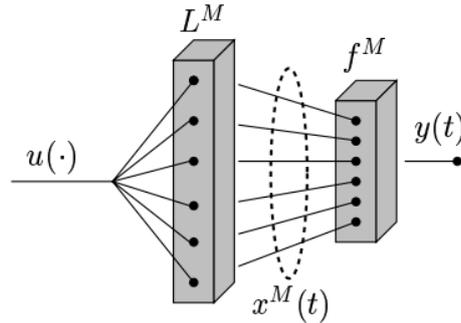


Figure 5.10: Architecture of an LSM. A function of time (time series) $u(\cdot)$ is injected as input into the liquid filter L^M , creating at time t the liquid state $x^M(t)$, which is transformed by a memoryless readout map f^M to generate an output $y(t)$ [Maass 2002].

5.8 Evolving Connectionist System (ECOS)

The ECOS is a set of principles that inspired a family of constructive artificial neural-networks first proposed in [Kasabov 1998b], on which NeuCube is also based on. It draws analogies from human brains in that it can be described as intelligence, incremental learning, and knowledge representation systems that are adaptive and evolving based on continuous incoming data [Kasabov 2009]. The main principles or requirements can be enumerated as such:

1. fast learning from a large amount of data;
2. real-time, incremental adaptation to new data;
3. an open structure, where new features (either inputs or outputs) can be added;
4. able to reasonably keep track of and retrieve data that have been seen previously;

5. continuous improvement throughout the lifetime of the system; (i.e., evolving in this context as opposed to Evolutionary Algorithms)
6. able to analyse and explain themselves through, for example, rule extraction;
7. able to represent spatial and temporal elements of data

These principles have inspired algorithms such as EFuNN [Kasabov 1998a], DENFIS [Kasabov 2007], and also SNN-based algorithms such as the Evolving SNN (eSNN) [Wysoski 2006, Wysoski 2010] and further to its latest iteration the Dynamic Evolving SNN (deSNN) [Kasabov 2013b], which is used extensively with the current implementation of NeuCube. A review of the ECOS principle, several of its implementations, and meta-study of its applications can be found in [Watts 2009].

5.9 The NeuCube Architecture

In this section, a Spatiotemporal Data Machine developed by Professor Nikola Kasabov and the team at KEDRI is explained. NeuCube is a modular architecture, integrating several different modules that each serve a purpose for spatiotemporal data analysis[Kasabov 2012, Kasabov 2014a]. As illustrated in Figure 5.11, the NeuCube mainly comprises of :

1. An input transformation and encoding module.
2. A SNN reservoir defined in 3-dimensional structure.
3. A classifier as the readout mechanism.
4. An optional Gene Regulatory Network for fine-tuning and optimisation.

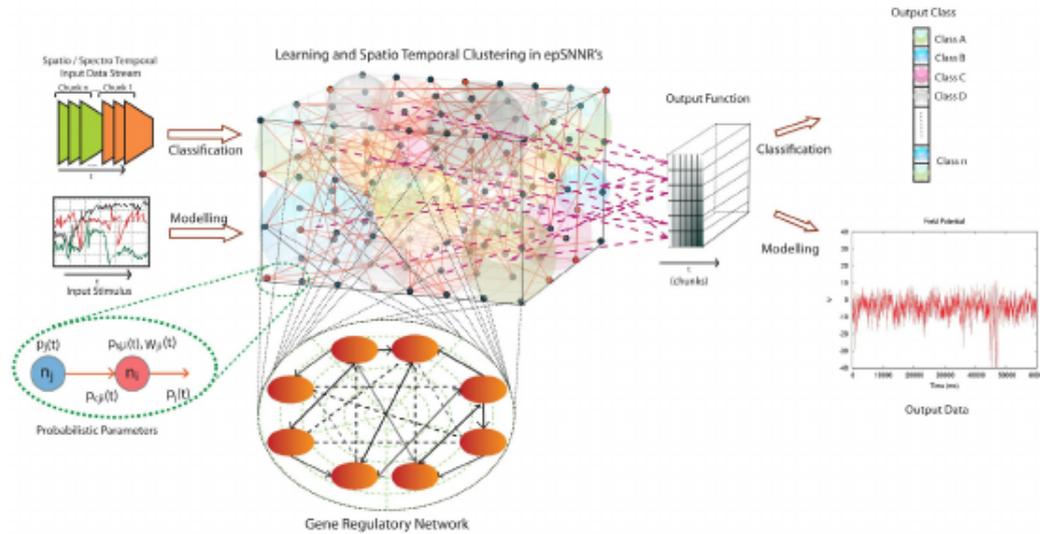


Figure 5.11: A schematic diagram of a NeuCube architecture for brain data modelling [Kasabov 2012].

The design of NeuCube is modular and different encoding algorithms can be used to obtain the spike trains from the time-series input. For example, a simple thresholding algorithm is available [Dhoble 2012], along with Ben Spiker Algorithm [Schrauwen 2003].

The three-dimensional cube is implemented with a population of LIF neurons separated by a small-world connectivity. The encoded spike trains are fed into the cube through input neurons whose spatial location have been pre-determined according to prior knowledge of the data. This is how the spatial information is encoded in the reservoir. For example, the original implementation of NeuCube follows the Talairach coordinate system with 14 input neurons that follows the spatial distribution of the channels on Emotiv EEG headsets (Figure 5.12).

NeuCube employs a two-pass learning phases:

1. Unsupervised learning: This process is where all the training data is presented to the SNN cube and the STDP learning algorithm is used to update the randomly initialised connection weights inside the reser-

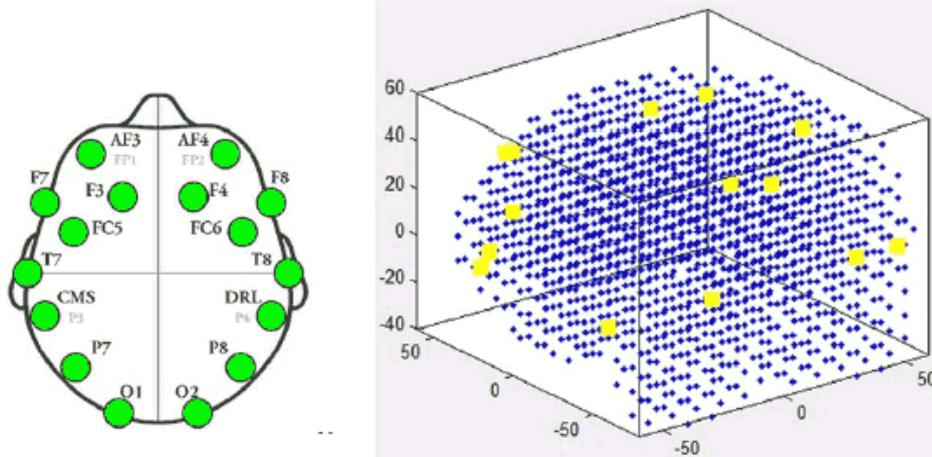


Figure 5.12: *The spatial distribution of the EEG nodes on an Emotiv EEG headset (left) and the location of the input neurons in NeuCube's reservoir network (right).*

voir according to Hebbian learning rules. The cube learns to activate the same groups of spiking neurons when similar stimuli is present in the input spike train, also known as the Polychronisation effect [Izhikevich 2006]. After learning, the final connectivity and spiking activity generated in the network can be analysed, interpreted, and visualised.

2. Supervised learning: After the cube has gone through the unsupervised training phase, the same data used in the previous phase then again propagated through the trained reservoir. The state of the cube as the data is passed through then can be learned by a classifier. In the standard implementation, the algorithm deSNN [Kasabov 2013b] can be used to evolve classifying output neurons.

Most of the computationally intensive operations can be theoretically executed on specialised hardware like the GPU and even SNN-specific microprocessors. For example, NeuCube has been implemented in PyNN that

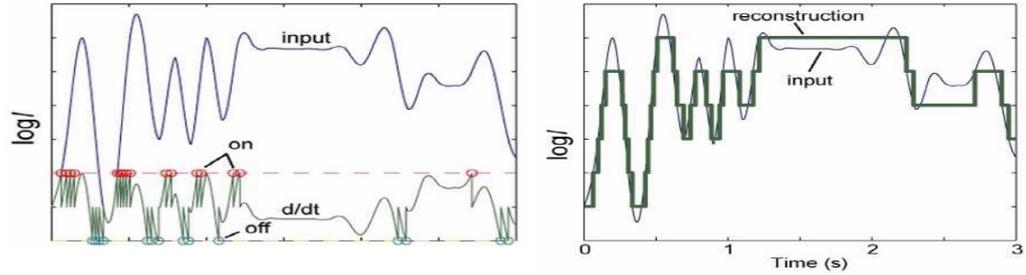


Figure 5.13: *Thresholding method of encoding continuous time-series data into a spike train (left) and the reconstruction of the signal from the spike train (right)[Kasabov 2014b].*

enabled it to run on SpiNNaker boards, an ARM-based neuromorphic hardware [Scott 2013] that uses very small energy to simulate the SNN.

5.9.1 Input Data Encoding

NeuCube works with trains of spikes, and therefore continuous value data needs to be transformed into spike trains. There exist several different algorithms whose function is to do this transformation. The one used in this research is the thresholding method, where the encoding is based on calculating the difference between two consecutive values of the same input variable over time. This makes it suitable to work with stream input data [Kasabov 2014c]. It works by performing a bi-direction thresholding of the signal gradient with respect of time d/dt . For a signal $f(t)$, the mean m and standard deviation s of the gradient df/dt is calculated, then the threshold is set to:

$$\theta = m + \alpha s \quad (5.19)$$

where α is a user-defined parameter to control the spike rate. Figure 5.13 illustrates how a spike train is obtained from a signal and how the signal can

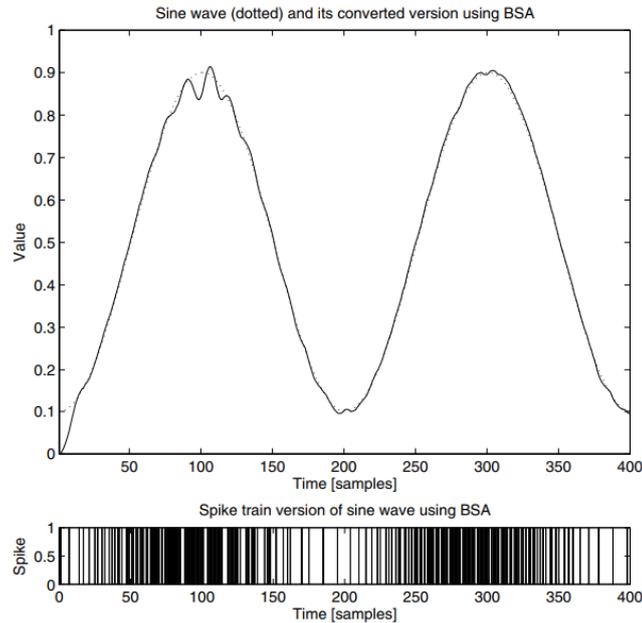


Figure 5.14: *Sine wave converted using BSA. The original sine wave is plotted in dotted line, and the converted version in solid line. The lower plot visualises the spike train. [Schrauwen 2003].*

be reconstructed back, showing that the spike train is a valid representation of the original signal.

One of other encoding methods is the Ben’s Spiker Algorithm (BSA) [Schrauwen 2003]. This method is an extension of the Hough Spiker Algorithm [Hough 1999]. Both algorithms utilise a convolution/deconvolution filter that was optimised using a genetic algorithm to minimise the error in the encoding and decoding process [Glackin 2012]. BSA has been shown to be effective for analysing EEG-based brain data [Nuntalid 2011]. Figure 5.14 gives a visual exemplification of the input and output of BSA, showing the reconstructed wave compared to the source sine wave and the spike train used to encode signal.

When BSA converts spike trains to analog values, it uses a linear filter. BSA requires that the decoding filter has a finite impulse response (FIR

filter). This is because it uses the Stimulus Estimation, which estimates the stimulus of a biological neuron by applying a linear filter. The estimate stimulus s_{est} can be written as:

$$S_{\text{est}} = (h \star x)(t) = \int_{-\infty}^{+\infty} x(t - \tau)h(\tau)d\tau = \sum_{k=1}^N h(t - t_k) \quad (5.20)$$

where the spike train of the neuron $x(t)$ is given by

$$x(t) = \sum_{k=1}^N \delta(t - t_k) \quad (5.21)$$

with t_k denoting the set of firing times of the neuron, and $h(t)$ the impulse response of a linear filter.

5.9.2 SNN Reservoir

In the heart of NeuCube is a recurrent three-dimensional SNN reservoir. The reservoir used in NeuCube can be considered a special form of LSM as discussed in Section 5.7, and is arguably the most complex component of the framework. The neurons used in this research is the leaky-integrate-and-fire (LIF) model (Section 5.4) with recurrent connection. The learning inside the reservoir is done in an unsupervised manner using the Spike-Time Dependent Plasticity (STDP) algorithm (Section 5.6). This algorithm enables a meaningful construction of connections within the reservoir, as opposed to the random, fixed one in LSM. The cube accumulates temporal information of all input spike trains and transform them into a high-dimensional intermediate states or trajectories that can be measured over time.

During the cube’s initialisation, the connectivity between neurons is constructed based on the small-world network principle. This is to reflect the biological observation in which nearby neurons are connected with a higher probability than to those who are far away [Sporns 2004].

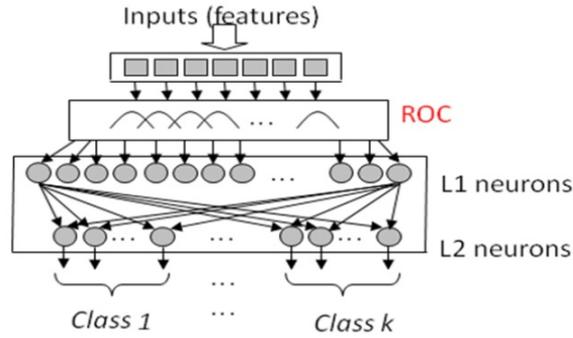


Figure 5.15: *Evolving SNN (eSNN) for classification [Kasabov 2007].*

5.9.3 Evolving Output Classification

The activity in the SNN cube when a data sample passes through it will be entered into an evolving SNN classifier which was trained in the supervised learning phase to be classified into particular cases. Several evolving SNN classifiers can be used with NeuCube. In this research the classifier used is the Dynamic Evolving SNN (deSNN) [Kasabov 2013b], which in turn was inspired by the eSNN [Wysoski 2006].

