Neuromorphic Spiking Neural Network Algorithms for Machine Learning and Pattern Recognition

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

Supervisors

Prof. Jacqueline Whalley

Assoc. Prof. Grace Wang

2022

By

Mahima Milinda Alwis Weerasinghe

School of Engineering, Computer and Mathematical Sciences

Abstract

This thesis presents novel Machine Learning (ML) and pattern recognition algorithms based on Spiking Neural Networks (SNNs). SNNs mimic biological information processing more closely than the contemporary rate-coded neural networks (ANNs) with a premise of greater efficiency. However, since information in SNNs is represented by discrete spikes, developing robust learning algorithms is a challenge due to its non-differentiable nature. This complexity has led to overgrown structures and computationally costly learning strategies to achieve higher ML performances which negatively impacts the efficiency premise of SNNs. This research attempts to address this challenge by drawing inspiration from plasticity-based learning in the biological brain.

The brain learns complex patterns with minimal energy expenditure using various plasticity techniques. Plasticity in the brain occurs at synaptic, neuron, circuit, and region-wise levels across millisecond to annual time scales. Inspired by these plasticity functions, this thesis presents developments of adaptive SNN algorithms for spatiotemporal data modelling under batch mode and online learning. The novel algorithms proposed are tested using electroencephalogram (EEG) data in classification tasks related to mental stress, emotions, and motor movements.

The first contribution of this thesis is a basic three-layered SNN algorithm developed using Spike Time Dependent Plasticity (STDP) and an evolving classifier that learns from local and global spiking activity in a single pass. Structural plasticity (SP) was incorporated into this SNN via Differential Evolution (DE) algorithm, which optimised the hidden layer neuron population operating with STDP. The resulting architectures performed better than overgrown and undergrown structures and produced heuristics for neuron population selection. Using the network heuristics of this study, a more brain-like learning method with STDP and Intrinsic Plasticity (IP) is introduced as the second contribution of this study. IP had found to be regulating the neurons' excitability to maintain network spiking stability in STDP setups. However, the particulars in guiding STDP+IP learning lack clarity. In this work, STDP+IP learning is guided using entropy and neuron redundancy measures. Selected neurons are pruned at the end of the training according to the *use it or lose it* strategy, a phenomenon also observed in the brain. In some cases, pruning improved the generalisation capability of the SNN. By developing this method further, an online learning method named Online-Neuroplasticity Spiking Neural Network (O-NSNN) was introduced as the third contribution. The O-NSNN is tested in classifying acute stress of individuals, and network behaviour is examined to find links between acute and perceived stress (PS). The highest classification accuracy was recorded at 93.63% and the lowest at 85.29%,

outperforming SNN without SP. According to the knowledge extractions, high PS individuals had less sensitivity to daily acute stressors.

Experiments of this thesis aims to develop biologically plausible learning algorithms to enhance ML performance while improving efficiency. These developments are discussed at a fundamental level of network spiking behaviour. Therefore, the contributions reduce the knowledge gap between plasticity, spiking behaviour, and ML performance while providing insights into how the brain learns.

Table of Contents

Abstract i
Table of Contents iii
List of Figures vii
List of Tables xi
Attestation of Authorship xii
Co-Authored Works xiii
Acknowledgements xiv
Chapter 1 Introduction 1
1.1 Rationale and motivation 2
1.2 Research Questions 4
1.3 Thesis structure 5
Chapter 2 Evolution of Spiking Neural Networks 8
2.1 Review Method 9
2.2 Development of Artificial Spiking Neuron 10
2.1.1 First-generation neurons 10
2.1.2 Second-generation neurons 12
2.2.1 Third-generation neurons 14
2.3 Types of Artificial Spiking Neurons 17
2.3.1 Hodgkin-Huxley Model 18
2.3.2 Izhikevich Model 19
2.3.3 Leaky integrate-and-fire neuron 20
2.4 Spike-based learning techniques in ASNNs 22
2.4.1 Single-spike learning 23
2.4.2 Multi-spike learning 25
2.5 Summary 30
Chapter 3 Electroencephalography and Spike Encoding in SNNs 31
3.1 Spatiotemporal data and electroencephalogram 31
3.2 Background to EEG technology 32
3.3 EEG electrode placement 33
3.4 Spike encoding 34
3.5 Analogue EEG to spike conversions found in SNNs 35
3.5.1 Standalone conversions 35
3.5.2 Conversions using network neurons 36
Chapter 4 Preliminary Studies on Brain Data Modelling using SNNs 37
4.1 The SNN learning algorithm 37
4.2 SNN Framework 38
4.3 The EEG datasets 41
4.3.1 Wrist dataset 41
4.3.2 DEAP dataset 42

4.4 Experimental Procedure 43
4.5 Evaluation Measures 43
4.6 Results: Wrist dataset 44
4.7 Results: DEAP dataset 45
4.8 Conclusion 46
Chapter 5 Emotional Stress Classification using Spiking Neural Networks 47
5.1 Reference: 47
5.2 Author contribution: 47
5.3 Preamble: 47
5.4 Abstract 49
5.5 Introduction 49
5.5.1 Electroencephalography and Spiking Neural Networks 50
5.5.2 Stress recognition using EEG 51
5.6 Method 52
5.6.1 Data preparation 52
5.6.2 SNN Architecture 54
5.6.3 Experimental framework 55
5.6.4 Network optimisation 56
5.6.5 NeuCube architecture 56
5.7 Results and Discussion 57
5.8 Conclusion 61
Chapter 6 Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling 64
6.1 Reference: 64
6.2 Author contribution: 64
6.3 Preamble: 64
6.4 Abstract 65
6.5 Introduction 66
6.6 Structural Plasticity (SP) in SNNs 67
6.7 Stress recognition using EEG 69
6.7.1 EEG Datasets 69
6.7.2 Data Annotation 70
6.7.3 Data Encoding 71
6.8 SNN Architecture 72
6.8.1 Leaky Integrate and Fire neurons 73
6.8.2 Spike Time Dependent Plasticity 74
6.8.3 Classifier learning 75
6.9 Network Optimisation 76
6.9.1 Differential Evolution 77
6.9.2 Experiment Framework 78
6.10 Results 80
6.10.1 Testing Network Performances 80
6.10.2 Inactive Neurons and LIF Threshold 86

6.11 Discussion 87
6.12 Conclusion and Future Work 88
6.13 Appendix 89
Chapter 7 Brain Plasticity and Learning in SNNs 91
7.1 Plasticity in the Brain 91
7.1.1 Synaptic level plasticity 92
7.1.2 Cellular level plasticity (Intrinsic Plasticity) 93
7.1.3 Circuit and regional-level plasticity 94
7.2 Biologically inspired plasticity for Machine Learning 95
7.2.1 Synaptic and cellular level plasticity 95
7.2.2 Circuit level plasticity 97
Chapter 8 Ensemble Plasticity and Network Adaptability in ASNNs 99
8.1 Reference: 99
8.2 Author contribution: 99
8.3 Preamble: 99
8.4 Abstract 100
8.5 Introduction 101
8.6 Related work 103
8.6.1 Standalone IP 103
8.6.2 IP and STDP - Ensemble Learning 103
8.6.3 Structural adaptation in ASNNs 104
8.7 Methodology and Data 105
8.7.1 Leaky Integrate and Fire Neuron 106
8.7.2 Spike Time Dependent Plasticity 107
8.7.3 Intrinsic Plasticity 108
8.7.4 Neurons Pruning 110
8.7.5 Classifier learning 110
8.8 Experimental Framework 111
8.8.1 Data sets used 111
8.8.2 Data encoding 112
8.8.3 Performance evaluation criteria 114
8.8.4 Experiment layout 114
8.9 Results 114
8.9.1 Wrist dataset 114
8.9.2 DEAP dataset 119
8.10 Discussion 122
8.11 Conclusion 124
8.12 Appendix 125
Chapter 9 Mental Stress Recognition on the fly using Neuroplasticity Spiking Neural Networks 127
9.1 Reference: 127
9.2 Author contribution: 127
9.3 Preamble 127

9.4 Abstract 128
9.5 Introduction 129
9.5.1 Implications of mental stress 129
9.5.2 Stress and electroencephalogram 129
9.5.3 Stress recognition on the fly 129
9.5.4 Data drifts and online learning 130
9.5.5 Spiking Neural Networks (SNNs) 130
9.5.6 Acute stress and data collection 132
9.6 Methods 136
9.6.1 Neuroplasticity spiking neural network 136
9.6.2 Performance evaluation 139
9.6.3 Ethics approval and consent to participate 140
9.6.4 EEG Data 140
9.7 Results 142
9.7.1 Increased accuracy and robustness in O-NSNN 144
9.7.2 The efficiency of O-NSNN 146
9.7.3 O-NSNN knowledge extraction 147
9.8 Discussion 149
9.8.1 The performance of the O-NSNN. 149
9.8.2 Biological Plausibility in Learning 151
9.9 Conclusion 152
Chapter 10 Conclusion & Future Work 153
10.1 Summary 153
10.2 Revisiting the research questions 158
10.3 Contribution 161
10.4 Limitations 163
10.5 Future work 163
References 165
Glossary 184
Appendix – Copyright Letters for Figures 187

List of Figures

Figure 1.1: Organization of this thesis presented with the organization for each chapter. Journal publications are shown as a green clipped box (A publication summary is given in Table 0.1) 7

Figure 2.1: Avenues of machine learning development and impact of psychology and neuroscience.. 9

Figure 2.2: Schematic diagram of McCulloch-Pitts neuron..... 11

Figure 2.3: Activation functions of second-generation neurons 13

Figure 2.4: Basic behaviour of a spike..... 14

Figure 2.5: Computational representation of the biological neuron. (a) Drawing of a biological neuron. Soma (circled), axon extending from the soma and branching into dendrites. (b) Graph indicating how input spikes impact membrane potential and finally reach the threshold θ to produce a spike. 15

Figure 2.6: Artificial spiking neurons in terms of neuro-computational features and implementation cost 17

Figure 2.7: Neuro-computational properties of spiking neurons 18

Figure 2.8: Leaky integrate-and-fire neuron representation using resistor-capacitor circuit 20

Figure 3.1: Electrode placement according to international 10-20 system. The proportionate distance between each electrode from Nasion to Inion left side view (Left). Electrode placement top view (Right). Electrode naming and corresponding brain areas: Fp (pre-frontal), F (Frontal), T (Temporal), P (Parietal), C (Central) and O (Occipital). Electrodes of the midline sagittal plane (FpZ, Fz, Cz,Oz). Odd and even numbers followed by letters refer to left-hemispherical and right-hemispherical electrodes respectively. The A1, A2 (also denoted as M1, M2) are connected to the mastoids (behind the outer ear) to provide contralateral reference to other electrodes. 34

Figure 4.1: The basic SNN architecture developed 40

Figure 4.2: Average accuracy and robustness (interquartile range of a box) with increasing number of neurons – Wrist Dataset. Each box represents the results of 30 train-test cycles. 44

Figure 4.3: Average accuracy and robustness (interquartile range of a box) with increasing number of neurons – DEAP Dataset. Each box represents results of 30 train-test cycles..... 45

Figure 5.1: Overview of EEG sample preparation..... 53

Figure 5.2: The spiking neural network used for the classification of EEG data 54

Figure 5.3: NeuCube spiking neural network architecture (N. K. Kasabov, 2014) mapped using Talairach brain template (Talairach & Tournoux, 1988) for network behaviour analysis (View from the left side of the 3D reservoir) 57

Figure 5.4: Split testing accuracy (60% training, 40% testing) over 30 iterations 58

Figure 5.5: Five-fold cross-validation accuracy over 30 iterations..... 58

Figure 5.6: Input neuron firing correlation analysis after training two networks separately with (a)stressed (b)relaxed EEG data samples 59

Figure 5.7: Input neuron cluster analysis using 2D views from left, right, back and top of the NeuCube reservoir after training separately using class-specific data (a) Stressed (b) Relaxed. Brighter neurons represent higher activations and darker neurons lower activation. Darker neurons are contoured in red for better visibility 60

Figure 5.8: Synaptic connection formation according to small world algorithm(Watts & Strogatz, 1998) after training the network with class-specific incoming spike trains (a) Stressed (b) Relaxed 60

Figure 5.9: Confusion matrix of the best model that produced 88% accuracy for 60:40 split testing .. 61

Figure 6.1:Data preparation process used for the second dataset 70

Figure 6.2: Spike representation of FP1 channel extracted from a stressed class sample. The top plot illustrates original signal (Blue) and reconstruction of the same using spikes generated (Red) 71

Figure 6.3: SNN architecture used for classification task using dataset 2 accommodating 32 inputs .. 72

Figure 6.4: Fitness function of a single network initiation 78

Figure 6.5: Representation of a sample trial solution vector with 6 hyperparameters (given under headings highlighted in blue) and hidden layer neuron count (highlighted in red) which was only used in Exp. 1 79

Figure 6.6: Statistical analysis of the performance for dataset 1 across 30 random initiations 80

Figure 6.7: Comparative statistical analysis of the generalisation capability of the 3 networks for dataset 1 80

Figure 6.8: Statistical analysis of the performance for dataset 2 across 30 random initiations 82

Figure 6.9: Comparative statistical analysis of the generalisation capability of the 3 networks for dataset 2 83

Figure 6.10: Graphical representation comparing inactive neurons to total neurons of different structures tested with dataset 1(Blue) and dataset 2(Red) 86

Figure 6.11: LIF Threshold selections for the SNNs with different η values for dataset 2 86

Figure 6.12: Hyperparameter (i.e., Neurons & synaptic properties) preferences over 3 SNN structures compared for dataset 1(Left) and dataset 2 (Right) 90

Figure 6.13:Comparison of firing count of each neuron after training with dataset 1 for network with 111 neurons (Blue) and 167 neurons (Orange) 90

Figure 7.1: Plasticity methods associated with computation in neurobiological systems 92

Figure 7.2: Classic STDP function (Bi & Poo, 1998) and Dopamine-modulated STDP (J. C. Zhang et al., 2009) 95

Figure 8.1: SNN algorithm used for experimentation with a single hidden layer. The input layer consists of LIF neuron pairs capable of propagating both excitatory and inhibitory spikes (Red-excitatory and black-inhibitory). Synapses between input and hidden layer 106

Figure 8.2: Spiking demonstration of the LIF with multiple threshold levels. Membrane potential is represented in blue and spike emissions are represented in orange dashed lines. a) Spiking activity at 1v threshold; b) Spiking activity at 0.8v threshold 109

Figure 8.3: Flow chart representation of the ensembled learning and pruning process. Analogue EEG signals are converted to spikes using Address Event Representation (AER) (Delbruck & Lichtsteiner, 2007) encoding method given in Figure 8.13. STDP (Equations (29),(30) and (31)) and IP (Equations (32)) implemented as an ensemble learning method to operate simultaneously. If average active neurons η is maximised, entropy of the spiking PDF is evaluated (Equation (32)) to adjust IP learning rates, θ_{pos} and θ_{neg}. Once the classifier is trained (Equations (36) & (37)), network pruning is applied (Equations (34) & (35)). 113

Figure 8.4: Result of sweeping θ_{neg} from 1×10^{-7} to 1×10^{-3} on η (Orange) and Entropy (Blue) while θ_{pos} is kept constant at 1×10^{-3} 115

Figure 8.5: Result of sweeping θpos from 1×10^{-5} to 1 on η (Orange) and Entropy (Blue) while θneg is kept constant at 1×10^{-3} 115

Figure 8.6: Spiking raster plots and firing distributions obtained after propagating the training data over the trained network a) Standalone STDP raster b) STDP+IP raster c) Standalone STDP firing distribution d) STDP+IP firing distribution 117

Figure 8.7: Performance comparison of STDP only and STDP+IP training. Cross-validation is denoted by *C* and split testing by *T*. *Acc* for Accuracy, *F1* for F1-Score and *Kap* for Kappa value. 118

Figure 8.8: a) Pruned network accuracy performance compared under 70/30 train-test split b) Comparison of 70/30 train-split performance after STDP only, STDP+IP and STDP+IP Pruned (denoted as Pr) approaches. Split testing is denoted by T,Acc for Accuracy, F1 for F1-Score and Kap for Kappa value. 119

Figure 8.9: Spiking raster plot obtained after propagating the training data over the trained network using STDP+IP 120

Figure 8.10: a) Spiking distribution under standalone STDP b) Spiking distribution under STDP+IP by T. Acc for Accuracy, F1 for F1-Score and Kap for Kappa value. 120

Figure 8.11: Performance comparison of STDP only and STDP+IP training for the DEAP dataset. Cross-validation is denoted by C and split testing 121

Figure 8.12: Spiking raster plots under different θpos and θneg conditions a) $\theta pos < \theta neg$ b) $\theta pos = \theta neg$ 125

Figure 8.13: AER encoding algorithm 125

Figure 9.1: (a) The proposed O-NSNN architecture for stress recognition. EEG originating from FP1, FP2, T7 and T8 channels are encoded into spikes (using the AER algorithm) and propogated through a three-layered SNN architecture. An STDP rule is used for temporal learning between the input layer and the hidden layer. Hidden layer neurons use IP to adapt excitability based on the incoming data. The output layer learns using RO and SDSP rules. Each hidden layer neuron prunes itself according to soft-pruning rule and, the output layer evolves. (b) Stress class input samples of P1 with different spike rate distribution (Input drift) (c) Two separate classes of P1 (Critical and Positive) with the same input spiking distributions (Concept drift). 134

Figure 9.2: Performance variation of individual models. Performance distribution obtained from 30 testing cycles. At each cycle the initial weights between the input to hidden layers are selected pseudo randomly according to gaussian ditribution. S – Online learning with SP, N – Online learning without SP, B – Batch mode learning without SP..... 145

Figure 9.3:(a) Number of output neurons evolved by O-NSNN during 30 testing cycles for each participant model (b) Prequential accuracy progression with the number of samples increasing (c) Sample spiking raster plot of the hidden layer for P1 146

Figure 9.4: Euclidean Distance between initial(Blue) and final(Red) output neurons. The initiation process use the first 15 samples to evelove 15 output neurons. (a) without pruning or evolving new neurons (O-RSNN) (b) with pruining and evolving new neurons (O-NSNN) 144

Figure 9.5: (a) Average differences between EEG samples represented by Euclidean distances. The signals during Neutral stimuli is selected as the baseline. (b) Spiking interaction pattern between channels for the High stress group (c) Spiking interaction pattern between channels for the Medium stress group (d) Spiking interaction pattern between channels for the Low stress group 148

Figure 9.6: Cumulative weights of the synapses fanning out from respective inputs calculated according to perceived stress groups 147

List of Tables

Table 2.1 Summary of ASSN algorithms using single neuron architecture 28

Table 2.2 Summary of ASSN algorithms using single or multiple layers of neurons (feed-forward).. 29

Table 3.1 Comparison of brain imaging techniques 33

Table 3.2 Advantages and disadvantages of different encoding algorithms 35

Table 4.1 Hyperparameter settings of the experiment 41

Table 5.1 Network hyperparameter values 62

Table 5.2 Performance comparison with previous studies 63

Table 6.1: Hyperparameters of the experiment 77

Table 6.2: Network performance comparison dataset 1 81

Table 6.3: Dataset 1 classification accuracies 82

Table 6.4: Network performance comparison dataset 2 83

Table 6.5: Dataset 2 classification accuracies 83

Table 6.6: Dataset 2 classification valence and arousal dimensions 85

Table 6.7: Optimised hyperparameter values 89

Table 8.1: Average ML performances summarised for Wrist data 118

Table 8.2: Average ML performances summarised for DEAP data 122

Table 8.3: Hyper-parameter values used for experimenting 126

Table 9.1: Participant categorisation based on Perceived Mental Stress scores calculated from PSS-14 questionnaire 135

Table 9.2: O-NSNN hyperparameters 142

Table 9.3: Accuracy comparison between online (with (O-NSNN) and without SP (O-RSNN)) and batch mode (B-RSNN) learning 143

Attestation of Authorship

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

13/12/2022

Signature

Date

Co-Authored Works

Chapters	Publication	Author (%)
Chapter 5	Weerasinghe, M.M.A, Wang, G. & Parry, D. (2022). Emotional Stress Classification using Spiking Neural Networks. Psychology & Neuroscience. https://doi.org/10.1037/pne0000294	MMAW: 90 GW: 5 DP: 5
Chapter 6	Weerasinghe, M. M. A., Espinosa-Ramos, J. I.,Wang, G. Y., & Parry, D. (2021). Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling. IEEE Access, *9*, 117338–117348. https://doi.org/10.1109/ACCESS.2021.3099492	MMAW:82.5 JIER: 7.5 GW: 5 DP: 5
Chapter 8	Weerasinghe, M. M. A., Parry, D., Wang, G., & Whalley, J. (2022). Ensemble plasticity and network adaptability in SNNs. https://doi.org/10.48550/arXiv.2203.07039	MMAW: 87.5 GW: 2.5 DP: 2.5 JW: 7.5
Chapter 9	Weerasinghe, M.M.A., Wang, G., Whalley, J., Crook-Rumsey, M. (2022). Mental Stress Recognition on the fly using Neuroplasticity Spiking Neural Networks. Research Square. https://doi.org/10.1037/pne000029410.21203/rs.3.rs-1841009/v1	MMAW: 84 GW: 7 JW: 6 MCR: 1

We, the undersigned, hereby agree to the percentages of participation to the chapters identified above:

…….