Figure 5.15 describes the schematic diagram of eSNN. It uses the Rank Order (RO) learning rule to initialise the synaptic weights based on the order of the coming spikes and then STDP for the following spikes. For every new input sample, a new neuron is dynamically allocated and connected to the input neurons. The initial connections are established based on the RO rule. These new neurons are the centroids of the clusters in regards to the synaptic weights. For each training sample, only a neuron with the highest postsynaptic potential (PSP) has its weight updated. The PSP of neuron i at time t is:

$$PSP(i, t) = \sum mod^{order(j)} w_{j,i} \quad (5.22)$$

where:

- mod is a modulation factor;
- j is the index for incoming spike;
- $w_{j,i}$ is the corresponding synaptic weight;
- $order(j)$ represents the rank of the spike arriving from all synapses to neuron i at the synapse j, i .

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

$$\Delta W_{j,i} = mod^{order(j,i(t))} \quad (5.23)$$

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

$$\Delta W_{j,i}(t) = e_j \cdot D \quad (5.24)$$

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.

After the whole training set 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 the recall mode. The threshold is calculated as a fraction (C) of the total PSP , calculated as:

$$PSP_{max} = \sum_{j=1}^m \sum_{t=1}^T (mod^{order(j,i(t))} w_{j,i(t)}) \quad (5.25)$$

$$PSP_{th} = C \cdot PSP_{max} \quad (5.26)$$

The pseudo-code to this algorithm is:

Algorithm 5.2 The eSNN training algorithm

INPUT: Spike trains, eSNN Parameters (Mod, Sim, C)

Initial neuron repository R

for each input i belonging to the same output class **do**

Create an output neuron and compute the connection weights (Eq. 5.23)

Calculate PSP_{max} (Eq. 5.25)

Calculate PSP_{th} (Eq. 5.26)

Calculate weight vectors similarity between new neurons and existent neurons

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

Merge new neuron with the most similar pre-existing neuron

else

Add the neuron to the repository

end if

end for

Instead of STDP, the deSNN algorithm [Kasabov 2013b] implement spike-dependent synaptic plasticity (SDSP) learning rule [Fusi 2000] 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 moments of simulation.

5.9.4 Applications

The NeuCube architecture has found many applications with various modes of brain data and non-brain spatiotemporal data alike [Kasabov 2016]:

- EEG Analysis: EEG data contains temporal, spatial, and spectral information that is difficult to truly explore using standard statistical or ML techniques. Though these techniques are often used to process STBD, they lack the ability to classify differences in neurological dynamics that occur over time, to identify the functional brain areas involved, and to quantify the information involved. NeuCube has successfully been used for the study of 6-channel EEG data recorded from the scalp of seven subjects performing different mental tasks [Kasabov 2015a]. This research identifies that the NeuCube is able to classify and analyse changes in functional brain activities. This is significant, as it allows for the identification of the appearance of mild cognitive impairment (MCI) to stage its degeneration towards Alzheimer's Disease (AD).
- fMRI Analysis: Recently there has been a huge interest in using functional magnetic resonance imaging (fMRI) to understand, analyse and predict behaviour and cognition. The ability of fMRI to sample high resolution spatial information over time has been successfully used in correlating high-resolution neural activity with behaviour. In contrast to statistical analysis and traditional machine learning methods, NeuCube is a rich computational model for fMRI data analysis [Doborjeh 2014]. This method can be applied to fMRI data across areas of brain study and applications [Doborjeh 2015].
- Personalised Modelling: A feasibility study on the applicability of NeuCube to make personalised models to predict the occurrence of stroke was published [Othman 2014] where the dataset was taken from Auckland Regional Community Stroke Study population.
- Ecological Modelling: The NeuCube architecture has been used to

model and predict the population of a harmful species, *Rophalosiphum padi*, in Southern New Zealand based on weather and climate factors [Tu 2014].

5.10 Conclusions

In this chapter, an STDM called NeuCube has been described. It is based on SNN, a third-generation artificial neural networks that works with spikes as its internal representation of data. With the three-dimensional reservoir trained with the STDP algorithm, it is able to learn complex patterns from various kind of spatiotemporal data. In Chapter 6, the same architecture will be used to work with spatiotemporal seismicity data.

Seismogram Analysis for Event Prediction Experiment

This chapter presents the application of NeuCube to analyse seismograph readings collected from different sites within a certain geographical area to learn and predict the occurrence of strong earthquakes. The SNN-based NeuCube architecture employs a reservoir computing paradigm capable of capturing the spatial and temporal relationships embedded in the data. Experiments on data collected from the Canterbury region of New Zealand and comparison with other computational intelligence methods indicates that this novel approach of analysing seismic data is promising and warrants further examination and research.

6.1 Introduction

Earthquake prediction is a challenging and compelling problem, especially in New Zealand. Several high-intensity earthquakes have struck highly populated regions of Canterbury and Wellington and caused a high number of casualties and loss within the last decade. The immense capacity for destruction of earthquakes prompts for the ability to predict, within a reasonable time horizon, their occurrence so that proactive actions could be taken to minimize damage. However, earthquake prediction in general remains a controversial topic and there seems to be an overly pessimistic outlook on its

success rate, especially in modern times. This is most likely a product of disappointment from a series of failed attempts at predicting earthquakes since the height of this field in the 1970s [Jordan 2006], with some researchers even going to the extent of abandoning the idea of prediction [Geller 1997]. Despite a track record of modest success in earthquake prediction, a copious amount of geological data is continuously being collected and analysed. This research will try to push the boundaries and investigate the feasibility of using machine learning approaches in predicting the incidence of strong earthquakes using the seismic time series data recorded from various seismometer sites as the precursor.

6.2 Research in Earthquake Prediction

6.2.1 Earthquake Precursors

The basic premise on which earthquake prediction techniques stand is that there are some phenomena, called precursors, which consistently occur before an earthquake. One of the most prominent approaches in this area is the measurement of anomalies in the different parts of the atmosphere which seems to change due to seismogenic effects [Pulinets 2003], for example, the temperature [Oyama 2008] and density [Liu 2001] of electrons in the ionosphere. Other approaches extend from measuring the amount of radon emissions in the soil and ground water [Ghosh 2009] and by observing the behavior of animals such as mice [Li 2009] and common toad [Grant 2010].

Lately there have been several studies that suggest that the existence of some signatures in the seismograph readings prior to the occurrence of earthquakes. The possibility of using high frequency components of micro-seismic noise readings has been studied in [Sović 2013], which reports a characteristic change one or two days before an earthquake. Another study reports

that pulsed vibrations are recorded between 5 to 10 days before earthquakes around Russia [Sobolev 2006]. Another study done with the Tottori earthquake in the year 2000 also revealed that there are seismic quiescence anomalies before the earthquake [Huang 2006], which are also observed leading to the massive Taiwan Chi-Chi earthquake in 1999 [Wu 2006b]. Based on these literatures there is a scientific basis in using the readings of seismographs as precursors for short-term prediction of strong earthquakes. The challenge is to develop methods that can learn from the patterns that are hidden in the intricate interactions between spatial and temporal components.

6.2.2 Computational Intelligence Methods for Earthquake Prediction

Despite various precursor variables having been proposed, the application of Computational Intelligence methods to deal with the problem of earthquake prediction has unfortunately been scarce.

A method employing artificial neural networks has been developed to predict earthquakes in Chile, by using the b-value, the Bath's law, and the Omuri-Utsu's law as input parameters [Reyes 2013]. This promising research built and used multiple models corresponding to the geographical regions or cities it wanted to analyse, and since classical artificial neural networks are not suitable to work with the temporal aspect of the data, it employs several fundamental geophysics laws to extract the input features from the available time-variant data. A study by the same authors using similar technique has also been done for earthquakes around the Iberian Peninsula [Morales-Esteban 2013].

Another approach using an adaptive neural fuzzy inference system (ANFIS) has also been proposed, using location of the earthquake as the input and the magnitude as the output on the assumption that the system will

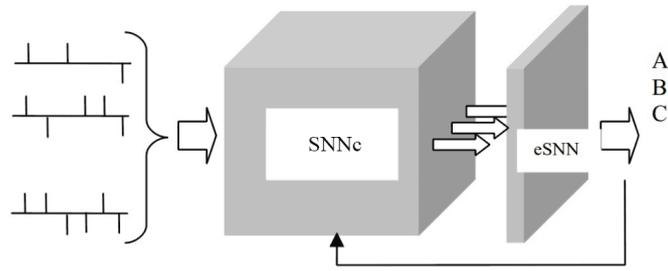


Figure 6.1: *Simplified block diagram of the NeuCube Architecture*

tune itself to model the principle of conservation of energy and momentum of annual earthquakes [Shibli 2011]. Another ANFIS-based approach was proposed by [Zamani 2013] in which historical earthquake data is mapped into two kinds of input: spatial and temporal, which are analysed separately. Yet another ANFIS based approach was proposed by [Joelianto 2008] in which the inference system is used to predict a time-series of earthquake parameters of the Sunda region in Indonesia. A rule-based system for earthquake prediction was also proposed by [Ikram 2014] which claims 100% accuracy within 15 hours, although the spatial resolution of the prediction area is low, covering areas as large as a hemisphere.

Almost all of the previous research in employing CI methods seem to extract features such as the b-value (Gutenberg-Richter law), Bath's law, Omori's law and so forth from a historical sequence of previous earthquakes in a region. None in particular proposed the use of multiple time series readings of seismic activities prior to the earthquakes to capture predictive spatiotemporal patterns. In this research, we investigate the effectiveness of a spatiotemporal modelling approach with SNN for prediction, based on the seismicity prior to the occurrence of the earthquake.

6.3 NeuCube for Spatiotemporal Data

The classifier system used in this paper is the NeuCube SNN architecture which has been proposed by Kasabov [Kasabov 2014c]. SNN are the third-generation artificial neural networks that can simulate the workings of neurons more closely to the biological counterparts. Designed initially to map and model brain signals captured in the form of EEG, the NeuCube architecture has found its way as well into several non-brain data case studies such as the prediction of stroke occurrence with weather variables [Kasabov 2014b, Kasabov 2015c].

NeuCube can be considered as a special case of a Liquid State Machine, with the input neurons mapped in the reservoir according to the spatial distribution of the actual input and a readout mechanism employing SNN-based classifiers as output neurons [Schliebs 2013]. In case of analysing brain EEG data, the input neurons correspond to the spatial location of the nodes on the scalp. For other applications, the input neurons can be mapped either automatically or according to the characteristic of the data [Kasabov 2015c]. For example, [Kasabov 2015b] mapped fMRI data to the cube based on the voxel coordinates of the images.

The basic components of this architecture is depicted in Fig. 6.1. Input data is transformed into spike trains using encoding algorithms like simple thresholding or Ben's Spiker Algorithm (BSA) introduced in [Schrauwen 2003]. These spike trains are then fed into the cube (SNNc) in an unsupervised learning procedure so that the reservoir's network can learn to activate the same groups of spiking neurons when similar spatiotemporal input stimuli are presented. After the unsupervised training phase, the same data is propagated again and output neurons are evolved to learn to classify the SNNc activity into predefined classes. Different SNN methods can be used to learn and classify spiking patterns from the SNNc, including the deSNN [Kasabov 2013b].

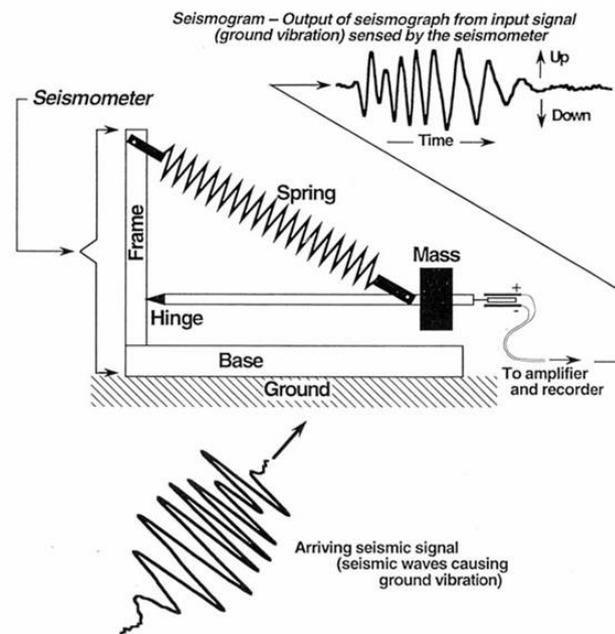


Figure 6.2: Schematic diagram of a seismometer. Ground motion causes the base and frame of the seismometer to oscillate. The mass, suspended by the spring and boom, tends to stay in one place because of inertia. The relative motion of the base as compared to the mass is recorded as the output of the seismometer. Image: Larry Braile, Purdue Uni.

6.4 The Seismometer

The spatiotemporal data that is used in this research is obtained from seismometers, which are instruments that measure motion of the ground, including those of seismic waves generated by earthquakes, volcanic eruptions, and other seismic sources. The fundamental observations used in seismology (the study of earthquakes) are recorded with seismometers in the form of seismograms - a record of the ground motion at a specific location. Figure 6.2 depicts a simple diagram that explains the working mechanism of electric seismometers.



Figure 6.3: *The New Zealand National Seismograph Network consists of primary sites located at approximately 100 km spacing, with extra regional sites established at places of geophysical significance.*

6.4.1 The GeoNet Project & The New Zealand National Seismograph Network

In 2001, GNS Science, a New Zealand government-owned research institute, received funding from the New Zealand Earthquake Commission (EQC), a New Zealand government mandated natural disasters insurer, to develop GeoNet as a non-profit 'public good' initiative to enhance New Zealand's ability to respond to and prepare for natural hazards such as earthquakes, volcanic eruptions, and tsunamis [Petersen 2011]. The GeoNet project uses a wide variety of sensing equipment located throughout New Zealand.

Seismographs are used to measure accurately the magnitude, location and other characteristics of earthquakes. The New Zealand National Seismograph Network is made up of 51 backbone stations located throughout the country and offshore islands to provide a uniform location and data collection capability. The sites consists of:

- Guralp CMG3-ESP or Streckeisen STS-2 broadband seismometers;
- Kinematics Episensor strong-motion accelerometers;
- Quanterra Q330 24-bit data loggers;
- High-speed data connections using standard internet protocols (TCP/IP) to the data centres.