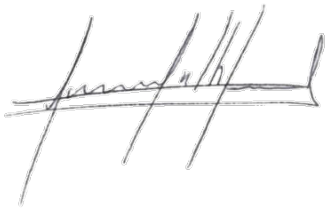
Prof. Jacqueline Whalley

……

Assoc. Prof. Grace Wang

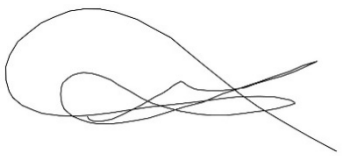
……

Prof. Dave Parry



……

Dr. Josafath Israel Espinosa- Ramos



……

Dr.Mark Crook-Rumsey

Acknowledgements

I pen this down with immense gratitude to everyone and everything that helped me on this adventure. "*Everything in Life is Vibration,"* said the great Albert Einstein, and to those vibrations that propelled me forward in this journey, this appreciation is for you.

Professor Jacqueline Whalley – for guiding me during a pandemic and a personally challenging time. I feel fortunate to have you as my supervisor. Your unwavering attention to detail and dedication to students is remarkable, and I am inspired to replicate them.

Associate Professor Grace Wang – I am incredibly grateful for your valuable time spent on me as a supervisor, helping to build my future in academia and providing me with subject expertise.

Professor Dave Parry – for encouraging me to experiment even with the craziest scientific ideas. I am genuinely thankful for all the encouragement and support during my study.

I would also like to thank Professor Nikola Kasabov for helping me start my PhD at AUT and providing a comfortable atmosphere approving financial assistance for the first year of my study.

A special thanks to Joyce D'Mello, a mother to all of us in the lab, always inquiring about our well-being and supporting and motivating us to get things done. I am ever so thankful to you.

Big thank you to Dr Josafath Israel Espinosa-Ramos and Dr Mark Crook-Rumsey for sharing their developments, enlightening academic arguments and great insights during my study.

I sincerely appreciate Professor Ajith Narayanan and Professor Denise Taylor for evaluating and helping me start the study. To Anne, Vinita, Dhvani and Akshay for sharing their subject expertise and helping me avoid unforeseen obstacles. To Anna, Sajith and Kaushalya for their great company during the study. I also appreciate Sadie Lo for getting me involved with Teaching Assistantships, which has helped me sharpen my knowledge-sharing skills. A big thank you to Karishma and Jennifer from the CMS PhD team and Jessica and Ksenya from GRS for helping with the smooth functioning of the PhD. Supporting with computing resources for the study, I salute the BJ (Bumjun Kim) and the IT support team at AUT.

Much appreciation to Matua John Tapene, Professor Valance Smith, Lorraine and the beautiful voices of the AUT Whanau choir for making my time at Aotearoa extra special. With you in the Marae feels home to me.

I forward my gratitude to all at AUT for making this study possible.

A special appreciation for people I have known for a long time (or I have felt so). Uncle Sunil, Aunty Wasathamala and Malli for providing a great place to rest after a long day of work, often with Sri Lankan food and great political debates. Randy, for being a great friend and a brother. Dinusha for the great times we had, and Rachel, we will never forget. To Amber and Andrea for their smiles, kindness, and kinship. Sala, Iranga, Mandara and their loved ones for always being there for me to help in New Zealand. Siri Mama and family for caring for me and not letting me miss home. Dinouk, Deva, Dulari, Ranil, Koshitha, Anuradha Akka, Asanthika and Swapna, for all their support and great friendship. Also, I greatly appreciate Professor Upul Sonnadara, Professor Rangika Halwathura and Dr M K Jayananda for helping me apply for PhDs and writing recommendations whenever needed. A big thank you to Champa Nanda for helping my dreams come true.

Finally, my reverence to Buddha, Dhamma and the Sanga for the great wisdom that has helped me remain joyous no matter what. To the guardian deities of my life, Amma, Thaththa, Ayya and Amy, this is for you. Much merit to you for being the light of my life.

From the trees I have laid down upon to all of you I have met on this beautiful journey, my heartfelt gratitude!

Ethics statement

Ethics consents were approved by the Auckland University of Technology Ethics Committee on 2nd October 2019 and ethics approval was granted, AUTEC reference number 19/231.

Chapter 1 Introduction

Human mechanisms have inspired technological developments throughout history. Of these mechanisms, the brain has been a great source of inspiration for innovation in computer science. The emergence and development of Artificial Intelligence (AI) technology are driven by the same motivation, where human intelligence is idealized as the benchmark. Famous quotes from the pioneers in AI reveal this underpinning ideology.

> *"What we want is a machine that can learn from experience"* – Alan Turing (Copeland, 2020, p. 393)

> *"Artificial intelligence is the science of making machines do things that would require intelligence if done by men"* – Marvin Minsky (Dennis, 2021)

> *"Our ultimate objective is to make programs that learn from their experience as effectively as humans do"* – John McCarthy (McCarthy, 1963, p. 2)

In the quest to reach human-like intelligence, a differentiation is observed in terms of the approaches used for AI development; some studies draw inspiration predominantly from neuroscience and psychology, while others rely purely on mathematical and statistical concepts (James et al., 2017). Although drawing inspiration from neuroscience and psychology does not guarantee better performance, it is certainly a promising approach when considering the history of innovative solutions it has produced (Fukushima, 1975; Hebb, 2005; Hinton & Salakhutdinov, 2006; Jaeger, 2003; Maass et al., 2002; McCulloch & Pitts, 1943; Rosenblatt, 1958). In recent years, the gap between biological and AI systems has reduced due to the new knowledge from psychology and neuroscience that has revealed the underpinning mechanisms of human intelligence.

McCullough and Pitts proposed that the brain is an interconnected population of neurons that can perform simple logical computations individually (McCulloch & Pitts, 1943). This neurobiologically driven concept led to the production of artificial neurons known as threshold gates. Extending beyond an artificial neuron, psychologist Frank Rosenblatt produced a trainable artificial neural network (ANN) known as the perceptron (Rosenblatt, 1958). Although these implementations are high-level abstractions of the concepts found in neurobiology and human cognition, they laid the foundation for modern biologically inspired Machine Learning (ML) methods based on ANNs. This thesis presents novel ML and pattern recognition methods based on Artificial Spiking Neural Networks (ASNNs), a type of ANN surpassing the biological plausibility of traditional ANNs.

1.1 Rationale and motivation

The energy requirements of contemporary ANN algorithms have increased dramatically in recent years, making it hard to be implemented in low-power applications such as robotics and the Internet of Things (IoT) (Pfeiffer & Pfeil, 2018). This energy consumption can be attributed mainly to the massive number of neurons and synapses used (as an example in Deep Learning Neural Networks), especially in complex ML tasks. Despite the enormous power requirement, they have performed extraordinary well with the help of powerful computer hardware. The layered architecture of Deep Neural Networks (DNNs) accommodates a considerable number of computational elements (i.e., neurons and synapses) regulated by exhaustive learning techniques to enable successful function approximation. The AlphaGo program, which repeatedly outperformed expert human players in the game of Go, and the language prediction program the Generative Pre-trained Transformer (GPT-3) are famous examples of such successful DNN-based applications. However, the common limitation in such systems is the computational cost involved in training and inferring them. For example, the energy cost for training GPT-3 was estimated at 12 million US dollars (Brown et al., 2020). This underlines the energy requirement of the computational elements and the learning algorithms at a fundamental level. Hence moving forward, it is essential to investigate novel data processing concepts that are both energy efficient and robust in complex function approximation tasks.

The human brain is estimated to operate on 20 watts of power (Cox & Dean, 2014) while demonstrating remarkable learning capability. This is far more efficient than modern-day ANNs. This difference in learning capability and energy consumption provided one of the key motivations for the work of this thesis. If we make brain-like learning techniques for ANNs, this should lead to more energy-efficient learning capabilities. Neuroscientists have proposed that precisely timed *spikes* (action potentials) observed in the biological brain are the fundamental reason behind energy efficiency and learning capabilities (Bialek & Rieke, 1992; Gerstner et al., 1996; S. J. Thorpe & Imbert, 1989). Over seventy years ago, the ability of biological neurons to produce such spikes was investigated in vivo and mathematically modelled (FitzHugh, 1961; Hodgkin & Huxley, 1952). In seminal work, Maass (1997) introduced the mathematical abstractions called artificial spiking neurons as the third generation of artificial neurons. Maass argued that artificial spiking neurons could be powerful and efficient computational elements for machine intelligence tasks. He provided empirical evidence that artificial spiking neurons had superior processing capability compared to continuous function neurons typically found in the ANNs of the time.

Inspired by the representative power and efficiency, researchers in the domains of machine learning have adopted mathematical abstractions of spiking neurons and learning mechanisms, such as Spike

Time Dependent Plasticity (STDP) found in biology, to progress the development of ASNN applications (Diehl & Cook, 2015; Hao et al., 2020; N. Kasabov et al., 2013; Masquelier & Thorpe, 2007; Rathi et al., 2019). Using STDP, the weight adjustment between presynaptic and postsynaptic neurons are carried out based on the spiking synchrony between the neurons. Although these applications have not outperformed DNNs, the performance gap has started to reduce (i.e., in terms of ML metrics) (Tavanaei et al., 2019). However, the heavy emphasis on ML accuracy has negatively affected the computational efficiency that ASNNs were originally known (Pfeiffer & Pfeil, 2018). Typically, both network size (i.e., number of neurons, synapses, layers) and learning algorithms directly impact the efficiency of a neural network. In biological systems, neuroscientists have found regulation of the circuit size to be a primary requirement in refining learning and managing energy expenditure (Iglesias & Villa, 2006, 2007; Yamaguchi & Miura, 2015). Therefore, improving learning algorithms by leveraging on network structure (number of neurons and synapses) for better efficiency and inferencing performance is seen as a possibility and requirement for the development of ASNNs in the ML domain. The work presented in this thesis investigates biologically plausible learning techniques to enhance the pattern recognition capability without compromising the efficiency. The proposed algorithms are tested on spatiotemporal data classification.

In previous reviews on ASNNs, the potential for exploiting spatiotemporal data (ST) has been highlighted (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019). Space and time are ubiquitous aspects in much of the data collected in the real world. Preserving the features of these two aspects, space and time, during processing is a key requirement for successful pattern recognition (Atluri et al., 2018). Moreover, to harness the real potential of smart technologies, it is vital to explore efficient processing techniques to make automated systems for diagnosis, prediction, and prescription in real time using ST. Electroencephalogram (EEG) is a form of ST with useful applications in the brain-computer interface (BCI), such as learning, gaming, health, and wellbeing (Calvo & D'Mello, 2010). However, the current ML applications (i.e., non-ASNN) for EEG pattern recognition require pre-processing steps, like feature engineering and memory-intensive processing (Alarcao & Fonseca, 2019; Bastos-Filho et al., 2012; García-Martínez et al., 2017; Hosseini & Khalilzadeh, 2010; Pomer-Escher et al., 2014)which hinders efficient training and inferencing.

Taking EEG classification as an application area, this thesis attempts to optimise ASNNs by examining the impact of ASNN size on robustness and efficiency. Mental stress was chosen as a special application area of this research, along with motor movements and emotions. Mental stress has become a major concern because of its impact on both mental and physical health (Crowley et al., 2011; Wu et al., 2019; O'Connor et al., 2021). Many researchers have examined the effects of mental stress on human

cognition and performance and often suggest it to be a global issue (Seo & Lee, 2010; Epel et al., 2018). Mental stress recognition is the initial step in successful stress management, and several studies have proposed traditional ML techniques using EEG to detect mental stress (Betti et al., 2018; Khosrowabadi et al., 2011; Subhani et al., 2017; Y. Zhang et al., 2020). However, these approaches most often require pre-processing techniques that threaten generalisation capability, which in turn results in ML performance instabilities (Chikara & Ko, 2019; Ko et al., 2020), or when dynamic data is presented where feature representation changes over time (Hammoodi et al., 2018). EEG, in contrast to subjective psychological measures like questionnaires, is an information-rich data source that is more appropriate for mental stress recognition (Katmah et al., 2021). Therefore, the goal of this thesis is to determine biologically plausible learning mechanisms for SNN that improve ML performance and efficiency, with a focus on mental stress recognition from EEG data. The objectives formulated to achieve these goals are as follows:

1. Develop a feedforward SNN with biologically plausible learning techniques.
2. Examine structural adaptation techniques used by SNNs and their impact on ML performance.
3. Develop structurally adaptive SNNs for mental stress modelling.

1.2 Research Questions

1. How does the number of LIF spiking neurons in an SNN with Spike Time Dependent Plasticity (STDP) learning affect data classification performance?
2. How can brain-inspired plasticity techniques enable more efficient and accurate pattern separation in an SNN comprised of LIF neurons?
3. How can brain-inspired plasticity techniques be incorporated into SNNs for mental stress recognition on the fly?
    a. What knowledge can be extracted on stress using SNNs?

The first research question (RQ), addressed in Chapter 6 , investigates the relationship between the classification performance of EEG data and the number of neurons in an ASNN. A three-layer feed-forward ASNN is developed. Data modelling is optimised using a Differential Evolution (DE) algorithm with the objective of improving classification accuracy during each iteration by tuning the number of neurons in the hidden layer. The results of these experiments were explored through further experimentation to try and understand why certain SNN sizes were found to perform better than others.

In order to address RQ2, brain-inspired plasticity techniques were explored as a potential means of optimising network structure without the use of heuristics. In Chapter 8 , an ASNN with an ensemble of plasticity mechanisms for learning is developed that includes a spike regulation technique and experiments conducted to determine the contribution of neurons with different spiking rates to network pattern classification.

RQ3 is addressed through an investigation into the ability of an ASNN to recognize acute stress induced by audio comments (Chapter 9). An improved version of the ASNN used to address RQ2 was used and enhanced by including online learning. This novel ASNN prunes or grows itself based on the novelty present in the incoming data. Trained models are further investigated, using network behaviours, to discover relationships between acute and perceived stress. The network behaviour is interpreted to generate 'knowledge' rather than the network acting as a black box.

1.3 Thesis structure

This thesis is presented as an AUT Format Two thesis. The content of the study is presented in ten chapters, and an overview of the organization of the thesis is depicted in Figure 1.1. The terms artificial spiking neural networks and spiking neural networks are both used to identify computational (in silico) neural networks. Where the discussion is related to the biological brain, the terms biological, human, etc., preceded the term. The content of the chapters is as follows:

Chapter 2 begins with an introduction to spiking neurons and contrasts them with more traditional artificial neurons. Several types of artificial spiking neurons reported in the literature are evaluated in terms of biological plausibility and computational tractability. The discussion is further expanded to the literature that reports on ML applications of feedforward SNNs with different learning techniques. These ML applications are differentiated based on spike type and architecture and the advantages/disadvantages of the various approaches discussed. Based on this review, it was concluded that the ability to process multiple spikes, learn from long-term and short-term correlations of spikes, and an evolving architecture is essential for complex data modelling.

Chapter 3 first presents a comparison of brain imaging techniques and an introduction to EEG for the uninitiated reader. The discussion elucidates spike encoding algorithms available for analogue signals such as EEG and concludes with a review of encoding used by SNNs.

Chapter 4 provides a detailed description of the basic SNN framework developed for the experiments of this thesis. The preliminary experiments that investigated relationships between the number of neurones (Leaky Integrate and Fire neurons) operated with STDP learning are presented in this chapter.

Chapter 5 is comprised of a published journal article subjected "Emotional Stress Classification using Spiking Neural Networks". A network structure that produced the best results in the experiments reported in Chapter 4 is used to classify EEG data related to stressed and relaxed states. Apart from the learning strategies used, the SNN was supported with a Differential Evolution based hyperparameter optimisation process. Furthermore, the study looked at extracting knowledge, from trained networks, in the form of spatial connectivity patterns of stressed and relaxed brain signals. The classification results and knowledge extractions were compared with those reported by Machine Learning, Psychology and Neuroscience studies. In addition to selecting the appropriate number of neurons, this study revealed the importance of tuning the other hyperparameters (related to neurons and learning algorithms) on SNN for enhanced pattern recognition.

Chapter 6 is based on the journal paper "Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling" and contains work intended to answer RQ1. The chapter introduces a Differential Evolution based method to apply structural plasticity (changes the number of neurons based on learning performance) and hyperparameter tuning. The paper discusses experiments conducted on two separate EEG datasets and investigates the correlation between ML performance, network size and neuron excitability.

Chapter 7 discusses plasticity mechanisms related to memory and learning found in the mammalian brain and reviews biologically plausible learning techniques found in SNNs. The intention of this review was to investigate biologically inspired plasticity methods to replace the exhaustive computational method presented in chapter six. This chapter concludes with suggestions for improvements and provides a prognostic on biologically plausible learning techniques that could be developed for SNNs.

Chapter 8 is based on the journal paper "Ensemble plasticity and network adaptability in SNNs" which presents an SSN model to answer RQ2. Brain-inspired plasticity techniques reviewed in Chapter seven are used in this study with an information theoretic approach to control them. This chapter demonstrates the patterns of spiking behaviour that promote better ML performance, thereby enabling effective spike-based structural plasticity.

Chapter 9 is based on "Mental Stress Recognition on the fly using Neuroplasticity Spiking Neural Networks", a journal paper that presents a novel SNN architecture with brain-like plasticity techniques used for mental stress recognition. This architecture is used for online ML and addresses the RQ3. Furthermore, knowledge extractions are made on the connection between acute and perceived stress based on network behaviour. This chapter uses a dataset collected specifically for stress recognition.

Chapter 10 provides a summary of the work presented in this thesis, revisits the research questions, and highlights the key contributions. The chapter concludes with suggestions for future research.

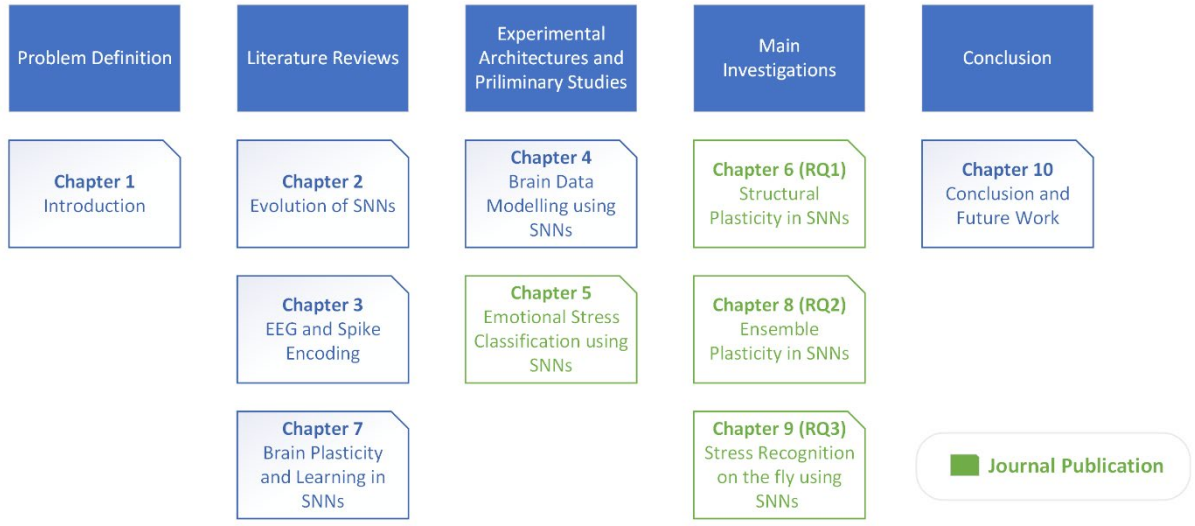


Figure 1.1: Organization of this thesis presented with the organization for each chapter. Journal publications are shown as a green clipped box.

Chapter 2 Evolution of Spiking Neural Networks

This chapter starts with a contextual understanding of spiking neurons with respect to the other artificial neurons and extends to review contemporary SNN algorithms. These algorithms are discussed with a fundamental focus on structural settings to highlight opportunities and limitations. This review also influenced the basic SNN framework development presented in Chapter 4 In this chapter: Section 2.2 compares different classes of artificial neurons, highlighting advantages and disadvantages, and Section 2.3 compares types of artificial spiking neurons. Section 2.4 discusses the SNN algorithms found in the literature. Based on the literature, a justification is provided for selecting a multi-layer, muti-spike output SNN architecture for the experiments reported in this thesis.

Given the focus of the thesis on developing more *biologically plausible learning* techniques, traditional SNN applications in ML, which are comparatively less biologically plausible when it comes to learning (e.g., ANN to SNN conversions), are excluded from the literature review presented in Section 2.4. The judgement of biologically plausible learning in SNNs was made by evaluating whether neurobiology or computational-simulation findings of biological systems such as the brain inspire the relevant techniques. It is important to note that biological plausibility does not guarantee better performance. However, the discoveries of how biological neuron systems work have influenced machine learning (ML) development throughout history (see Figure 2.1).

Figure 2.1 describes the development of ML in three avenues: Dynamic ML, Statistical ML and Artificial Neural Nets (ANNs). Compared to other avenues, ANNs are much closer to Neuroscientific and Psychology findings. Especially the concepts of Learning, Neuron function, Temporal coding and Population representation have already been adopted into ANNs. However, it is important to note that the level of abstraction between biological systems and artificial representations has further reduced in recent times.