These sites are spread out all across New Zealand, as depicted in Figure 6.3. This spatiotemporal nature of the data is of interest in this research. Much like how EEG nodes record the electrical activity of the neuron sensed on the scalp, these sites are recording the seismic activity of the earth captured on the surface and might contain complex spatiotemporal information that we can recover when the appropriate tools are employed.

6.5 Experiment: Very Large Earthquakes Prediction

The experiments in this research was designed to investigate whether by building a model to learn from seismometer readings preceding a seismic event, the imminence of large earthquakes can be predicted. This question can be formulated and tested as a binary classification problem of differentiating a positive class from a negative class. This experiment serves as a feasibility study to test the potential of NeuCube working with seismogram datasets.

6.5.1 Experiment Design

In this first experiment, instances in the positive class corresponded to earthquakes which are historically notable, felt by the general population in the

region and classified as *strong* or *severe* in intensity by GNS Science New Zealand as displayed in the GeoNet website (www.geonet.org.nz). GeoNet provides access to extensive data recorded by sensors belonging to the New Zealand National Seismograph Network [Petersen 2011]. As in [Reyes 2013], the location of the earthquake is considered to be known since the model was built for a specific geographical area, namely the region of Canterbury in the South Island of New Zealand in which the city of Christchurch is located. The samples were taken after the year 2010 since most of the strong and well known earthquakes in the region happened afterwards, and the data quality is more consistent in recent times. It should be noted that strong aftershocks which usually occur within a few days after a large earthquake were excluded.

For both classes, appropriate samples of earthquakes needed to be selected. The 12 events considered as the positive class are listed in Table 6.1. The small number of samples is the consequence of the fact that strong earthquakes happen very rarely throughout history, and more so in a particular region. Another 12 samples were taken from the catalogue from around the same time period and region where there were no big earthquakes and the maximum magnitude experienced in the surrounding days did not see any significant jump. These samples were the negative class, representing episodes of low overall seismic activity.

For the purpose of this study, the reading used is Seismic time-series data from the Long Period Band Type, which corresponds to a 1 Hz sample rate. The instrument code used is H, which means High Gain Seismometer. The Orientation Code is N, which means that the displacement measured is along the direction of North-South horizontal axis. Four seismic stations from the Canterbury area (McQueen's Valley, Oxford, Lake Taylor Station, and Kahutara) were selected for their proximity within and around the region of interest, and generally higher uptime. Other nearby candidate stations were excluded because significant amount of their recordings have gaps and

Table 6.1: *Earthquakes within the Canterbury region used as positive samples*

Public ID	Date	Magnitude	Depth (km)
3366146	September 3 2010	7.1	11
3450113	January 19 2011	5.1	9
3468575	February 22 2011	6.3	5
3474093	March 5 2011	5.0	10
3497857	April 16 2011	5.3	9
3505099	April 29 2011	5.2	11
3525264	June 5 2011	5.5	9
3528810	June 13 2011	5.9	9
3591999	October 9 2011	5.6	8
3631359	December 23 2011	5.8	10
2015p012816	January 5 2015	6.0	5
2015p305812	April 24 2015	6.2	52

empty values, presumably due to downtime periods. The geographical location of these stations along with the others in the New Zealand National Seismograph Network can be seen in Fig. 6.4.

Alongside NeuCube, several different classifiers were used for comparison. Traditional machine learning methods like the Multilayer perceptron (MLP) and Support Vector Machine (SVM) were included, as well as the more advanced Evolving Classifier Function (ECF) that is based on the evolving connectionist systems principle as laid out by [Kasabov 2013a]. The parameters of the other algorithms are manually optimised in a heuristic manner since the number of hyperparameters are relatively fewer than NeuCube.

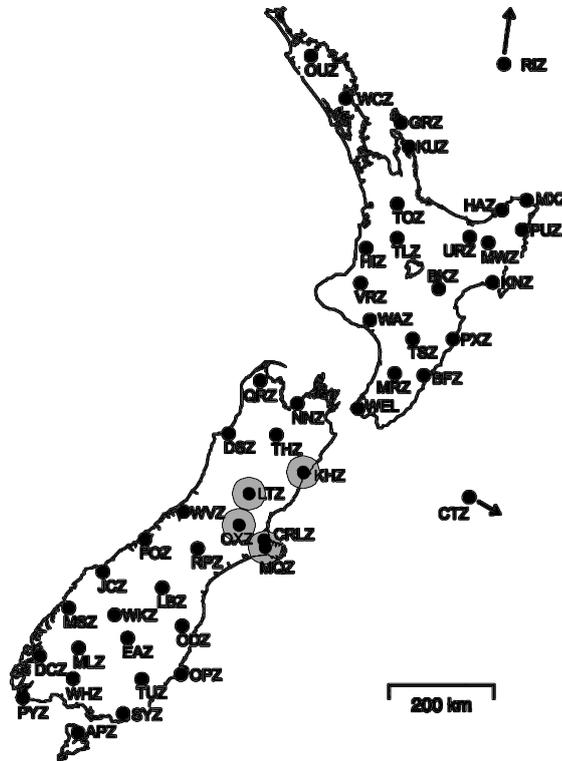


Figure 6.4: *New Zealand National Seismograph Network with the 4 selected sites around Canterbury area grayed (McQueen's Valley: MQZ, Oxford: OXZ, Lake Taylor Station: LTZ, and Kahutara: KHZ)*

6.5.2 Data Acquisition and Preparation

The seismometer readings preceding the sample earthquakes as described in Section 6.5.1 was obtained from the New Zealand GeoNet's Continuous Waveform Buffer web services. The website provides access to an immense amount of data collected since digital recording in New Zealand commenced in 1986. Waveforms in the GeoNet CWB (Continuous Waveform Buffer) data repository can be accessed using a Java client provided for the public to download. The client accesses the server and downloads files in MiniSEED, SAC or plain text formats. In this research we used the default file type

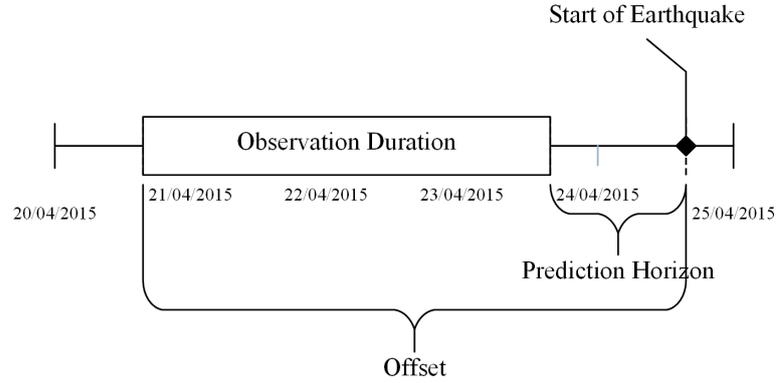


Figure 6.5: *This timeline illustrates the relationship between the actual time the earthquake starts, the offset, and the observation duration used in this experiment.*

Seismic Analysis Code (SAC), which is a general purpose interactive program designed for the study of sequential signals, especially time series data [Tapley 1992].

To predict ahead an actual event, data needs to be offset by a certain amount of time. The duration of the observation also needs to be chosen, which in turn will determine the length of the prediction horizon. This arrangement is depicted in Fig. 6.5. In this experiment, the effect of varying the prediction horizon on classification accuracy is analysed. For the purpose of this study, the observation duration length is fixed to 5 days (120 hours).

After the raw data is obtained, simple preprocessing steps needed to be applied to prepare the data to be fed to the models. The input data I of a sample for this earthquake prediction problem is defined as L_1, L_2, \dots, L_s where s is the number of seismic sites which are taken into account. Each vector $L \in I$ is a time-series $L = a_{-t-d}, a_{-t-d+1}, \dots, a_{-t}$ in which the values are a chronologically ordered set of d real-valued variables, d being the duration of observation and t the prediction horizon, i.e. the time before the earthquake occurs, and assuming a_1 is the value at the occurrence of the earthquake.

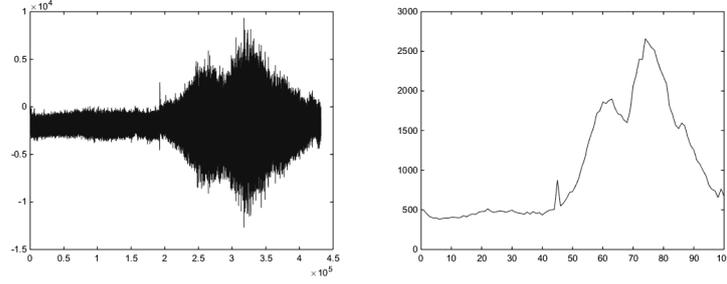


Figure 6.6: *Raw seismogram (left), and after pre-processing (right). X-axis is the discrete time unit and Y-axis is the seismicity readings.*

Since the seismograph reading is high-resolution spanning over a long period of time, the standard deviation of the signal is computed in a piecewise manner in order to reduce the length and dimensionality of the time series. So I is now $\bar{I} = \{\bar{L}_1, \bar{L}_2, \dots, \bar{L}_s\}$, in which the i^{th} element of time series \bar{L} of length w is calculated as:

$$\bar{a}_i = \sqrt{\frac{w}{d} \sum_{j=\frac{d}{w}(i-1)+1}^{\frac{d}{w}i} (a_j - \mu)^2} \quad (6.1)$$

where

$$\mu = \frac{w}{d} \sum_{k=\frac{d}{w}(i-1)+1}^{\frac{d}{w}i} a_k \quad (6.2)$$

This dimensionality and noise reduction step is illustrated in Fig. 6.6, showing that after the transformation, the resulting time series gives a clear indication of the strength of the seismic activity over a certain period of time. While the other classifiers require the signals to be flattened out of the into one feature vector, they can be directly fed into NeuCube, which will discretise the signals into spike trains as shown in Fig. 6.7. The spike encoding method used in this experiment is the Thresholding-based temporal difference encoding method. In short, some threshold δ is defined such that we calculate an output spike $s(t)$ as:

$$s(t) = \begin{cases} + & \text{if } i(t) \geq \delta + i(t_{last}) \\ - & \text{if } i(t) \leq \delta + i(t_{last}) \end{cases} \quad (6.3)$$

Where $i(t)$ is the input value at the current time and $i(t_{last})$ the previous one. $+$ and $-$ represent excitatory and inhibitory spikes respectively. This method is conceptually similar to the one used in Dynamic Vision Sensor (DVS) cameras and is argued to be biologically plausible [Kuang 2012].

6.5.3 Sampling Bias

The presence of biases limits the generalisability of any learning method trained on a specific dataset [Tommasi 2017]. While it remains an open question whether creating an unbiased dataset is possible given limited resources, it is essential to be aware of the biases that might manifest in the dataset that is used in this research.

The *capture bias* is related to how the waveforms were acquired both in terms of the condition of the sensors that were used and of the variation in external circumstances (weather, non-seismic noises) at the time of recording. The *category* or *label bias* comes from the fact that determining an earthquake's severity category is not as simple as binning its moment magnitude reading: it has a lot to do also with the depth of the epicentre, duration of the shock, geological condition of the surrounding areas, slip geometry, and whether one is a main shock or an aftershock [Purcaru 1982]. The binary labelling of *positives* and *negatives* in this research could potentially be inadequate in capturing the subtle technical differences between earthquakes.

Additionally, the dataset may contain only a distinct set of shocks and this causes the *negative bias*. To put it in another way, the model trains only from the classes shared among them as opposed to the universal set of earthquake types which were not considered in this research. Moreover, the

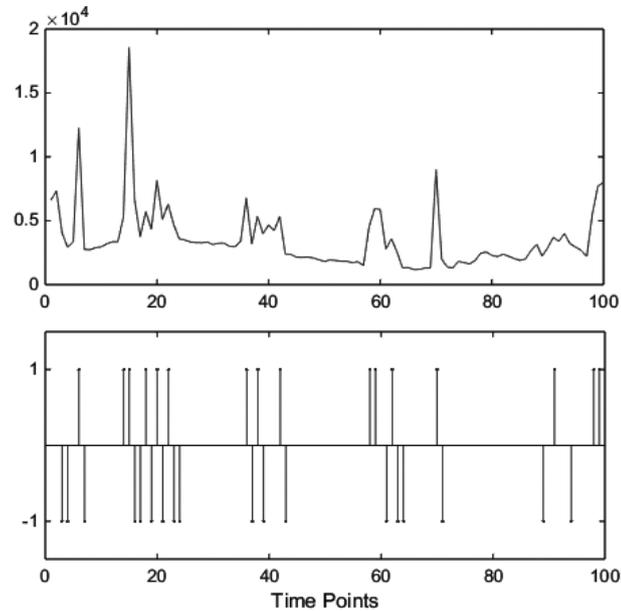


Figure 6.7: *Preprocessed seismogram and the resulting spike train*

discrete spatial region-of-interest selection boundary might also introduce the *exclusion bias* to the dataset. In other words, earthquakes outside but close to the perimeter were excluded, although they might have affected the sensors' readings and thus are excellent sample candidates. Finally, the dataset is balanced to yield equal amount of positive and negative samples. This is a special form of *selection bias*, in which proper randomisation is not achieved. In reality, the periods of quiescence vastly outnumber the earthquake shocks.

Indeed, the problem of *dataset bias* in this research originates from the complications non-exhaustively listed above; the limited nature of the dataset, curated to train and evaluate learning models, might lead to false conclusions. This predicament and its countermeasures go beyond this research and it is a long-standing problem in machine learning.

6.5.4 Results

The experiment was carried out by running the data through the different classifiers and varying the length of the prediction horizon. The parameters for each classifier were tuned heuristically to obtain the best results from each of them. In addition to the accuracy, the performance of the classifiers is measured in terms of the balanced F-score on the positive earthquake class, which is the harmonic mean of Precision and Recall of binary classification problems and can be formulated as: $F_1 = 2TP/(2TP + FP + FN)$. This additional measure is important, since overall accuracy alone does not reveal the actual performance within each of the classes which is of interest in a binary classification problem. Since the number of samples is small, the training/validation scheme used is leave-one-out cross validation [Kasabov 2013a].