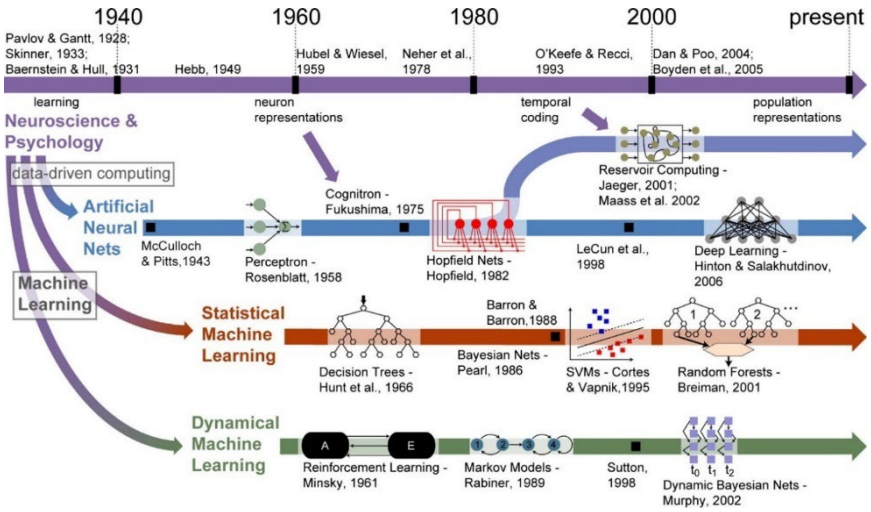


Figure 2.1: Avenues of machine learning development and impact of psychology and neuroscience

Note. From "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications", by James, C. D., Aimone, J. B., Miner, N. E., Vineyard, C. M., Rothganger, F. H., Carlson, K. D., Mulder, S. A., Draelos, T. J., Faust, A., Marinella, M. J., Naegle, J. H., & Plimpton, S. J., 2017, *Biologically Inspired Cognitive Architectures*, 19, 49–64. https://doi.org/10.1016/j.bica.2016.11.002, Copyright 2016 by Elsevier. Reproduced with permission.

2.1 Review Method

A *Snowballing* approach (Wohlin, 2014) is used to navigate and explore the literature presented in this chapter. This investigation begins from a selected set of seminal studies (Maass, 1997; Grüning & Bohte, 2014; Diehl & Cook, 2015; James et al., 2017b; Pfeiffer & Pfeil, 2018; Roy et al., 2019; Taherkhani et al., 2020; Tavanaei et al., 2019b), to which both backward and forward snowballing is applied (Wohlin, 2014).

Maass categorised artificial neurons and demonstrated the potential of SNNs for function approximation by comparing them to other non-spiking artificial neurons (1997). Diehl and Cook introduced the notion of biologically plausible SNNs for image classification applications and reported that the performance of these SNNs is on par with that of deep ANNs (2015). The remaining studies selected are detailed review studies that covered all the significant learning algorithms related to SNNs. These studies initiated the backward and forwards snowballing.

2.2 Development of Artificial Spiking Neuron

By the 1990s, a wide variety of artificial neurons had been developed by researchers. Maass grouped these different types of neurons into generations based on their transfer function (1997). Artificial spiking neurons were labelled as t*hird-generation* neurons. Maass's work included an experimental analysis of s*econd-generation* neurons (rate-coded neurons such as sigmoidal neurons) and highlighted the comparative advantages of artificial spiking neurons for function approximation. The rest of this section uses the same neuron classification of Maass and provides a contextual understanding of spiking neurons with respect to other types of neurons.

2.1.1 First-generation neurons

The operating function of an artificial neuron has changed from a basic threshold gate to a Dirac delta function. This transition made the spiking neuron more biologically plausible with the energy-efficient temporal coding feature. One of the most significant developments that initiated the advancement of ANNs was made by neuropsychologist Warren McCulloch and self-taught logician Walter Pitts in 1943 with the introduction of an artificial neuron known as the McCulloch-Pitts neuron. This type of neuron model was also referred to as a threshold gate. The McCulloch-Pitts neuron demonstrated *biologically favourable* characteristics such as spike behaviour (introduced through a digital *all-or-none* approach), summation of inputs within a short temporal time scale, and inhibitory and excitatory connections in a network setup (McCulloch & Pitts, 1943). McCulloch and Pitts also proposed the idea of *learning* in terms of an interconnected network changing its structure according to changes in the excitation thresholds of neurons. This neuron model was later adopted by ANNs in the form of multilayer perceptron (Rosenblatt, 1958), Hopefield nets (1982) and Boltzmann machines (Sherrington & Kirkpatrick, 1975). McCulloch-Pitts neuron can compute all Boolean functions in a single hidden layer perceptron setup. Apart from replicating Boolean functions, early applications of artificial neurons included reflex movement modelling (Baernstein & Hull, 1931) and maze-navigation (C. Shannon, 2009).

$$y(x) = \begin{cases} 1, & if \sum w_i x_i > t \\ 0, & otherwise \end{cases} \quad (1)$$

Equation (1) gives the mathematical representation of the McCulloch-Pitts neuron. This neuron functions in fixed clock cycles (i.e., it is synchronous), and if the summation of the inputs x_i, multiplied by the corresponding weights w_i reaches a threshold t , the output $y(x)$ becomes 1. If the threshold t is not reached the output *y(x)* is 0. The function of this neuron is illustrated in the schematic diagram given in Figure 2.2.

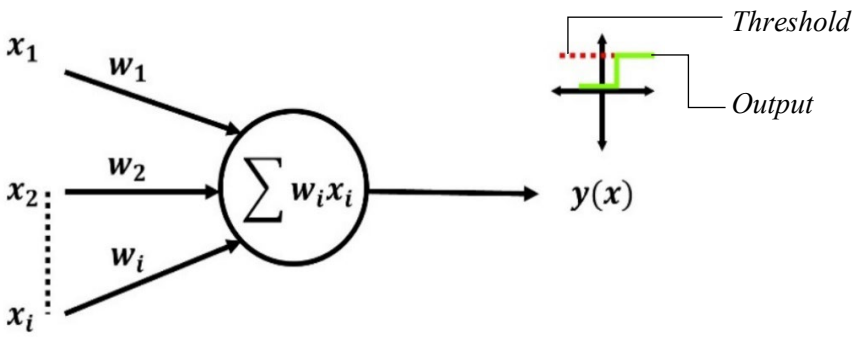


Figure 2.2: Schematic diagram of McCulloch-Pitts neuron

The McCulloch-Pitts neuron demonstrates biological plausibility to a certain extent since it operates in one or zero protocol and can generate voltage impulses similar to what is found in biological neurons. However, unlike the biological neurons, McCulloch-Pitts neurons function on synchronous clock cycles and use partial rate coding where the magnitude of the incoming signal influences the neuron activation (but the output signal magnitude is not representative of the input signal magnitude). Also, it does not consider the time factor in data processing (the time stamps of inputs and previous outputs do not influence the current output). Therefore, McCulloch-Pitts neuron is more energy-consuming, and its information encoding capacity is less compared to a biological neuron.

2.1.2 Second-generation neurons

Since the 2000s, machine intelligence has grown exponentially and, in some cases, even outperformed human intelligence on specific tasks owing to ANNs developed using second-generation neurons. The victory of the AlphaGo program over the world's rank number two Go player Lee Sedol in 2016 is one such historical event (C.S. Lee et al., 2016). When compared to the first generation, the second generation of neurons used continuously differentiable transfer functions where information is encoded using rate coding, producing a real value output. ANNs with second-generation neurons produced better function approximation compared to the first-generation with partial rate coding. In addition, the gradient decent backpropagation (BP) algorithm for supervised learning could be used due to the continuously differentiable nature of the transfer function (Rumelhart et al., 2013). This contributed massively to the success of ANNs in ML applications. Some of the most used activation functions include sigmoid, radial basis, ReLu, and softmax (refer Figure 2.3). Apicella et al. conducted a comprehensive survey on the modern activation functions of second-generation neurons and introduced a categorization (Apicella et al., 2021). The categorization included fixed and trainable (adaptable) transfer functions, where the authors highlighted the ability of trainable transfer functions to replace layers of fixed transfer function neurons. According to the study, a Deep Neural Network (DNN)[1] of trainable neurons are much more suitable for complex function mapping. Furthermore, such networks are comparatively more biologically plausible than neurons with fixed transfer functions.

[1] A network with at least two hidden layers with non-linear transfer functions is generally considered to be a DNN (Pfeiffer & Pfeil, 2018)

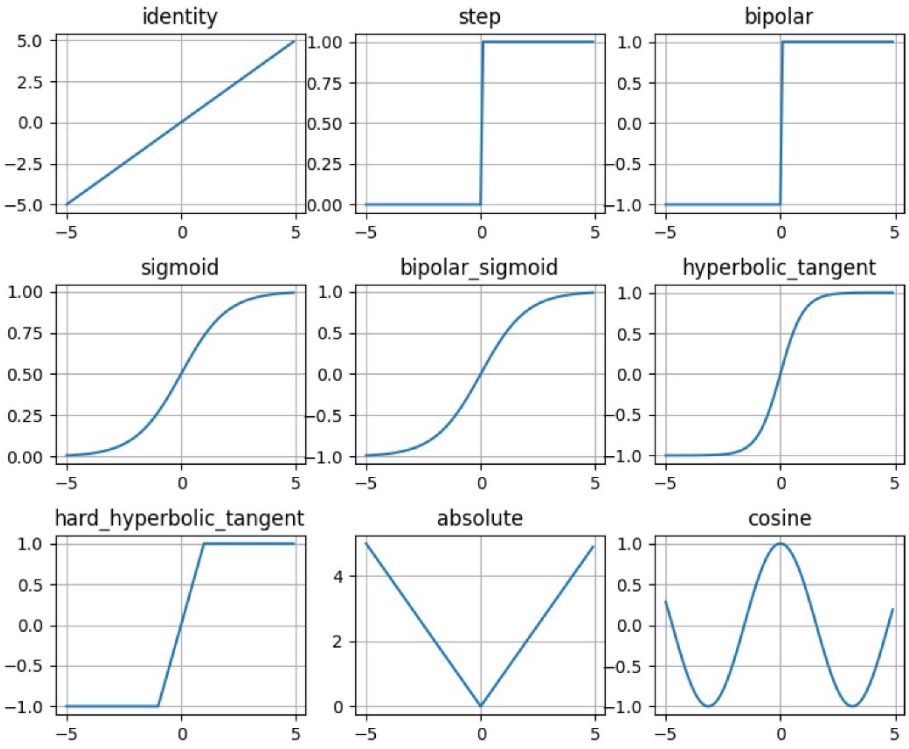


Figure 2.3: Activation functions of second-generation neurons

Note. From "A survey on modern trainable activation functions. Neural Networks" by Apicella, A., Donnarumma, F., Isgrò, F., & Prevete, R., 2021, *Neural Networks*, 138, 14–32. https://doi.org/10.1016/j.neunet.2021.01.026, Copyright 2021 by Elsevier Ltd. Reproduced with permission.

Despite the remarkable success of second-generation neurons in ML applications, the energy efficiency and computational power of these networks have not been able to reach their full potential compared to biological neural networks (K. Roy et al., 2019). As an example, the complex functions of the brain that includes control, movement, reasoning and recognition are performed with an energy budget of ~20W (Cox & Dean, 2014). Whereas the recognition task between 1000 objects in a general computer using second-generation ANNs consumes 250W (Milakov & Devtech, 2014). For ANNs to progress towards realising the power and efficiency of biological networks, the transfer function of the neuron and learning algorithms must be improved beyond what is used in second-generation networks. One possible avenue is to draw inspiration from neuroscience and psychology, which have made detailed discoveries on how biological systems such as the brain functions.

2.2.1 Third-generation neurons

The anatomical understanding of the biological neuron dates back to the 1880s and pioneering work by Santiago Ramóny Cajal and Wilhelm Waldeyer. Cajal's detailed sketches of nervous tissue revealed many insights into the structure of individual neurons and circuits of neurons. Cajal's work formed the basis for Wilhelm Waldeyer to initiate *neuron doctrine* in which Waldeyer suggested that nervous tissues were composed of discrete cells called *neurons.* Later, researchers expanded on this knowledge and found dendrites, soma, and axon as major parts of a neuron (see Figure 2.5). The dendrites act as the input channels, the soma being the processing unit that converts the input into non-linear output signals distributed to other neurons via the axon. Although the anatomical understanding of neurons was established in the 1800s, functional aspects of biological neurons began to be understood in the 1950s.

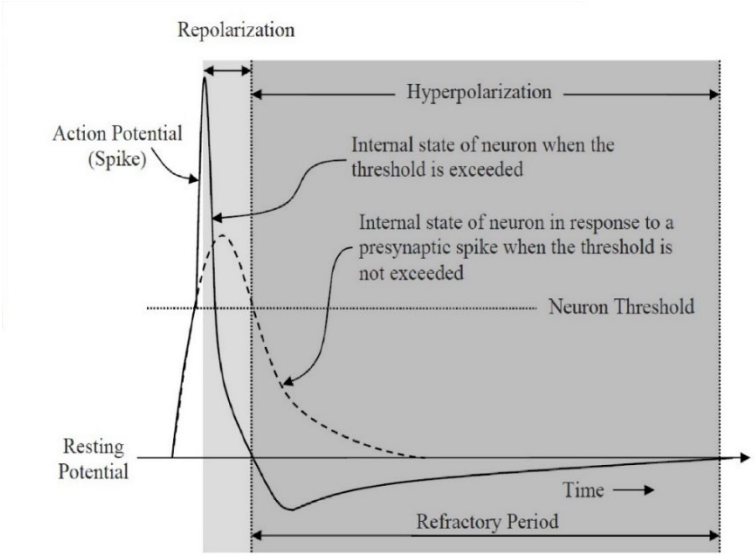


Figure 2.4: Basic behaviour of a spike

Note. From "Third Generation Neural Networks: Spiking Neural Networks" by Ghosh-Dastidar, S., & Adeli, H., 2009, *Advances in Intelligent and Soft Computing*, 61 AISC, 167–178, https://doi.org/10.1007/978-3-642-03156-4_17, Copyright 2009 by Springer, Berlin, Heidelberg. Reproduced with permission.

One of the earliest attempts to mathematically model the function of a biological neuron was executed by Hodgkin and Huxley (1952). They modelled the *action potential* (spike) generation of a cell membrane taken from the squid axon. Figure 2.4 provides a detailed representation of an action potential. When the input provides sufficient stimulation to reach a Neuron Threshold, an action potential is produced. Thereafter, the neuron enters a refractory period where the neuron refrains from

producing spikes (see Figure 2.4). This idea of action potentials paved the way for more biologically realistic artificial neurons to be explored that modelled the membrane potential and spike generation mechanisms at different levels of detail. The key function that sets *biologically plausible* artificial neurons (third-generation) different from other neurons is the use of discrete, asynchronous time-stamped spikes for information encoding. Many other neuroscientific studies have confirmed the existence of this spike function in biological neurons (Bialek & Rieke, 1992; Gerstner et al., 1996; M.Abeles, 1991; S. J. Thorpe & Imbert, 1989). Since time is a continuous function and spike is a fixed cost activity, the ability to encode a large amount of information with increased efficiency and reduced computational cost made ASNNs highly appealing to the ML community. These capabilities were also demonstrated by Maass by comparing ASNNs to networks with sigmoidal (second-generation) and threshold unit (first-generation) neurons (Maass, 1997). Maass concluded that ASNNs are able to approximate functions with fewer neurons than sigmoidal and threshold-gate neurons.

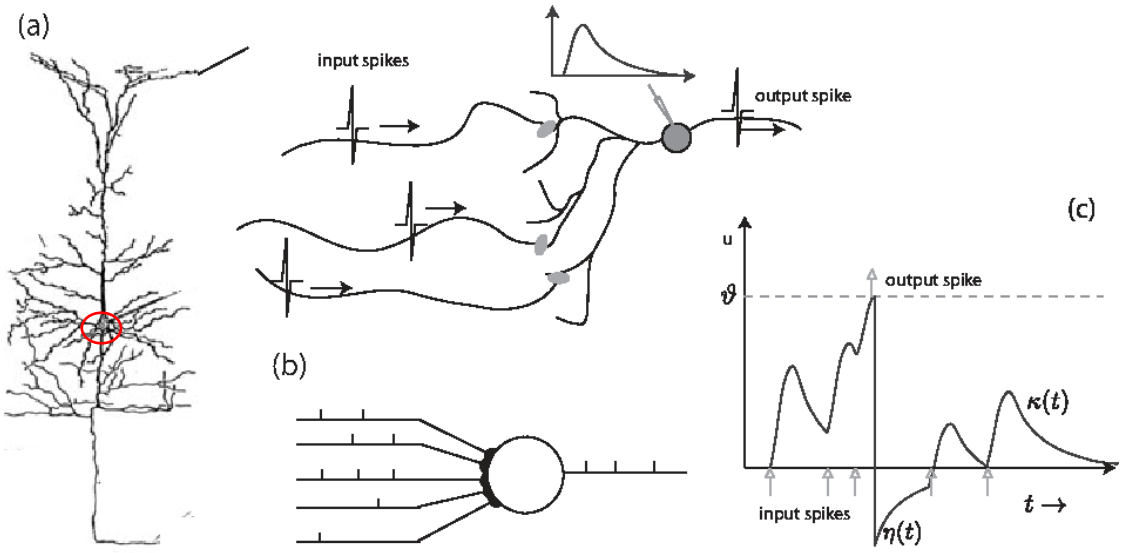


Figure 2.5: (a) Drawing of a biological neuron. Soma (circled), axon extending from the soma and branching into dendrites. (b) Computational representation of incoming spikes producing output spikes (c) Graph indicating how input spikes impact membrane potential and finally reach the threshold θ to produce a spike.

Note. From "Third Generation Neural Networks: Spiking Neural Networks" by Ghosh-Dastidar, S., & Adeli, H., 2009, *Advances in Intelligent and Soft Computing*, 61 AISC, 167–178, https://doi.org/10.1007/978-3-642-03156-4_17, Copyright 2009 by Springer, Berlin, Heidelberg. Reproduced with permission.

The computational power and efficiency found in spiking neurons are primarily caused by the asynchronous spike encoding strategy. Figure 2.5 illustrates how biological neurons communicate with

spikes and how it is typically represented computationally. In biological neurons, the spike communication happens via synapses which are the connection points between dendrites stemming from different neurons (see Figure 2.5a) where input spikes are proceeding towards synapses. As shown in Figure 2.5: Computational representation of the biological neuron, pre-synaptic neurons (neurons before the synaptic junction) propagate spikes, and these spikes are accumulated at the post-synaptic neurons. When the input spikes cause the post-synaptic neuron to exceed its spiking threshold voltage, an output spike is produced.

In biology, synapses pass either electrical or chemical information between the pre and post-synaptic neurons[2]. In a chemical synapse, when the action potential arrives, neurotransmitters are released through a biochemical process. Once these neurotransmitter molecules reach the postsynaptic neuron, a dedicated set of receptors of the postsynaptic cell membrane opens to let the ions flow into the cell through specialised ionic channels. This ionic influx into the postsynaptic neurons changes the membrane potential converting the chemical signal to an electrical signal. In an electrical synapse, the ions directly flow from presynaptic to postsynaptic neuron without a chemical to electrical conversion. Therefore, the signal transmission of the electrical synapse is faster than the chemical synapses (Cline, 2001). The unifying property of both types of biological synapses is activity-dependent efficacy.

This idea of synaptic efficacy is represented using synaptic weights in ASNNs. The alteration process of these synaptic weights according to incoming data is often referred to as *learning.* When it comes to ML, the discrete nature of spikes restricted the straightforward use of back-propagation-based learning technique in ASNNs since spikes are non-differentiable (Grüning & Bohte, 2014; Pfeiffer & Pfeil, 2018; Tavanaei et al., 2019). Therefore, the development of learning algorithms for ASNNs can be separated into biological and non-biological (i.e., efforts to incorporate backpropagation or statistical ML techniques) methods (These techniques are discussed in detail in Section 2.4).

When it comes to a selection of artificial spiking neurons, a variety is available in the literature with a diverse level of biological realisation (i.e., biological plausibility is increased if the neuron can demonstrate a higher number of neuro-computational features) and computational cost. Therefore, the selection of the suitable computational spiking neuron type is an important decision which is discussed in the following section.

[2] In the case of spikes being propagated from one neuron to another, the term presynaptic neuron refers to the neuron situated prior to the synapse and the term postsynaptic neuron refers to the one after the synapse

2.3 Types of Artificial Spiking Neurons

Artificial spiking neurons are mainly used in computational neuroscience studies to understand biological functions. Another emerging use is in ML applications that aim to process data for pattern recognition. Regardless of the use, connectivity algorithms and the neuro-computational features of the neurons act as the two main factors of consideration in network initialization. This section presents how biological plausibility differs between existing neuron models with respect to computational cost (see Figure 2.6). The computational cost of a neuron is dependent on the number of variables used and the floating-point operations employed for the neuron function. In ML applications, it is important to understand the computational cost involved, which is related to memory allocation and the number of computational operations carried out during data processing. This computational cost has implications on the implementation process (imposing hardware requirements) and requires data processing time. An important study conducted by Eugene Izhikevich compared the most commonly used spiking neuron models in terms of biological plausibility and their computational costs (2004).