The result of the experiment is laid out in Table 6.2. As expected, shorter prediction horizons produced better prognosis. The effect of this prediction horizon to the accuracy of these different models is visualized in Figure 6.9. It should be noted that in a balanced binary classification problem, there is a baseline accuracy of 50%, which can be achieved statistically by random guessing or giving the same answer to all the cases. The best prediction accuracy obtained with the NeuCube model successfully predicted 11 out of 12 strong earthquakes and raised only 1 false alarm, 1 hour prior to the actual event, which is indeed promising. It is interesting to note that no models were able to differentiate between the two classes 48 hours ahead of an earthquake event, suggesting the possibility a certain temporal limit to the prediction horizon in this particular experiment.

6.5.4.1 Experimental Rigor

In total, there were only 24 examples in the data set. This naturally raises some concerns about whether this is large enough for a relatively complex

Table 6.2: *Classification accuracy result with varying prediction horizon length*

Classifier		1 hour	6 hour	24 hour	48 hour
MLP	Accuracy	58.33%	54.16%	41.66%	41.66%
	F-Score	0.58	0.52	0.41	0.41
	TP Rate	0.58	0.50	0.41	0.41
	FP Rate	0.41	0.41	0.58	0.58
SVM	Accuracy	54.16%	50%	37.5%	37.5%
	F-Score	0.58	0.52	0.41	0.41
	TP Rate	0.58	0.50	0.41	0.41
	FP Rate	0.41	0.41	0.58	0.58
ECF	Accuracy	70.83%	66.67%	66.67%	50%
	F-Score	0.63	0.60	0.66	0.64
	TP Rate	0.50	0.50	0.66	0.91
	FP Rate	0.04	0.16	0.33	0.91
NeuCube	Accuracy	91.67%	83.33%	70.83%	54.17%
	F-Score	0.91	0.80	0.72	0.42
	TP Rate	0.91	0.83	0.75	0.33
	FP Rate	0.08	0.25	0.25	0.25

algorithm like NeuCube. This is unfortunately one of the natural limitations with large earthquakes studies. Moving backwards in time is also not a possibility since the sensors used were only available recently. It should be noted that this is a highly-localised feasibility study and should not be interpreted as claiming a general capability of short-term earthquake prediction.

As a supplement, another way of confirming that the algorithms are learning the two patterns is proposed. Table 6.3 shows an interesting perspective on how the classifiers would work if the input is the stream data of the seis-

Table 6.3: *Classification output results with stream data from 2010-2015*

Classifier	Output	Actual Data	Random Data
MLP	Neg	34553	27106
	Pos	17881	25335
	Ratio	0.58	0.50
SVM	Neg	34040	28597
	Pos	18400	23844
	Ratio	0.65	0.54
ECF	Neg	34058	29185
	Pos	18382	23256
	Ratio	0.64	0.55
NeuCube	Neg	42135	10426
	Pos	10426	24043
	Ratio	0.80	0.54

mographs taken from 2010 to 2015 (around 52000 hours in total). A random stream data with the same statistical property as the actual testing data was generated and streamed to the classifiers to see their output vectors. Trained in a balanced manner, the probability of producing either output (positive or negative) is equal and this is apparent by observing the 'Random Data' column on Table 6.3. Using the random stream data, all the trained classifiers produced outputs of the expected 0.5 ratio. However, the ratios were different when using the actual data. Most of the classifiers will noticeably produce more 'negative' decisions despite trained symmetrically, reflecting the fact that seismic events are much rarer than periods of low seismicity. Although this result is not directly usable in a practical sense, it gave a promising indication that the two patterns were able to be learned and differentiated by machine learning algorithms and NeuCube was found

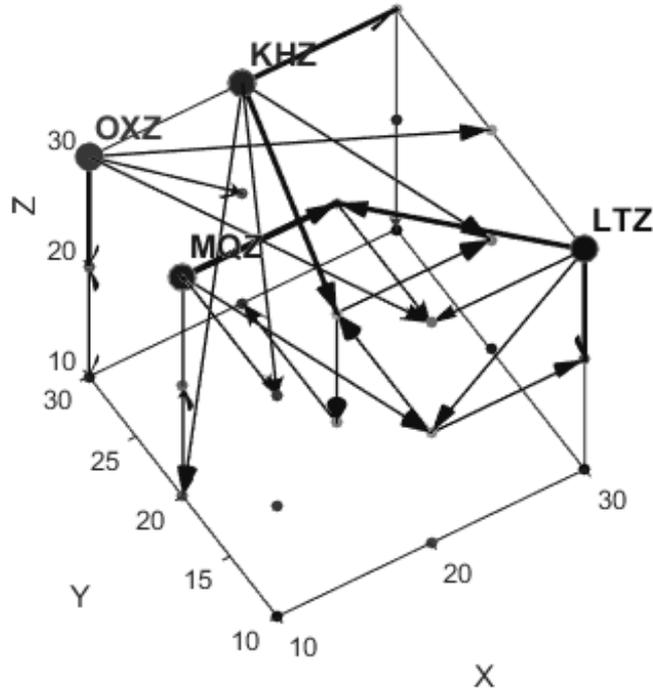


Figure 6.8: *SNN reservoir with input neurons and synapses after training. The line thickness and arrow signify the connection weight and direction respectively, which are obtained by applying STDP unsupervised learning with the training input data. The more spikes transmitted between two neurons the stronger they are connected to each other.*

to be the most promising tool for the job.

These findings also lend further support to previous studies mentioned and discussed in Sect. 6.2.1 which suggested that there are certain patterns exhibited by seismicity readings that can be used to predict the imminence of large earthquakes. The results shown in Table 6.2 and 6.3 seem to indicate that seismicity data is a viable precursor for short-term earthquake prediction. The connectivity of the $3 \times 3 \times 3$ cube after training is depicted in Fig. 6.8. The amount of neurons in the reservoir is notably more than the number of samples and raises the concern of over-fitting the training sam-

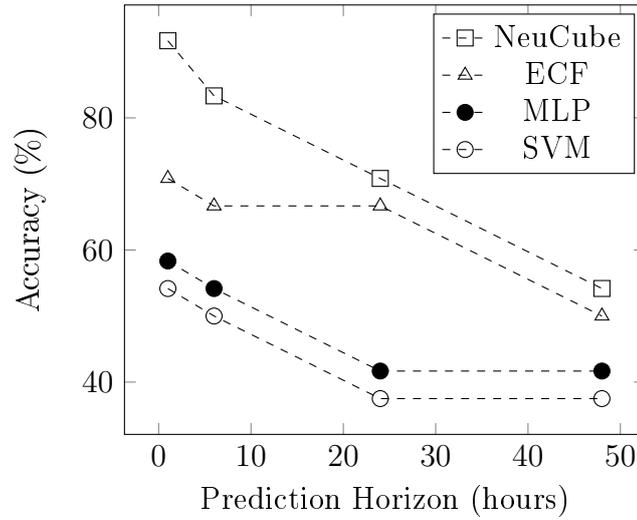


Figure 6.9: *The effect of the prediction horizon to the accuracy of different classifiers. Performance deteriorates the further away readings are taken from the events.*

ples. However, it has to be noted that the reservoir acts as a map to higher dimension feature space and is not a classifier on its own.

6.5.5 Analysis

The NeuCube architecture is devised and implemented with interpretability of the model in mind. The connection weights and the spiking activity in the reservoir can be used to visualise the spatiotemporal patterns and relationships between input variables learned from the dataset. From the performance that were obtained, we have the parameters of the model from which knowledge and further analysis can be gathered. Further investigation into the model may reveal knowledge about the dynamics of and the precursors to the earthquakes.

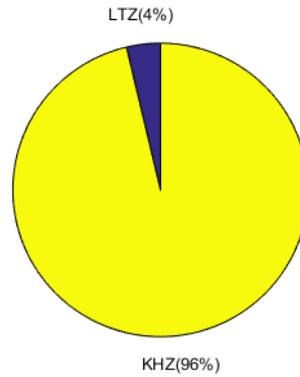


Figure 6.10: *Neuron proportion based on spike communication clustering for negative samples.*

6.5.5.1 Network Analysis

The input neurons are the information sources of the SNN reservoir and we can determine the neuron clusters belonging to each inputs based on the spike transmission in the reservoir [Othman 2015]. As the result of the unsupervised training phase with STDP rules, the more spikes transmitted between two neurons the stronger they will be connected to each other.

In this section, the connections that are made and the spiking activity inside the reservoir between the positive and negative classes will be compared to see the difference in the patterns that are developing prior to large earthquakes and periods of low seismicity.

Surprisingly, with the negative samples spiking activities are dominated by the neurons that are clustered as belonging to the KHZ input, located in Kaikoura, a town situated on the east coast of South Island 180 km north of Christchurch (Fig. 6.10). It has been noted that the seismicity of the Kaikoura region is considerably high [Van Dissen 1991]. This finding suggests that in the low-seismicity periods, most of the seismic activity in the region comes mostly from the Kaikoura fault lines and Lake Taylor (LTZ).

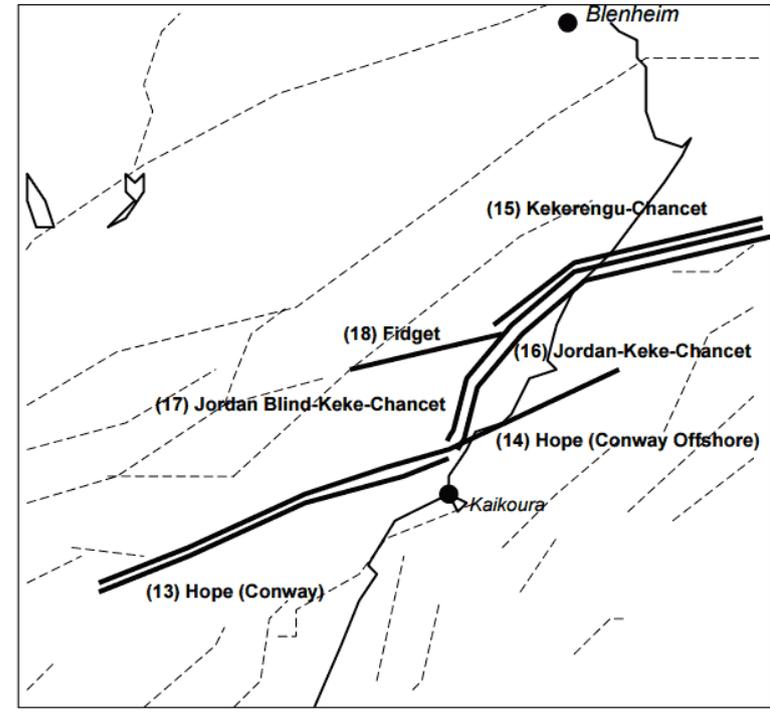


Figure 6.11: *The new active fault sources of northeastern Canterbury developed in the new model by Russ Van Dissen of GNS Science. Overlapping sources are shown side-by-side to distinguish them from one-another [Stirling 2008].*

Furthermore, Figure 6.12 showed that although the spiking activity of LTZ neurons are notable, it was not temporally correlated with the other inputs. In later reports, Kaikoura has been designated as one of the regions that has been seeing significant increase in seismic activity and hazard due to having multiple major fault lines clumped together in a very small area [Stirling 2008]. Figure 6.11 shows the active fault lines that stretch over the north of Kaikoura.

In contrast to what was happening in low-seismicity period, Figure 6.13 showed that during the high-seismicity periods, there are more interactions between LTZ and other sites. This suggests that periods with higher prob-

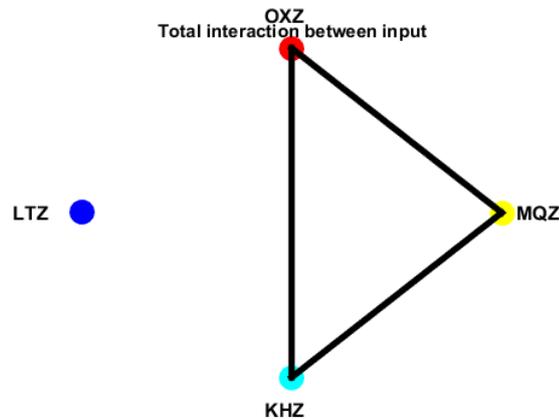


Figure 6.12: *Total interaction between the input neurons for negative samples. LTZ is not temporally correlated with the other inputs in periods of low seismicity.*

ability of big earthquakes could be indicated by not just stronger activity coming from LTZ (Fig. 6.14) but also if these activities are temporally correlated with other stations.

From these figures, observations can be made that:

- The seismic activity of the Kaikoura region (KHZ) is the strongest compared to the other 3 sites.
- Prior to large earthquakes, the activity in the Lake Taylor station (LTZ) increases and are more temporally correlated with the other sites.
- Although the seismicity of OXZ and MQZ are much lower, they are well correlated with each other and KHZ.

As an additional consideration, Figure 6.15 shows the Canterbury region divided into several structural domains, each distinct in terms of neotectonic setting, style, geometry, and rates of deformation [Stirling 2001]. Domain 7,

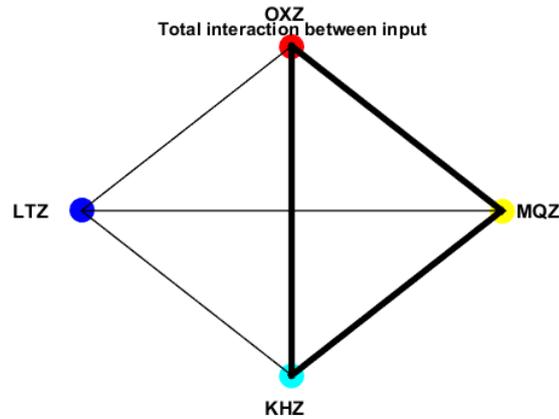


Figure 6.13: *Total interaction between the input neurons for positive samples.*

or Canterbury Plains Zone on which the sites OXZ and MQZ are stationed, has the lowest rates of deformation in the region due to having the furthest distance from plate boundary. This information supplemented with the observations in the NeuCube reservoir, could indicate that when it comes to figuring out the earthquakes in the Canterbury region, the seismicity of Domain 2,3 & 4 (where KHZ and LTZ are) holds more relevant information due to being closer to the Australian and Pacific plate boundary.

6.5.6 Individual Earthquake Analysis

In the previous section, a way of interpreting the learned patterns have been presented. In this section, we will observe the individual samples by using a spike raster graph. A spike raster plot displays the spiking activity of a group of neurons over time. In this thesis a raster plot each row (y-axis) corresponds to the index of a neuron in a neuron group. The columns (x-axis) corresponds to the time in the simulation.