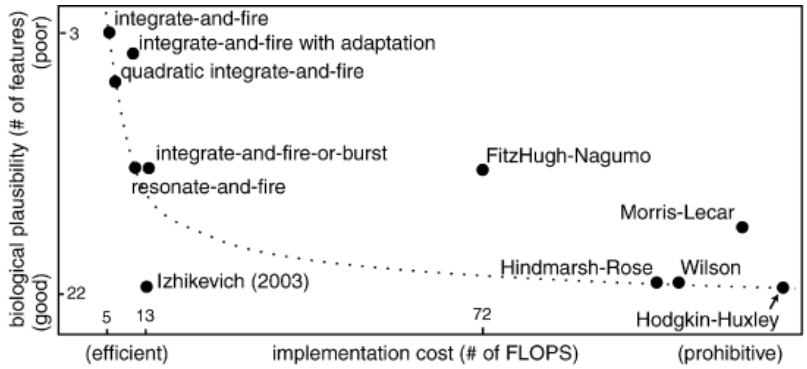


Figure 2.6: Artificial spiking neurons in terms of neuro-computational features and implementation cost

Note. From "Which Model to Use for Cortical Spiking Neurons?" by Izhikevich, E. M., *IEEE Transactions on Neural Networks*, 2004, 15(5), 1063–1070. https://doi.org/10.1109/TNN.2004.832719, Copyright 2004 by IEEE. Reproduced with permission.

The biological neurons have the capability to produce a variety of spiking behaviours. These different behaviours are defined as neuro-computational features in artificial spiking neurons. As seen in Figure 2.6, when the number of neuro-computational features produced (biological plausibility) increases, the computational cost also increases. The following sections analyse the Hodgkin-Huxley model (the neuron with the highest computational cost), Leaky-Integrate-and-Fire model (the neuron with the least computational cost) and Izhikevich model (the neuron with the highest cost-benefit) in detail.

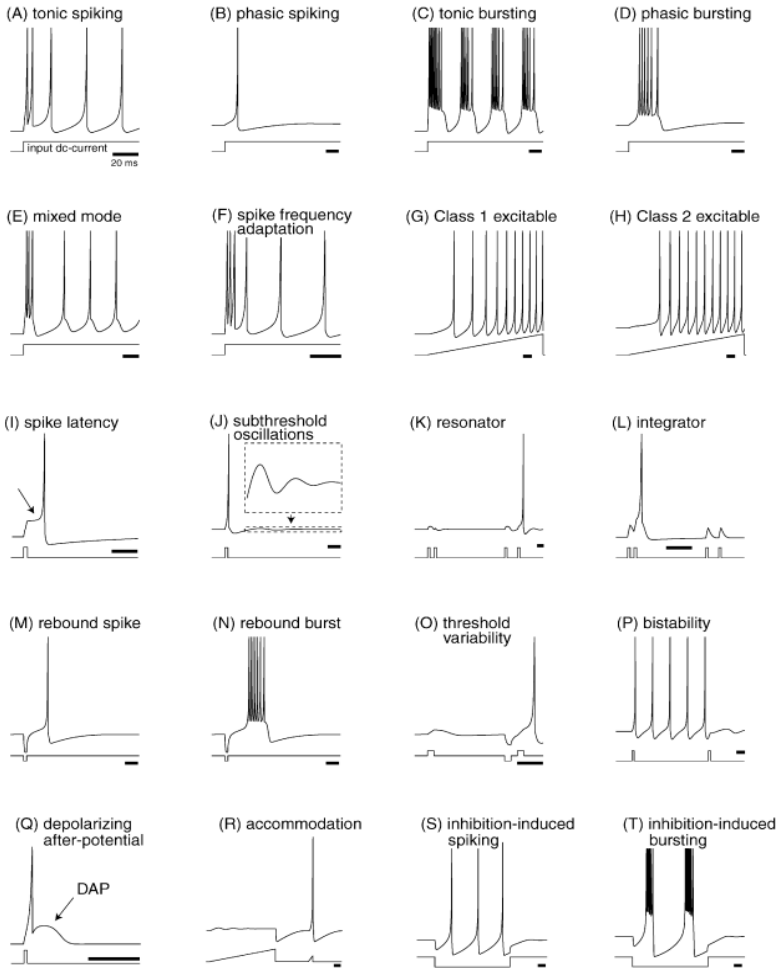


Figure 2.7: 20 Common neuro-computational properties that can be found in biological spiking neurons.

Note. From "Which Model to Use for Cortical Spiking Neurons?" by Izhikevich, E. M., *IEEE Transactions on Neural Networks*, 2004, 15(5), 1063–1070. https://doi.org/10.1109/TNN.2004.832719, Copyright 2004 by IEEE. Reproduced with permission.

2.3.1 Hodgkin-Huxley Model

The Hodgkin-Huxley model is considered to be one of the most biophysically accurate neuron models that can produce a wide variety of neuro-computational features. It is important to note however that the tonic, phasic and inhibition-induced bursting illustrated in Figure 2.7 are not part of the Hodgkin-Huxley model (Hodgkin & Huxley, 1952; E. M. Izhikevich, 2004). However, the implementational cost involved is extremely high, only enabling a handful of neuron activations in real-time simulations (E. M. M. Izhikevich, 2003). The function of the HH model is represented as parallelly connected resistor-capacitor (RC) circuits describing time-varying input and output currents. The varying input current is

a unification of total membrane and ionic currents, where the ionic current is calculated using three independent equations described in detail in Hodgkin & Huxley's original paper (1952). Therefore, the number of parameters and floating-point operations involved with HH operation is costly, particularly making it prohibitive for ML applications. Consequently, the ASNNs designed in this doctoral study have not used this type of neuron as one of the performance metrics is the computational cost.

2.3.2 Izhikevich Model

The spiking neuron model introduced by Izhikevich is said to demonstrate the computational features of HH whilst also being cost-effective, like the Leaky-Integrate and Fire model (LIF) (E. M. M. Izhikevich, 2003). In comparison to the HH model, the Izhikevich model is highly beneficial for computational neuroscience studies that require the reproduction of different spiking behaviours in a network of neurons. However, it is noteworthy that this model, unlike the HH model, does not provide the flexibility of incorporating separate ionic currents. This is due to the simplification that the Izhikevich model has achieved by applying bifurcation methods (Sejnowski & Poggio, 2007) to HH equations. Furthermore, the understanding of how different neuro-computational features contribute to information encoding and the values of variables and parameters required to get the neurons to produce different neuro-computational features stably is less known. The Izhikevich model's behaviour is mathematically expressed using the following two equations that use two variables and four parameters, both dimensionless.

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

$$\frac{du}{dt} = a(bv - u) \quad (3)$$

In these equations, v is the membrane potential, and u is the membrane recovery variable and I is the current. The dimensionless parameters include a representing the time scale of the recovery variable enabling faster or slower recovery, b describing the sensitivity parameter of u, to sub-threshold fluctuations of v. Two other parameters are used separately to set the after-spike reset value of v for fast high-threshold and slow high-threshold conductance respectively.

Suppose the computational cost and the ability to produce a wide range of neuro-computational features are the selection criteria for determining a neuron model. In that case, this model is an appealing option. However, ML applications require neurons to work with learning algorithms. There is not much knowledge on how Izhikevich neurons would behave with biological learning methods such as STDP, especially with the diverse spiking behaviours it can produce. This can pose a challenge in handling the parameters effectively to suit ML algorithms. Hence, it is sensible to use a simple and efficient form of a neuron model for ML applications that have been used often in previous ML studies.

2.3.3 Leaky integrate-and-fire neuron

Gerstner introduced the Leaky integrate-and-fire neuron (LIF) model, expressed mathematically with even fewer variables, parameters, and floating-point operations, than the Izhikevich model (Gerstner & Kistler, 2002). Therefore, LIF is one of the most cost-effective models to implement and operate in ASNNs (see Figure 2.8). Moreover, LIF models can detect coincidental input events (Abeles, 1982; König et al., 1996), which is an advantage in detecting salient events in data streams. But the LIF cannot demonstrate the variety of spiking behaviour to the level of HH or Izhikevich models.

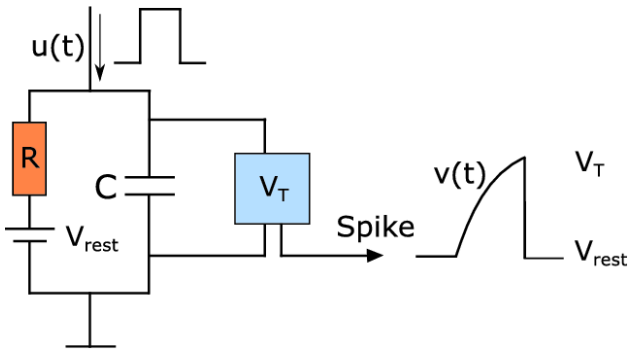


Figure 2.8: Leaky integrate-and-fire neuron representation using resistor-capacitor circuit

Note. Input spike represented by u(t), resistance by R, capacitance by C, resting voltage by v_{rest}, threshold voltage by v_T and instantaneous voltage by $v(t)$. The rate of voltage increase and decrease is determined by R and C values. From "Fundamental Limits of Forced Asynchronous Spiking with Integrate and Fire Dynamics" by Nandi, A., Schättler, H., Ritt, J. T., & Ching, S. N., *Journal of Mathematical Neuroscience*, 2017, 7(1), 11. https://doi.org/10.1186/s13408-017-0053-5, Copyright 2017 by Nandi, Schättler, Ritt, and Ching. Reproduced with permission.

$$\frac{dv}{dt} = RI_t - [(v_t - v_{rest})]/\tau_m \quad (4)$$

$$\tau_m = RC \quad (5)$$

The LIF model is expressed using a Resistor-Capacitor (RC) circuit (Figure 2.8). The membrane potential v is given by a first-order differential equation (Equation (4)) where: I_t represents input current, v_t corresponding voltage at time t, resting voltage as v_{rest} and resistance R and capacitance C. The multiplication of R and C is referred as the time constant τ_m (Equation (5)(4)). This constant decides the rate of membraned potential leakage. Once the membrane potential reaches a certain threshold θ, v_t increases infinitely according to a dirac delta function to produce a spike. Afterwards, the neurons go through a hyperpolarization phase where no spiking takes place during the refractory period (see Figure 2.4).

The ASNNs developed as a part of this thesis consisted of LIF neurons modelled according to Equations (4) and (5). As discussed previously, the LIF is the most computationally cost-effective model discussed (see Figure 2.6) (E. M. Izhikevich, 2004). Since this thesis is intended to expand the knowledge on how the number of neurons of an ASNN impacts ML performance and extend the same to develop structurally adaptive learning techniques, the level of cost-effectivity was a high priority. This decision was expected to produce faster processing of extensive ML experiments under general computing capabilities (Intel Core i7 with 16 GB memory). Furthermore, since this research also focused on developing online learning algorithms[3], the requirement for faster data processing was identified to be crucial once again and supported the LIF model selection.