Two samples from the positive group were taken and compared to one

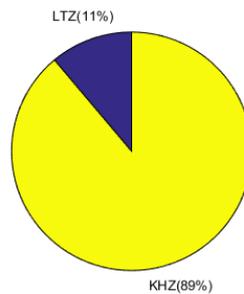


Figure 6.14: *Neuron proportion based on spike communication clustering for positive samples.*

sample from the negative group. The first example is the infamous 2010 Darfield earthquake, which was 7.1 magnitude strong. Figure 6.16 shows an increase in activity of LTZ, consistent with the findings from previous section. Moreover, there is a discernible synchronicity between the three of the input channels (except for MQZ). This could be interpreted that before large earthquakes, there would be an increase in seismicity over a large area, picked up by most of the sites at the same times. This is also consistent with the increased interaction and similarity between features observed in positive samples from Section 6.5.5.1. This phenomenon can be seen more clearly in Figure 6.17, which is the raster plot for the 5.1 magnitude earthquake on June 5, 2011. Not only the input channels spike almost in a uniform and synchronised manner, the spike rasters for both positive example also exhibit a wave-like spike propagation in the reservoir over the course of time in respect to how the input neurons fire together. As a comparison, an example from a low seismicity period is shown in Figure 6.18. In this particular instance, the input channels are spiking independent of each other and the spike activity across the cube does not form a wave propagation pattern as compact as the others. This could be interpreted as an absence of an increase in seismicity over a large area, the existence of which could be a potential

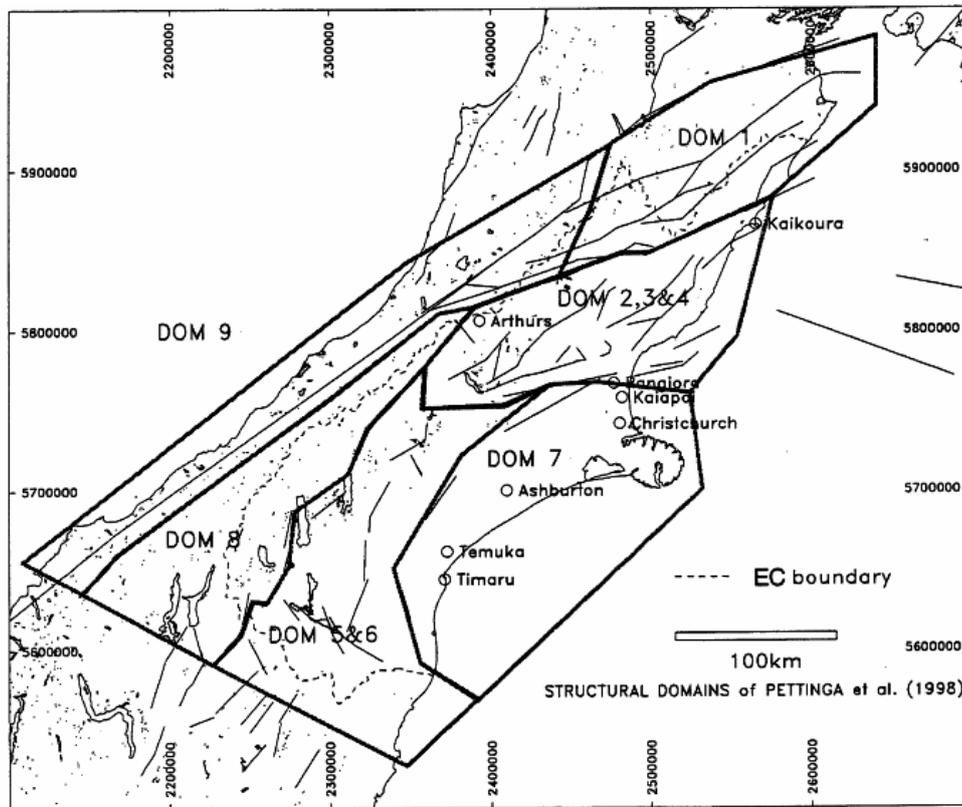


Figure 6.15: *Structural domains of the Canterbury region. The domains are as follows: DOM 1=Marlborough Fault Zone; DOM 2, 3 & 4=West Culverden Fault Zone, Porters Pass-Amberley Fault Zone & North Canterbury Fold & Thrust Belt; DOM 5&6=Mt Hutt-Mt Peel Fault Zone & South Canterbury Zone; DOM 7=Canterbury Plains Zone; DOM 8=Southern Alps Zone; and DOM 9=Alpine Fault Zone. [Stirling 2001].*

precursor to the imminence of very large earthquakes. A similar phenomenon has been previously observed and proposed as one of the precursors to large events, namely the 'Earthquake chains' [Shebalin 2006], which are clusters of moderate-size earthquakes which extend over large distances and are formed by statistically rare pairs of events that are close in space and time.

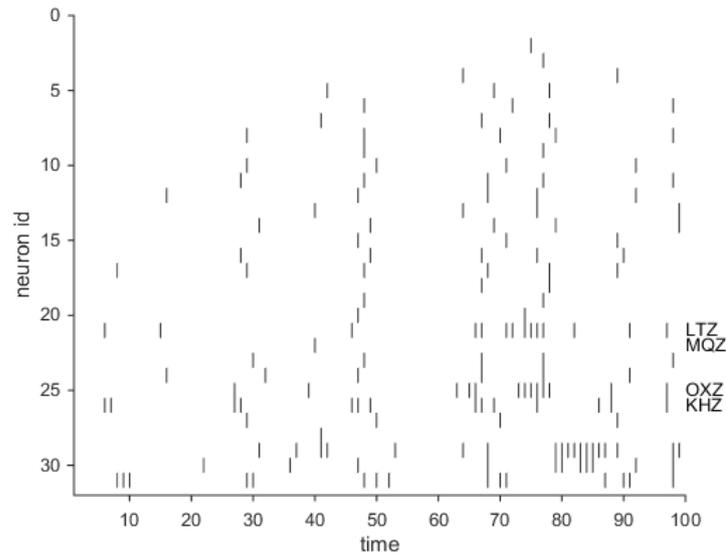


Figure 6.16: *Spike raster plot of the activity in the reservoir 5 days before the September 2010 Darfield earthquake. Notice the increase of activity on the LTZ channel prior to the earthquake, and the spiking activity of the input neurons are becoming highly correlated to each other.*

6.6 Visualisation

The application of NeuCube for seismic data analysis is not limited to predictive modelling. As part of this doctoral study, a visualisation tool for NeuCube called NeuVis originally developed by Dr Stefan Marks at AUT Colab [Marks 2014] has been adapted to work with seismic data as well. NeuVis is an immersive virtual reality (VR) visualisation environment that is designed around the use of consumer VR headsets in an existing wide area motion capture suite to display scientific or engineering data.

The motivation of this visualisation is to explore the potential of NeuCube to be used as a generic method to visualise spatiotemporal data and its dynamics in an interactive and immersive way. Different from the rest of the experiment where a small cube is used and the input channels are limited

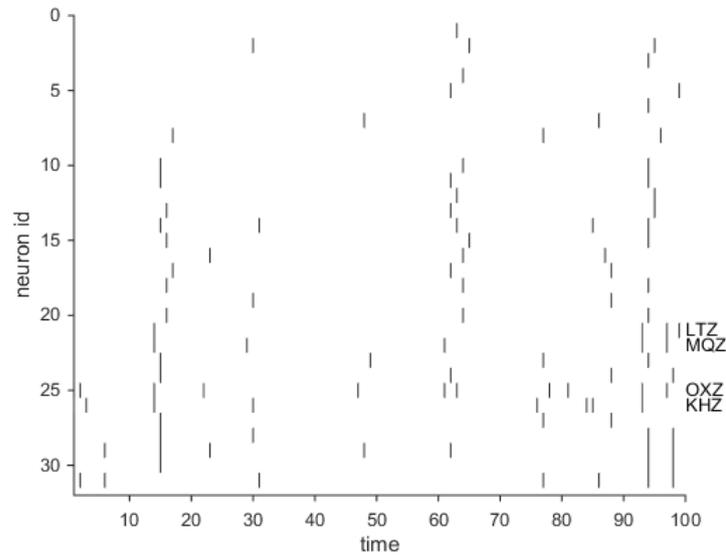


Figure 6.17: *Spike raster plot of the activity in the reservoir 5 days before the earthquake in June 2011. Although there is no increase in seismic activity, LTZ channel became highly correlated with the other stations, spiking in sync with MQZ and KHZ.*

to four sites, in this section the cube is $20 \times 20 \times 3$ neurons large and all 52 channels that are available from the New Zealand Seismograph Network are used.

Figure 6.19 is a screenshot taken from a version of NeuVis that has been adapted to work with seismic data. It presents the SNN reservoir in a 3D Virtual Reality environment on top of a map of New Zealand, giving users an interactive view of the structure of the cube and its activities.

The NeuVis can display various kinds of visualisation of the cube. For example, Figure 6.20 shows the spiking activity inside the cube above a certain threshold during a particular time. This tool provides a way for experts to visualise the seismic activities on a certain area both in real-time or back and forward in time, giving the users the power to choose what

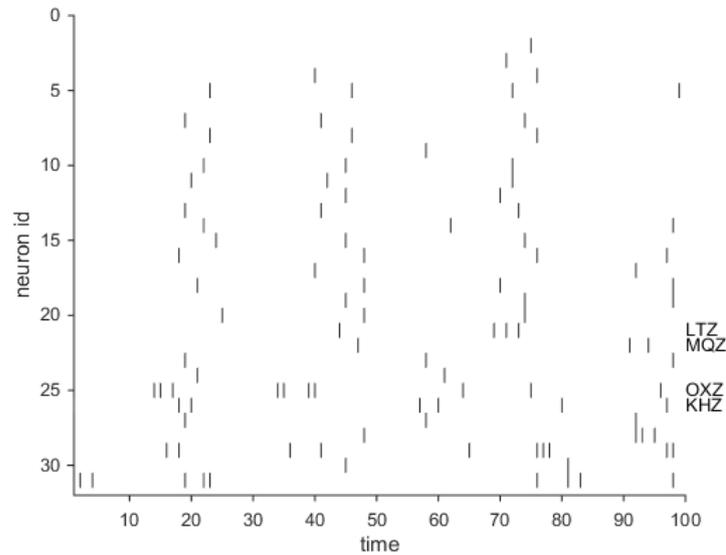


Figure 6.18: *Spike raster plot taken from a sample of low-seismicity period.*

phenomena they want to observe and in a way that is easier to understand compared to inspecting seismogram from different locations and time.

6.7 Conclusions

This chapter has shown a novel and promising way to analyse spatiotemporal seismic data by trying to predict the occurrence of strong earthquakes through training a model to differentiate between strong and weak earthquakes based on spatiotemporal seismicity precursors. This research also showed that SNN has the potential to be successfully used for early and accurate prediction of hazardous events. The capability of a more advanced SNN-based method like NeuCube to capture complex spatiotemporal signal has been demonstrated, in relation to traditional techniques like MLP and SVM. Several ways of interpreting the model have also been shown, including 3D Virtual Reality environment to visualise the cube activity.

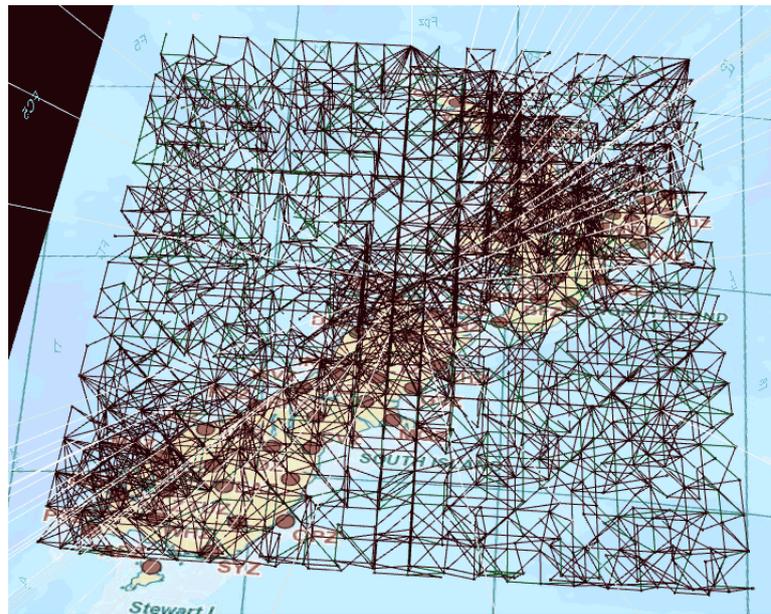


Figure 6.19: *NeuCube SNN reservoir rendered in a 3D Virtual Reality environment on top of a map of New Zealand, enabling users to immerse themselves and walk around the neurons and observe the connection building and spiking activity in time and space.*

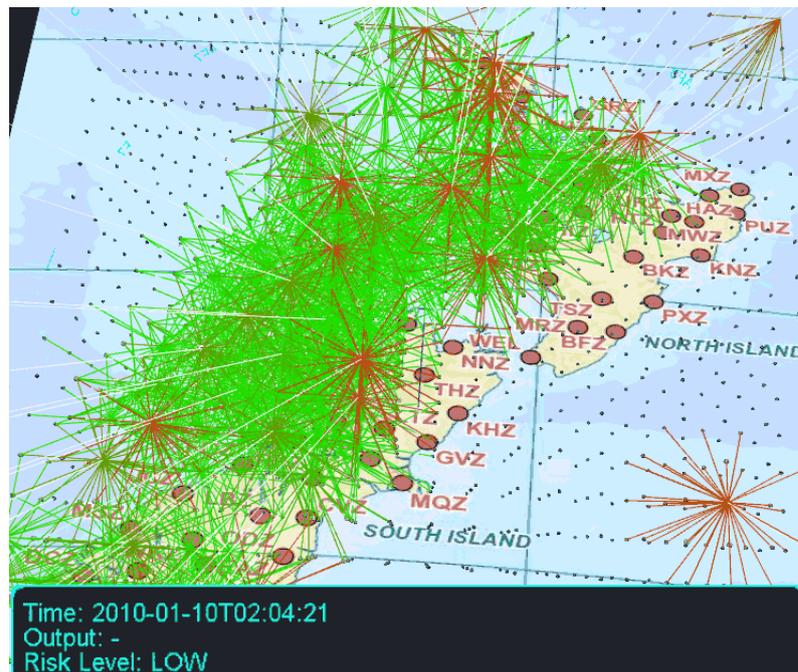


Figure 6.20: *NeuCube SNN reservoir rendered in a 3D Virtual Reality environment on top of a map of New Zealand, showing the spiking activity of the cube.*

Conclusions

7.1 Summary

In this thesis three different methods that tackle various types of data and employ distinct machine learning and data mining approaches to tackle common problems faced in spatiotemporal data mining have been presented. Each of the method is summarised in the following sections.

7.2 Rule Mining from Multiple Time-series

Discussed in detail in Chapter 2, this method aims to generate temporal association rules from multiple time-series. Based on the concept of association rule mining, the main component of the framework is a modified Apriori algorithm that was expanded to also work with time dimension and multiple time-series. The input time-series data is transformed to symbolic representation by discretisation.

As part of the research, a binary discretisation technique is also proposed. It is a simple sliding window algorithm that works by having two segments of a particular size starting from the beginning of the series which slide across the series incrementally, employing a concept-drift detection algorithm. The proposed method is adaptive to the data stream and produces better binning for data with Poisson distribution.

An application of the proposed framework is demonstrated on a dataset

that tracks the number of aphids caught in traps along with the weather variables over almost twenty years in the Lincoln region. The dataset tracks the weekly number of aphids caught in a suction trap along with weather variables like the cumulative weekly rainfall, wind run, average air temperature, potential deficit, Penman potential evaporation, and solar radiation. The experiment focuses on whether the proposed framework and methods could discover interesting rules from the multiple time series that can act as a precursor of aphid infestation. The results of this experiment have shown that the rules generated are useful in identifying high levels of aphid infestation. These temporal rules are actionable and are descriptive of the relationships between variables that also take time information into account.

7.2.1 Future Research Directions

This work could be extended in various ways and on many aspects. The use of fuzzy representations and rules is promising, since it reflects better how the rules are represented in human language and concepts, as has been argued in [Chen 2012]. Building a classifier which is able to employ the mined rules to improve long-term prediction is also a possible extension. A class of algorithm that uses association rules to produce classifications is discussed in [Balaji 2013]. Decomposing the time-series into trend and seasonal decompositions is also a promising way to pre-process the data, as it could better reveal the interactions between the variables. Rule extension that encode spatial information in the rules is also an important enhancement.

7.3 Adaptive Local Models

The second method deals with building local models for a time-step ahead spatiotemporal prediction problem. To construct these local spaces of con-

cepts, the problem is treated as an exercise of detecting concept drifts in space and time. Traditionally, global models are employed with the implicit assumption that there is no variability of data characteristics as a function of space. By making a specific adaptation to handle the problem of spatiotemporal nonstationarity, we can improve the performance of various machine learning solutions.

The Hilbert curve is a type of space-filling curve whose locality-preserving property is well studied. Taking advantage of this attribute of Hilbert curve by using it as an inter-dimensional mapping scheme, the 2-dimensional spatial data can be untangled and is able to work with existing concept drift detection algorithms. Every time a change is detected, local regions are identified and models are built to learn them. As an additional step, a multivariate statistical test can be used to check the repository for recurring concepts and reuse old models.

The framework was tested on a dataset curated from the earthquake catalogue data around the Christchurch region. The experiment confirmed that superior predictive capability can be attained with local models compared to global approach. A further analysis of the system's behaviour is also discussed.

7.3.1 Future Research Directions

One of the limitations of this research is the lack of testing on synthetically generated data. Curating a synthetic dataset in which the ground-truth regarding the concepts and the changes are known will provide a better insight and validation to the adaptability of the method. Another limitation is the repository management aspect. In handling recurring concepts, searching the repository for previously learned models is a computationally expensive operation and grows linearly as the number of model learned increases.

An improvement in the way the local model information is used can significantly increase the performance of the system. This can be carried out by having a meta-learner or an ensemble weighting mechanism instead of building a model for each local spaces. For example, in [Sakthithasan 2015] an ensemble approach was used to aggregate learned concepts stored in the form of Fourier spectra.

The mapping of 2-dimensional data into 1-dimension so it can work with existing concept-drift detectors poses big limitations. This approach necessitates that the dataset is in discrete input space and there are so many scenarios where the traversal produces suboptimal partitioning. In terms of developing a more general method for spatiotemporal data mining with concept drifts, the development of a true multi-dimensional concept-drift detector is necessary. For example, the use of clustering to identify local regions of concepts based on grid density has been explored [Sethi 2016, Lee 2012]. However, using density as the measure of change can be misleading and the sensitivity of the system relies on a manually-set threshold value.

7.4 NeuCube for Seismic Data Analysis

The last method employs a Spiking Neural Network (SNN)-based system called NeuCube to build an event prediction system. SNN is considered the next-generation model of neural networks. One of the main difference in the mechanics of SNN is the inherent concept of time. NeuCube transforms spatiotemporal data into trains of spikes and feeds them into a reservoir of spiking neurons which functions as a mapping to a higher-dimensional feature space.

As a feasibility study, the system was trained to differentiate seismicity readings obtained from four spatially scattered seismograms before large earthquakes happen vs. periods of low seismicity. The region and time pe-

riod of interest were around the city of Christchurch from 2010. After heuristic parameter optimisation, a model which perform comparatively well with leave-one-out cross validation was obtained. A further analysis of the system revealed interesting insight to the dataset. The experiment results showed that seismicity readings could be a viable precursor of large earthquakes and NeuCube proved to be a promising tool for such analysis.

7.4.1 Future Research Directions

This preliminary study has several challenges that should be overcome to extend it into a systematic research. The main problem with very large earthquake prediction is the scarcity of samples, thus limiting the ability of the model to be validated. A potential workaround to this problem is to increase the sample count by lowering the criteria of selection for the positive samples at the cost of also including possibly irrelevant smaller earthquakes. Building additional local models for other earthquake prone geographical regions such as Japan, California, Indonesia and Chile is also a possible way to further test the proposed methodology. An interesting problem to take into account in earthquake sampling is the additional properties of the earthquakes such as depth, magnitude, and the exact location of epicenters.

To supplement the seismicity data, spatiotemporal information from other sensors can be incorporated. This includes but not limited to satellite remote sensing technologies such as measuring disturbances in the total electron content of the ionosphere [Rhoades 2015], geodetic and GPS sensors, and groundwater Radon content [Planinić 2004]. This can be done by building multiple NeuCube reservoirs for different types of stream data. In the future, a framework to interpret and enable NeuCube to work with multimodal spatiotemporal stream data will be needed.

A further interesting aspect would be the extraction of knowledge in the

form of symbolic and human understandable rules of the spatiotemporal patterns exhibited by seismicity readings in regards to the occurrence of earthquakes and our knowledge about the underlying mechanism of these seismic activities. This spatiotemporal patterns are captured in the connections made in the reservoir, and a formal methodology to mine this knowledge would be a great addition to the capability of NeuCube. An excellent platform to build this idea upon is to induce and represent knowledge from SNN using Finite State Machines [Natschläger 2002]. This methodology is also a promising line of research to be extended for the prediction and analysis of other disastrous events like tsunami and land slides.

Bibliography

- [Abe 2005] Hidenao Abe, Miho Ohsaki, Hideto Yokoi and Takahira Yamaguchi. *Implementing an integrated time-series data mining environment based on temporal pattern extraction methods: a case study of an interferon therapy risk mining for chronic hepatitis*. In Annual Conference of the Japanese Society for Artificial Intelligence, pages 425–435. Springer, 2005. 16
- [Abel 1990] David J Abel and David M Mark. *A comparative analysis of some two-dimensional orderings*. International Journal of Geographical Information System, vol. 4, no. 1, pages 21–31, 1990. 45
- [Adepeju 2016] Monsuru Adepeju, Gabriel Rosser and Tao Cheng. *Novel evaluation metrics for sparse spatio-temporal point process hotspot predictions-a crime case study*. International Journal of Geographical Information Science, pages 1–22, 2016. 50
- [Adrian 1926] Edgar D Adrian. *The impulses produced by sensory nerve endings*. The Journal of physiology, vol. 61, no. 1, pages 49–72, 1926. 80
- [Agrawal 1993] Rakesh Agrawal, Tomasz Imieliński and Arun Swami. *Mining association rules between sets of items in large databases*. In Acm sigmod record, volume 22, pages 207–216. ACM, 1993. 13
- [Baena-Garcia 2006] Manuel Baena-Garcia, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavaldá and R Morales-Bueno. *Early drift detection method*. In Fourth international workshop on knowledge discovery from data streams, volume 6, pages 77–86, 2006. 41

- [Balaji 2013] Bangaru Veera Balaji and Vedula Venkateswara Rao. *Improved classification based association rule mining*. International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 5, pages 2211–2221, 2013. 131
- [Balzanella 2010] Antonio Balzanella, Antonio Irpino and Rosanna Verde. *Dimensionality reduction techniques for streaming time series: A new symbolic approach*. In Classification as a Tool for Research, pages 381–389. Springer, 2010. 12
- [Bannister 2012] S Bannister and K Gledhill. *Evolution of the 2010–2012 Canterbury earthquake sequence*. New Zealand Journal of Geology and Geophysics, vol. 55, no. 3, pages 295–304, 2012. 55
- [Barddal 2015] Jean Paul Barddal, Heitor Murilo Gomes and Fabrício Enembreck. *Advances on Concept Drift Detection in Regression Tasks Using Social Networks Theory*. International Journal of Natural Computing Research (IJNCR), vol. 5, no. 1, pages 26–41, 2015. 37
- [Bifet 2007] Albert Bifet and Ricard Gavaldà. *Learning from Time-Changing Data with Adaptive Windowing*. In SDM, volume 7, page 2007. SIAM, 2007. 42
- [Bishop 1999] Christopher M Bishop and Wolfgang Maass. Pulsed neural networks. MIT Press Cambridge, MA, 1999. 79
- [Biswas 2000] Sambhunath Biswas. *Hilbert scan and image compression*. In Pattern Recognition, 2000. Proceedings. 15th International Conference on, volume 3, pages 207–210. IEEE, 2000. 45
- [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. 82

- [Bottou 1992] Léon Bottou and Vladimir Vapnik. *Local learning algorithms*. Neural computation, vol. 4, no. 6, pages 888–900, 1992. 37
- [Butz 1969] Arthur R Butz. *Convergence with Hilbert’s space filling curve*. Journal of Computer and System Sciences, vol. 3, no. 2, pages 128–146, 1969. 44
- [Chainey 2008] Spencer Chainey, Lisa Tompson and Sebastian Uhlig. *The utility of hotspot mapping for predicting spatial patterns of crime*. Security Journal, vol. 21, no. 1-2, pages 4–28, 2008. 51
- [Chen 2010] L Chen, P Dai and W Dou. *MTSKNN: multivariate two-sample tests based on K-nearest-neighbors*. R package version 0.0-5, 2010. 48
- [Chen 2012] Chun-Hao Chen, Tzung-Pei Hong and Vincent S Tseng. *Fuzzy data mining for time-series data*. Applied Soft Computing, vol. 12, no. 1, pages 536–542, 2012. 131
- [Das 1998] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan and Padhraic Smyth. *Rule Discovery from Time Series*. In KDD, volume 98, pages 16–22, 1998. 10, 16, 21
- [Dayan 2003] Peter Dayan, LF Abbott *et al*. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Journal of Cognitive Neuroscience, vol. 15, no. 1, pages 154–155, 2003. 80
- [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. 89

- [Dill 1990] Lawrence M Dill, Alex HG Fraser and Bernard D Roitberg. *The economics of escape behaviour in the pea aphid, Acyrthosiphon pisum*. *Oecologia*, vol. 83, no. 4, pages 473–478, 1990. 33
- [Doborjeh 2014] Maryam Gholami Doborjeh, Elisa Capecci and Nikola Kasabov. *Classification and segmentation of fMRI spatio-temporal brain data with a neucube evolving spiking neural network model*. In *Evolving and Autonomous Learning Systems (EALS)*, 2014 IEEE Symposium on, pages 73–80. IEEE, 2014. 97
- [Doborjeh 2015] Maryam Gholami Doborjeh and Nikola Kasabov. *Dynamic 3d clustering of spatio-temporal brain data in the neucube spiking neural network architecture on a case study of fmri data*. In *International Conference on Neural Information Processing*, pages 191–198. Springer, 2015. 97
- [Dougherty 1995] James Dougherty, Ron Kohavi, Mehran Sahami *et al.* *Supervised and unsupervised discretization of continuous features*. In *Machine learning: proceedings of the twelfth international conference*, volume 12, pages 194–202, 1995. 10
- [Dries 2009] Anton Dries and Ulrich Rückert. *Adaptive concept drift detection*. *Statistical Analysis and Data Mining*, vol. 2, no. 5-6, pages 311–327, 2009. 40
- [Elder 1991] D McG Elder, IF McCahon and MD Yetton. *The earthquake hazard in Christchurch: a detailed evaluation*. EQC Report, 1991. 54
- [Feldman 2012] Daniel E Feldman. *The spike-timing dependence of plasticity*. *Neuron*, vol. 75, no. 4, pages 556–571, 2012. 83
- [Finger 2001] Stanley Finger. *Origins of neuroscience: a history of explorations into brain function*. Oxford University Press, USA, 2001. 72

- [Fu 2011] Tak-chung Fu. *A review on time series data mining*. Engineering Applications of Artificial Intelligence, vol. 24, no. 1, pages 164–181, 2011. 9
- [Fusi 2000] Stefano Fusi, Mario Annunziato, Davide Badoni, Andrea Salammon and Daniel J Amit. *Spike-driven synaptic plasticity: theory, simulation, VLSI implementation*. Neural computation, vol. 12, no. 10, pages 2227–2258, 2000. 96
- [Gama 2004] Joao Gama, Pedro Medas, Gladys Castillo and Pedro Rodrigues. *Learning with drift detection*. In Brazilian Symposium on Artificial Intelligence, pages 286–295. Springer, 2004. 40
- [Gama 2014] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy and Abdelhamid Bouchachia. *A survey on concept drift adaptation*. ACM Computing Surveys (CSUR), vol. 46, no. 4, page 44, 2014. 38
- [Gammerman 1998] Alexander Gammerman, Volodya Vovk and Vladimir Vapnik. *Learning by transduction*. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pages 148–155. Morgan Kaufmann Publishers Inc., 1998. 37
- [Gao 2007] Jing Gao, Wei Fan, Jiawei Han and S Yu Philip. *A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions*. In SDM, pages 3–14. SIAM, 2007. 39
- [Geller 1997] Robert J Geller, David D Jackson, Yan Y Kagan and Francesco Mulargia. *Enhanced: earthquakes cannot be predicted*. Science, vol. 275, no. 5306, pages 1616–1620, 1997. 100
- [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. 83
- [Gerstner 2002] Wulfram Gerstner and Werner M Kistler. Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press, 2002. 78, 80, 81
- [Ghosh 2009] Dipak Ghosh, Argha Deb and Rosalima Sengupta. *Anomalous radon emission as precursor of earthquake*. Journal of Applied Geophysics, vol. 69, no. 2, pages 67–81, 2009. 100
- [Gilardi 2000] Nicolas Gilardi and Samy Bengio. *Local machine learning models for spatial data analysis*. Journal of Geographic Information and Decision Analysis, vol. 4, no. EPFL-ARTICLE-82651, pages 11–28, 2000. 37
- [Glackin 2012] Cornelius Glackin, Liam Maguire, Liam McDaid and John Wade. *Synchrony: A spiking-based mechanism for processing sensory stimuli*. Neural Networks, vol. 32, pages 26–34, 2012. 92
- [Gonçalves Jr 2013] Paulo Mauricio Gonçalves Jr and Roberto Souto Maior De Barros. *RCD: A recurring concept drift framework*. Pattern Recognition Letters, vol. 34, no. 9, pages 1018–1025, 2013. 50
- [Grant 2010] Rachel A Grant and Tim Halliday. *Predicting the unpredictable; evidence of pre-seismic anticipatory behaviour in the common toad*. Journal of Zoology, vol. 281, no. 4, pages 263–271, 2010. 100
- [Gutenberg 1956] Beno Gutenberg and Carl F Richter. *Earthquake magnitude, intensity, energy, and acceleration (second paper)*. Bulletin of the seismological society of America, vol. 46, no. 2, pages 105–145, 1956. 58

- [Hamilton 2008] Chris H Hamilton and Andrew Rau-Chaplin. *Compact Hilbert indices: Space-filling curves for domains with unequal side lengths*. Information Processing Letters, vol. 105, no. 5, pages 155–163, 2008. 44
- [Han 2002] Jiawei Han, Russ B Altman, Vipin Kumar, Heikki Mannila and Daryl Pregibon. *Emerging scientific applications in data mining*. Communications of the ACM, vol. 45, no. 8, pages 54–58, 2002. 1
- [Hand 1983] SC Hand. *The effect of temperature and humidity on the duration of development and hatching success of eggs of the aphid, Sitobion avenae*. Entomologia experimentalis et applicata, vol. 33, no. 2, pages 220–222, 1983. 33
- [Hartemink 2001] Alexander Hartemink and DK Gifford. *Principled computational methods for the validation and discovery of genetic regulatory networks*. Massachusetts Institute of Technology. PhD thesis, Ph. D. dissertation, 2001. 10
- [Hartono 2014] Reggio N Hartono, Rüssel Pears, Nikola Kasabov and Susan P Worner. *Extracting temporal knowledge from time series: A case study in ecological data*. In 2014 International Joint Conference on Neural Networks (IJCNN), pages 4237–4243. IEEE, 2014. 8, 19
- [Haverkort 2011] Herman Haverkort and Freek V Walderveen. *Four-dimensional Hilbert curves for R-trees*. Journal of Experimental Algorithmics (JEA), vol. 16, pages 3–4, 2011. 44
- [Hazan 2012] Hananel Hazan and Larry M Manevitz. *Topological constraints and robustness in liquid state machines*. Expert Systems with Applications, vol. 39, no. 2, pages 1597–1606, 2012. 86

- [Hebb 1949] Donald Olding Hebb. *The organization of behavior: A neuropsychological approach*. John Wiley & Sons, 1949. 74, 83
- [Henze 1988] Norbert Henze. *A multivariate two-sample test based on the number of nearest neighbor type coincidences*. *The Annals of Statistics*, pages 772–783, 1988. 48
- [Hilbert 1891] David Hilbert. *Ueber die stetige Abbildung einer Line auf ein Flächenstück*. *Mathematische Annalen*, vol. 38, no. 3, pages 459–460, 1891. 43
- [Ho 2005] Shen-Shyang Ho. *A martingale framework for concept change detection in time-varying data streams*. In *Proceedings of the 22nd international conference on Machine learning*, pages 321–327. ACM, 2005. 41
- [Hochachka 2007] Wesley M Hochachka, Rich Caruana, Daniel Fink, ART Munson, Mirek Riedewald, Daria Sorokina and Steve Kelling. *Data-Mining Discovery of Pattern and Process in Ecological Systems*. *The Journal of Wildlife Management*, vol. 71, no. 7, pages 2427–2437, 2007. 2
- [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. 76
- [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. 74, 75
- [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. 76
- [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. 76
- [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. 76
- [Hough 1999] Michael Hough, Hugo De Garis, Michael Korin, Felix Gers and Norberto Eiji Nawa. *SPIKER: Analog waveform to digital spike-train conversion in ATR's artificial brain (cam-brain) project*. In International Conference on Robotics and Artificial Life, 1999. 92
- [Huang 2006] Qinghua Huang. *Search for reliable precursors: A case study of the seismic quiescence of the 2000 western Tottori prefecture earthquake*. Journal of Geophysical Research: Solid Earth (1978–2012), vol. 111, no. B4, 2006. 101
- [Ikram 2014] Aqdas Ikram and Usman Qamar. *A rule-based expert system for earthquake prediction*. Journal of Intelligent Information Systems, vol. 43, no. 2, pages 205–230, 2014. 102
- [Izhikevich 2006] Eugene M Izhikevich. *Polychronization: computation with spikes*. Neural computation, vol. 18, no. 2, pages 245–282, 2006. 90
- [Jagadish 1990] Hosagrahar V Jagadish. *Linear clustering of objects with multiple attributes*. In ACM SIGMOD Record, volume 19, pages 332–342. ACM, 1990. 45

- [Joelianto 2008] Endra Joelianto, Sri Widiyantoro and Muhammad Ichsan. *Time series estimation on earthquake events using ANFIS with mapping function*. International Journal of Artificial Intelligence, vol. 3, no. A09, pages 37–63, 2008. 102
- [Jordan 2006] Thomas H Jordan. *Earthquake predictability, brick by brick*. Seismological Research Letters, vol. 77, no. 1, pages 3–6, 2006. 100
- [Kamel 1993] Ibrahim Kamel and Christos Faloutsos. *Hilbert R-tree: An improved R-tree using fractals*. 1993. 45
- [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. 88
- [Kasabov 1998b] Nikola K Kasabov. *ECOS: Evolving Connectionist Systems and the ECO Learning Paradigm*. In Iconip, volume 98, pages 123–128, 1998. 87
- [Kasabov 2007] Nikola Kasabov. *Evolving connectionist systems*. Springer, 2007. xv, 88, 94
- [Kasabov 2009] 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. 87
- [Kasabov 2012] 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. xiv, 71, 88, 89
- [Kasabov 2013a] N Kasabov. *Evolving computational intelligence: methods, systems, applications*. 2013. 108, 114

- [Kasabov 2013b] 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. 88, 90, 94, 96, 103
- [Kasabov 2014a] Nikola Kasabov. *Brain-like Information Processing for Spatio-Temporal Pattern Recognition*. In Springer Handbook of Bio-/Neuroinformatics, pages 813–834. Springer Berlin Heidelberg, 2014. 88
- [Kasabov 2014b] 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. xiv, 71, 91, 103
- [Kasabov 2014c] 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. 91, 103
- [Kasabov 2015a] Nikola Kasabov and Elisa Capecchi. *Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes*. Information Sciences, vol. 294, pages 565–575, 2015. 97
- [Kasabov 2015b] Nikola Kasabov, Maryam Gholami Dobarjeh and Zohreh Gholami Dobarjeh. *Mapping, Learning, Visualisation and Classification of fMRI data in evolving Spatio-Temporal Data Machines of Spiking Neural Networks*. Neural Networks and Learning Systems, IEEE Transactions on, 2015. 103

- [Kasabov 2015c] Nikola Kasabov, Nathan Scott, Enmei Tu, Stefan Marks, Neelava Sengupta, Elisa Capecci, Muhaini Othman, Maryam Gholami Doborjeh, Norhanifah Murli, Josafath Israel Espinosa-Ramos *et al.* *Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications*. Neural Networks, 2015. 103
- [Kasabov 2016] Nikola Kasabov, Nathan Matthew Scott, Enmei Tu, Stefan Marks, Neelava Sengupta, Elisa Capecci, Muhaini Othman, Maryam Gholami Doborjeh, Norhanifah Murli, Reggio Hartono *et al.* *Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: design methodology and selected applications*. Neural Networks, vol. 78, pages 1–14, 2016. 96
- [Kempton 1999] Richard Kempton, Wulfram Gerstner and J Leo Van Hemmen. *Hebbian learning and spiking neurons*. Physical Review E, vol. 59, no. 4, page 4498, 1999. 83
- [Keogh 2004] Eamonn Keogh, Selina Chu, David Hart and Michael Pazzani. *Segmenting time series: A survey and novel approach*. Data mining in time series databases, vol. 57, pages 1–22, 2004. 20
- [Kifer 2004] Daniel Kifer, Shai Ben-David and Johannes Gehrke. *Detecting change in data streams*. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, pages 180–191. VLDB Endowment, 2004. 41
- [Kostal 2007] Lubomir Kostal, Petr Lansky and Jean-Pierre Rospars. *Neuronal coding and spiking randomness*. European Journal of Neuroscience, vol. 26, no. 10, pages 2693–2701, 2007. 81

- [Kotsiantis 2006] Sotiris Kotsiantis and Dimitris Kanellopoulos. *Association rules mining: A recent overview*. GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, pages 71–82, 2006. 13
- [Kuang 2012] Xutao Kuang, Martina Poletti, Jonathan D Victor and Michele Rucci. *Temporal encoding of spatial information during active visual fixation*. Current Biology, vol. 22, no. 6, pages 510–514, 2012. 112
- [Kuncheva 2013] Ludmila I Kuncheva. *Change detection in streaming multivariate data using likelihood detectors*. IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, pages 1175–1180, 2013. 41
- [Lallich 2007] Stephane Lallich, Olivier Teytaud and Elie Prudhomme. *Association rule interestingness: measure and statistical validation*. In Quality measures in data mining, pages 251–275. Springer, 2007. 23
- [Larsen 1986] Richard J Larsen and Morris L Marx. *An introduction to mathematical statistics and its applications*. 1986. 12
- [Last 2001] Mark Last, Yaron Klein and Abraham Kandel. *Knowledge discovery in time series databases*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 31, no. 1, pages 160–169, 2001. 16
- [Lee 2012] Chung-Hong Lee. *Mining spatio-temporal information on microblogging streams using a density-based online clustering method*. Expert Systems with Applications, vol. 39, no. 10, pages 9623–9641, 2012. 133

- [Levine 2008] Ned Levine. *The "hottest" part of a hotspot: comments on the utility of hotspot mapping for predicting spatial patterns of crime*. Security journal, vol. 21, no. 4, pages 295–302, 2008. 52
- [Li 2009] Yonghong Li, Yanyou Liu, Zhou Jiang, Junwen Guan, Guixi Yi, Shuting Cheng, Bo Yang, Tianming Fu and Zhengrong Wang. *Behavioral change related to Wenchuan devastating earthquake in mice*. Bioelectromagnetics, vol. 30, no. 8, pages 613–620, 2009. 100
- [Liang 2008] Jan-Yie Liang, Chih-Sheng Chen, Chua-Huang Huang and Li Liu. *Lossless compression of medical images using Hilbert space-filling curves*. Computerized Medical Imaging and Graphics, vol. 32, no. 3, pages 174–182, 2008. 45
- [Lin 2003] Jessica Lin, Eamonn Keogh, Stefano Lonardi and Bill Chiu. *A symbolic representation of time series, with implications for streaming algorithms*. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pages 2–11. ACM, 2003. 11
- [Lin 2007] Jessica Lin, Eamonn Keogh, Li Wei and Stefano Lonardi. *Experiencing SAX: a novel symbolic representation of time series*. Data Mining and knowledge discovery, vol. 15, no. 2, pages 107–144, 2007. x, 12, 13
- [Liu 2001] JY Liu, YI Chen, YJ Chuo and HF Tsai. *Variations of ionospheric total electron content during the Chi-Chi earthquake*. Geophys. Res. Lett, vol. 28, no. 7, pages 1383–1386, 2001. 100
- [Lloyd 2010] Christopher D Lloyd. Local models for spatial analysis. CRC Press, 2010. 35

- [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. 81
- [López-Muñoz 2006] Francisco López-Muñoz, Jesús Boya and Cecilio Alamo. *Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal*. Brain research bulletin, vol. 70, no. 4, pages 391–405, 2006. 73
- [Lowe 1966] AD Lowe. *Some effects of weather on aphids in crops and pastures in Canterbury, New Zealand*. In Proceedings (New Zealand Ecological Society), pages 86–90. JSTOR, 1966. 33
- [Maass 2002] Wolfgang Maass, Thomas Natschläger 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. xiv, 87
- [Markram 1997] Henry Markram, Joachim Lübke, Michael Frotscher and Bert Sakmann. *Regulation of synaptic efficacy by coincidence of post-synaptic APs and EPSPs*. Science, vol. 275, no. 5297, pages 213–215, 1997. 83
- [Marks 2014] Stefan Marks, Javier E Estevez and Andy M Connor. *Towards the Holodeck: fully immersive virtual reality visualisation of scientific and engineering data*. In Proceedings of the 29th International Conference on Image and Vision Computing New Zealand, pages 42–47. ACM, 2014. 125

- [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. 74
- [Minsky 1969] Marvin Minsky and Seymour Papert. *Perceptrons*. 1969. 75
- [Montemurro 2008] Marcelo A Montemurro, Malte J Rasch, Yusuke Murayama, Nikos K Logothetis and Stefano Panzeri. *Phase-of-firing coding of natural visual stimuli in primary visual cortex*. Current biology, vol. 18, no. 5, pages 375–380, 2008. 82
- [Moon 2001] Bongki Moon, Hosagrahar V Jagadish, Christos Faloutsos and Joel H. Saltz. *Analysis of the clustering properties of the Hilbert space-filling curve*. IEEE Transactions on knowledge and data engineering, vol. 13, no. 1, pages 124–141, 2001. 45
- [Morales-Esteban 2013] A Morales-Esteban, F Martínez-Álvarez and J Reyes. *Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence*. Tectonophysics, vol. 593, pages 121–134, 2013. 101
- [Mörchen 2005] Fabian Mörchen and Alfred Ultsch. *Optimizing time series discretization for knowledge discovery*. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 660–665. ACM, 2005. 11
- [Natschläger 2002] Thomas Natschläger and Wolfgang Maass. *Spiking neurons and the induction of finite state machines*. Theoretical Computer Science, vol. 287, no. 1, pages 251–265, 2002. 135
- [Nelson 2004] ME Nelson. *Databasing the Brain: From Data to Knowledge*, 2004. 76

- [Nicol 2010] R Nicol, S Cox, R Langridge and K Pedley. *Surface rupture of the Greendale Fault during the Mw 7.1 Darfield (Canterbury) earthquake, New Zealand: initial findings*. Bulletin of the New Zealand Society for Earthquake Engineering, vol. 43, no. 4, 2010. 54
- [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. 92
- [O’hara 2010] Robert B O’hara and D Johan Kotze. *Do not log-transform count data*. Methods in Ecology and Evolution, vol. 1, no. 2, pages 118–122, 2010. 18
- [Omori 1894] Fusakichi Omori. On the after-shocks of earthquakes, volume 7. The University, 1894. 57
- [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. 97
- [Othman 2015] Muhaini Othman. *Spatial-temporal data modelling and processing for personalised decision support*. PhD thesis, Auckland University of Technology, 2015. 119
- [Oyama 2008] Koh-Ichiro Oyama, Yoshihiro Kakinami, Jann-Yenq Liu, Masashi Kamogawa and Tetsuya Kodama. *Reduction of electron temperature in low-latitude ionosphere at 600 km before and after large*

- earthquakes*. Journal of Geophysical Research: Space Physics (1978–2012), vol. 113, no. A11, 2008. 100
- [Peano 1890] Giuseppe Peano. *Sur une courbe, qui remplit toute une aire plane*. Mathematische Annalen, vol. 36, no. 1, pages 157–160, 1890. 43
- [Pears 2014] Russel Pears, Sripirakas Sakthithasan and Yun Sing Koh. *Detecting concept change in dynamic data streams*. Machine Learning, vol. 97, no. 3, pages 259–293, 2014. 41
- [Petersen 2011] Tanja Petersen, Ken Gledhill, Mark Chadwick, Nora H Gale and John Ristau. *The New Zealand national seismograph network*. Seismological research letters, vol. 82, no. 1, pages 9–20, 2011. 105, 107
- [Planinić 2004] Josip Planinić, Vanja Radolić and Branko Vuković. *Radon as an earthquake precursor*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 530, no. 3, pages 568–574, 2004. 134
- [Pradhan 2009] Gaurav N Pradhan and B Prabhakaran. *Association rule mining in multiple, multidimensional time series medical data*. In 2009 IEEE International Conference on Multimedia and Expo, pages 1720–1723. IEEE, 2009. 16
- [Pulinets 2003] S.A. Pulinets, A.D. Legen’Ka, T.V. Gaivoronskaya and V Kh Depuev. *Main phenomenological features of ionospheric precursors of strong earthquakes*. Journal of Atmospheric and Solar-Terrestrial Physics, vol. 65, no. 16, pages 1337–1347, 2003. 100

- [Purcaru 1982] G Purcaru and H Berckhemer. *Quantitative relations of seismic source parameters and a classification of earthquakes*. Tectonophysics, vol. 84, no. 1, pages 57–128, 1982. 112
- [Radil 2011] Steven M Radil. *Spatializing social networks: making space for theory in spatial analysis*. PhD thesis, University of Illinois at Urbana-Champaign, 2011. xi, 36
- [Reyes 2013] J Reyes, A Morales-Esteban and Francisco Martínez-Álvarez. *Neural networks to predict earthquakes in Chile*. Applied Soft Computing, vol. 13, no. 2, pages 1314–1328, 2013. 101, 107
- [Rhoades 2015] D.A. Rhoades, R. Buxton, Mueller C. and M.C. Gerstenberger. *Ionospheric Earthquake Precursors*. 2015. 134
- [Rosenblatt 1958] Frank Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, vol. 65, no. 6, page 386, 1958. 74
- [Ross 2012] Gordon J Ross, Niall M Adams, Dimitris K Tasoulis and David J Hand. *Exponentially weighted moving average charts for detecting concept drift*. Pattern Recognition Letters, vol. 33, no. 2, pages 191–198, 2012. 41
- [Rumelhart 1988] David E Rumelhart, Geoffrey E Hinton and Ronald J Williams. *Learning representations by back-propagating errors*. Cognitive modeling, vol. 5, no. 3, page 1, 1988. 75
- [Sagan 1993] Hans Sagan. *A three-dimensional Hilbert curve*. International Journal of Mathematical Education in Science and Technology, vol. 24, no. 4, pages 541–545, 1993. 44

- [Sakthithasan 2013] Sripirakas Sakthithasan, Russel Pears and Yun Sing Koh. *One pass concept change detection for data streams*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 461–472. Springer, 2013. 41
- [Sakthithasan 2015] Sripirakas Sakthithasan, Russel Pears, Albert Bifet and Bernhard Pfahringer. *Use of ensembles of Fourier spectra in capturing recurrent concepts in data streams*. In 2015 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2015. 133
- [Schilling 1986] Mark F Schilling. *Multivariate two-sample tests based on nearest neighbors*. Journal of the American Statistical Association, vol. 81, no. 395, pages 799–806, 1986. 48
- [Schliebs 2013] Stefan Schliebs and Nikola Kasabov. *Evolving spiking neural network-a survey*. Evolving Systems, vol. 4, no. 2, pages 87–98, 2013. 103
- [Schorlemmer 2010] Danijel Schorlemmer, J Douglas Zechar, Maximilian J Werner, Edward H Field, David D Jackson, Thomas H Jordan, RELM Working Group *et al*. *First results of the regional earthquake likelihood models experiment*. Pure and Applied Geophysics, vol. 167, no. 8-9, pages 859–876, 2010. 50
- [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. xv, 89, 92, 103
- [Schrauwen 2007] Benjamin Schrauwen, David Verstraeten and Jan Van Campenhout. *An overview of reservoir computing: theory, applications and implementations*. In Proceedings of the 15th European

- Symposium on Artificial Neural Networks. p. 471-482 2007, pages 471–482, 2007. 85
- [Scott 2013] Nathan Scott, Nikola Kasabov and Giacomo Indiveri. *NeuCube neuromorphic framework for spatio-temporal brain data and its python implementation*. In International Conference on Neural Information Processing, pages 78–84. Springer, 2013. 91
- [Sethi 2016] Tegjyot Singh Sethi, Mehmed Kantardzic and Hanqing Hu. *A grid density based framework for classifying streaming data in the presence of concept drift*. Journal of Intelligent Information Systems, vol. 46, no. 1, pages 179–211, 2016. 133
- [Shebalin 2006] P Shebalin. *Increased correlation range of seismicity before large events manifested by earthquake chains*. Tectonophysics, vol. 424, no. 3, pages 335–349, 2006. 124
- [Shibli 2011] Murad Shibli. *A novel approach to predict earthquakes using Adaptive Neural Fuzzy Inference System and conservation of energy-angular momentum*. Inter J Comp Inf Sys Ind Manag Appli. ISSN, pages 2150–7988, 2011. 102
- [Simmons 1963] George F Simmons. Introduction to topology and modern analysis. Tokyo, 1963. 44
- [Smyth 1992] Padhraic Smyth and Rodney M. Goodman. *An information theoretic approach to rule induction from databases*. IEEE transactions on Knowledge and data engineering, vol. 4, no. 4, pages 301–316, 1992. 23
- [Sobolev 2006] GA Sobolev and AA Lyubushin. *Microseismic impulses as earthquake precursors*. Izvestiya, Physics of the Solid Earth, vol. 42, no. 9, pages 721–733, 2006. 101

- [Sović 2013] Ivica Sović, Kristina Šariri and Mladen Živčić. *High frequency microseismic noise as possible earthquake precursor*. Research in Geophysics, vol. 3, no. 1, page e2, 2013. 100
- [Sporns 2004] Olaf Sporns, Dante R Chialvo, Marcus Kaiser and Claus C Hilgetag. *Organization, development and function of complex brain networks*. Trends in cognitive sciences, vol. 8, no. 9, pages 418–425, 2004. 93
- [Stein 1967] Richard B Stein. *Some models of neuronal variability*. Biophysical journal, vol. 7, no. 1, page 37, 1967. 78
- [Stein 2005] Richard B Stein, E Roderich Gossen and Kelvin E Jones. *Neuronal variability: noise or part of the signal?* Nature reviews. Neuroscience, vol. 6, no. 5, page 389, 2005. 80
- [Stirling 2001] Mark Stirling, Jarg Pettinga, Kelvin Berryman and Mark Yetton. *Probabilistic seismic hazard assessment of the Canterbury region, New Zealand*. Bulletin of the New Zealand National Society for Earthquake Engineering, vol. 34, no. 4, pages 318–334, 2001. xvi, 121, 124
- [Stirling 2008] Mark Stirling, Matthew Gerstenberger, Nicola Litchfield, Graeme McVerry, Warwick Smith, Jarg Pettinga and Philip Barnes. *Seismic hazard of the Canterbury region, New Zealand: New earthquake source model and methodology*. Bulletin of the New Zealand Society for Earthquake Engineering, vol. 41, no. 2, pages 51–67, 2008. xvi, 120
- [Stramondo 2011] Salvatore Stramondo, Christodoulos Kyriakopoulos, Christian Bignami, Marco Chini, Daniele Melini, Marco Moro, Matteo Picchiani, Michele Saroli and Enzo Boschi. *Did the September*

- 2010 (Darfield) earthquake trigger the February 2011 (Christchurch) event?* Scientific reports, vol. 1, 2011. 53
- [Tan 2002] Pang-Ning Tan, Vipin Kumar and Jaideep Srivastava. *Selecting the right interestingness measure for association patterns*. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 32–41. ACM, 2002. 23
- [Tapley 1992] William C Tapley and Joseph E Tull. *SAC-Seismic Analysis Code: Users Manual*. Lawrence Livermore National Laboratory, page 388, 1992. 110
- [Teulon 2002] DAJ Teulon and MAW Stufkens. *Biosecurity and aphids in New Zealand*. New Zealand Plant Protection, pages 12–17, 2002. 3
- [Thorpe 1998] Simon Thorpe and Jacques Gautrais. *Rank order coding*. In Computational Neuroscience, pages 113–118. Springer, 1998. xiv, 81
- [Tobler 1970] Waldo R Tobler. *A computer movie simulating urban growth in the Detroit region*. Economic geography, vol. 46, no. sup1, pages 234–240, 1970. 35
- [Tommasi 2017] Tatiana Tommasi, Novi Patricia, Barbara Caputo and Tinne Tuytelaars. *A deeper look at dataset bias*. In Domain Adaptation in Computer Vision Applications, pages 37–55. Springer, 2017. 112
- [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. 98

- [Utsu 1961] T Utsu. *A statistical study on the occurrence of aftershocks*. Geophysical Magazine, vol. 30, no. 4, 1961. 57
- [Van Dissen 1991] RJ Van Dissen. *An evaluation of seismic hazard in the Kaikoura region, southeastern Marlborough*. New Zealand Geological Survey Record, vol. 43, pages 93–99, 1991. 119
- [Wan 2007] Dingsheng Wan, Yitao Zhang and Shijin Li. *Discovery association rules in time series of hydrology*. In Integration Technology, 2007. ICIT'07. IEEE International Conference on, pages 653–657. IEEE, 2007. 10
- [Warasup 2006] Kittipong Warasup and Chakarida Nukoolkit. *Discovery association rules in time series data*. In KMUTT Research and Development Journal, volume 29, pages 447–462, 2006. 16
- [Watts 2009] Michael J Watts. *A decade of Kasabov's evolving connectionist systems: a review*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 3, pages 253–269, 2009. 88
- [Wells 1994] Donald L Wells and Kevin J Coppersmith. *New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement*. Bulletin of the seismological Society of America, vol. 84, no. 4, pages 974–1002, 1994. 55
- [Widiputra 2011] Harya Widiputra, Russel Pears and Nikola Kasabov. *Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 161–172. Springer, 2011. 37

- [Widmer 1996] Gerhard Widmer and Miroslav Kubat. *Learning in the presence of concept drift and hidden contexts*. Machine learning, vol. 23, no. 1, pages 69–101, 1996. 38
- [Wood 2016] Amy Wood, Ilan Noy, Miles Parker *et al.* *The Canterbury rebuild five years on from the Christchurch earthquake*. Reserve Bank of New Zealand Bulletin, vol. 79, pages 1–16, 2016. 3
- [Worner 1995] Susan P Worner, GM Tatchell and IP Woiwod. *Predicting spring migration of the damson-hop aphid *Phorodon humuli* (Homoptera: Aphididae) from historical records of host-plant flowering phenology and weather*. Journal of applied ecology, pages 17–28, 1995. 26
- [Worner 2002] SP Worner, GO Lankin, S Samarasinghe, DAJ Teulon, S Zytlenboset *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. 24
- [Wu 2006a] Si Wu, Shun-ichi Amari and Hiroyuki Nakahara. *Population coding and decoding in a neural field: a computational study*. Population, vol. 14, no. 5, 2006. 82
- [Wu 2006b] Yih-Min Wu and Ling-Yun Chiao. *Seismic Quiescence before the 1999 Chi-Chi, Taiwan, Mw 7.6 Earthquake*. Bulletin of the Seismological Society of America, vol. 96, no. 1, pages 321–327, 2006. 101
- [Wysoski 2006] Simeia Gomes Wysoski, Lubica Benuskova and Nikola Kasabov. *On-line 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. 81, 88, 94

-
- [Wysoski 2007] Simeia 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. 81
- [Wysoski 2010] Simeia 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. 88
- [Yao 2003] Xiaobai Yao. *Research issues in spatio-temporal data mining*. In Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Science, Virginia, 2003. 1
- [Zamani 2013] Ahmad Zamani, Mohammad Reza Sorbi and Ali Akbar Safavi. *Application of neural network and ANFIS model for earthquake occurrence in Iran*. Earth Science Informatics, vol. 6, no. 2, pages 71–85, 2013. 102
- [Zhang 1998] Yuefeng Zhang. *Space-filling curve ordered dither*. Computers & Graphics, vol. 22, no. 4, pages 559–563, 1998. 45, 84