Regarding biological plausibility, LIF is well behind HH or Izhikevich models (E. M. Izhikevich, 2004; E. M. M. Izhikevich, 2003). However, since the understanding of how different neuro-computational features relate to learning algorithms is little known under the current ML literature[4], the decision to use LIF for experiments over HH and Izhikevich was upheld. Moreover, the main aim of this study is to develop biologically plausible *learning algorithms* and not mimic biological neuron functions in detail.

[3] ML algorithms that train itself and produce outputs at the same time with minimum of no time lags
[4] Investigation on this aspect is beyond the research scope

Another reason for selecting LIF is that this neuron model is well understood and studied extensively in the literature on ASNNs and ML applications and learning algorithm development. ML applications that use LIF neurons have been used to process data related: images (Diehl & Cook, 2015; Hao et al., 2020; Wysoski et al., 2008), audio (W. Zhang & Li, 2019) and brain functions (Capecci et al., 2015; Doborjeh et al., 2019; Kumarasinghe et al., 2021; Tan et al., 2020; Taylor et al., 2014). This proved the feasibility of LIF neurons for ML tasks such as classification and provided much-needed knowledge on controlling the neurons with appropriate values for variable and parameter setting.

As discussed in this section, the model of the spiking neuron used can differ based on the purpose of the ASNN application. Since this thesis focuses on ML and pattern separation using ASNNs, the next important aspect is the network architecture and learning algorithms. The following section discusses and summarises learning algorithms found in different ASNN architectures. This provides a prologue to answer this thesis's second and third research questions related to developing ASNN learning algorithms.

2.4 Spike-based learning techniques in ASNNs

In ML applications, spiking neurons are connected by synapses (i.e., information-transmission structure) to form a single functional neuron or a network of neurons that make learning and memory formation possible. Computationally, the synapse has two properties; the weight that carries the efficacy of the synapse and a connection map that keeps track of the neuron connectivity detail. The connection map is a matrix with pre-synaptic and post-synaptic neuron addresses as rows and columns respectively. Learning algorithms enable the synaptic weights, connection maps, or neuronal properties to change according to a set of rules activated during the processing of input data. Other important aspects that impact the learning process include network architecture and information encoding protocols.

There are many ASNN architectures in literature. These may be classified as feed-forward, recurrent, and hybrid architectures (Taherkhani et al., 2020). Moreover, regarding information encoding, two of the most used approaches are temporal-based and rate-based coding. The rate-based methods include spike count, density, and population activity (Gerstner & Kistler, 2002). Temporal-based coding schemes include Rank-Order (RO), latency, phase, synchrony (Kasiński & Ponulak, 2006) and polychronisation (E. M. Izhikevich, 2006). Despite the wide variety of coding types researched, the role of such coding types in pattern separation is yet to be understood fully. Since this thesis is scoped to investigate feedforward ASNNs only, the following section focuses on learning mechanisms of feedforward ASNNs irrespective of the spike encoding. However, conversion methods that turn trained

ANNs with second-generation neurons into ASNNs are not considered relevant for the work presented in this thesis because the biological plausibility of such techniques is questionable.

The feed-forward ASNNs in the literature can be mainly classified into networks that either output a single spike or multiple spikes per neuron. Within these two classes, networks may use a single neuron, a single layer of neurons, or multiple layers of neurons (i.e., increasing structural complexity). In the following section, ASNN algorithms are reviewed regarding the output spike type (single or multiple) and ordered according to structural complexity. The review is carried out with an emphasis on biologically plausible learning, structural adaptation, the feasibility of online learning, and the ability to process dynamic data (such as brain data) since the research questions revolve around these topics.

2.4.1 Single-spike learning

The Evolving Spiking Neural Network (eSNN) is a single-layer ASNN used mainly for static image classification (Wysoski et al., 2008). The algorithm uses LIF neurons and an RO rule for learning. A new neuron map is produced each time a new sample is processed. The similarity between images is calculated using the inverse Euclidean distance of the synaptic weight vectors connected to the neuron map. If the similarity is high, the weight vectors are merged. This method highlights the importance of structural adaptation in ASNNs. The authors suggested that this algorithm is suitable for use as an online learning method. Online learning operates by performing network training and inferencing simultaneously. This requires the capability of the network to gain new knowledge without forgetting the already learnt knowledge but selectively discarding obsolete knowledge. The eSNN creates new neurons and merges if similar knowledge is found. But there is no selective method to eliminate obsolete knowledge. Since the work of this thesis also includes online learning and structural adaptation, eSNN provides insights into how structural adaptation can be adapted for online learning.

While this work was promising, the authors did not mention the possibility of a sample belonging to different classes that have a similar spiking order. Such an occurrence may result in misclassification, particularly in the presence of concept drift (changes in the relationship between input and output over time). This underlines the importance of information representation via multiple spikes and having sophisticated multi-spike learning algorithms beyond RO where timing between spikes is considered in addition to the order of spikes.

In contrast to the synaptic weight change approach adopted by (Wysoski et al., 2008), Pham et al. (2008) introduced the idea of a self-organised delay adaptation spiking neural network (SODA_SNN), which they used to cluster control chart patterns using a single layer of neurons. In the method proposed by Wysoski, the neurons operate as integrators on a longer time scale, whereas neurons of SODA_SNN operate as coincidence detectors (CD)[5] on a shorter time scale. In SODA_SNN, the delay functions of the neurons are adjusted while the synaptic weights are held constant. Ideally, the application of both weight adjustment and delays can improve the biological plausibility and computational ability of an ASNN. The method by which that should be done for better ML performance needs further investigation. However, since the SODA_SNN uses only the delay functions, a single sample must be passed through the network multiple times. Therefore, this method's computational cost and time spent on training are two disadvantages. Moreover, the inability to learn on a longer time scale (focusing only on shorter time scales using CD for learning) can be a concern, particularly if data types such as time series data need to be processed. Since this thesis aims to investigate brain data (spatiotemporal data, which are inherently time series) modelling, neurons with integrator functions are much more suitable than those having only delay functions.

Compared to eSNN and SODA_SNN, SpikeProp (Bohte et al., 2002) and its improved versions (Ghosh-Dastidar & Adeli, 2007; McKennoch et al., 2006; Shrestha & Song, 2015; Silva & Ruano, 2006; Verstraeten et al., 2005) can be considered as biologically implausible in learning due to the use of backpropagation. However, unlike direct conversion methods[6], they are spike-based backpropagation and have performed well in offline[7] ML tasks. Regarding drawbacks, the SpikeProp method suffered from the silent neuron issue, where neurons that did not fire during training became redundant since backpropagation was not applicable. Due to the lack of biological plausibility in learning, these methods are not investigated in detail here, but their key operational features are highlighted in Table 2.1 and Table 2.2.

In summary, the single spike output strategy consumes more time for model training if the classification is dependent on the output spiking pattern only because it requires multiple training epochs to refine the output spiking times (Pham et al., 2008). However, if synaptic weights are used for classification that gets modified over the entire period of sample processing, the use of multiple epochs for training may be avoided (Wysoski et al., 2008). When an ASNN is not geared to approximate a function with a

[5] Spikes happening within a shorter inter spike interval causes the given spike to fire, detecting the coincidental spikes
[6] ANNs with second generation neurons trained with traditional backpropagation converted into ASNNs
[7] Not online methods. Training and inferencing takes place separately

single pass (which is also the case with spike-backpropagation-based systems), it is not suitable for online learning strategies. Moreover, eSNN provided insights into how new neuron creation can be used to store new class information and merge neurons together when already learnt classes are presented to the network. This is an important insight into how structural adaptation can be used for online learning applications. However, sophisticated multi-spike learning strategies beyond RO learning are highly desired to tackle occurrences of concept drift found in dynamic and complex data types. In fact, its commonly understood that multi spike producing, multi-layer ASNNs would perform better with complex data due to the increased representation capability and the ability to recognise non-linear patterns better in the data (Tavanaei et al., 2019). The following section discusses learning algorithms that produce multiple spikes.

2.4.2 Multi-spike learning

In terms of biological plausibility, multi-spike coding is more realistic than single-spike coding (Sporea & Grüning, 2013). Moreover, its information representation richness is much higher (Borst & Theunissen, 1999). Therefore, researchers have adopted and developed biologically plausible multi-spike learning techniques.

One of the earliest examples of multi-spike learning algorithms using a single neuron, the Supervised Hebbian Learning method (SHL), was introduced by Ruf & Schmitt (1997). According to the input data class, this method uses a teacher signal to reinforce the neuron's specific target spike times. It was claimed that SHL has online learning capability since it uses locally available information for learning. However, the SHL did not guarantee convergence, and synaptic weights continued to alter even after the desired output spike times were produced. Another supervised learning algorithm was introduced as the Tempotron, which could only perform two class (binary) classifications (Gütig & Sompolinsky, 2006). Compared to SHL, this method lacked multi-class classification ability and biological plausibility since it used a gradient descent method[8] in the learning process.

A more biologically realistic method than SHL and Tempotron was introduced by Legenstein et al. (2005), which employed STDP[9] (Bi & Poo, 1998) for synaptic weight update. Additionally, a teacher signal was used to force the output spiking to a target pattern by altering the neuron thresholds. This pattern was then used for classification, where samples that produced similar spiking patterns were

[8] For controlling spiking patterns through neuron threshold altering
[9] STDP is a well-researched biologically plausible synaptic efficacy updating strategy that has been extensively researched both *in-vivo* and *in-vitro* (Markram et al., 2011)

clustered in the same class. However, this method also suffered from the same convergence issue found in SHL where weight updating continued even when the desired spiking pattern was already available (Lobo et al., 2020). ReSuMe is another biologically plausible supervised learning algorithm that, like Legenstein et al.'s approach, exhibited successful function approximation capability with spatiotemporal input data. ReSuMe used STDP and anti-STDP learning (Roberts & Leen, 2010) for weight alterations with feedback from the Widrow-Hoff type error calculation. The error calculation was based on the difference between the actual and the desired spiking output. ReSuMe provided a solution to the silent neuron problem found in SpikeProp and the continuous weight convergence problem of SHL.

One of the most important aspects of the methods introduced by Legnestein et al. (2005) and ReSuMe (Roberts & Leen, 2010) was the inclusion of biologically plausible STDP for synaptic weight update. Because of this inclusion, synapses are forced to operate in a competitive regime, and the weights of synapses are influenced by shorter-latency[10] and longer-latency[11] spike correlations (Song et al., 2000). This means the learning takes place in shorter and longer time scales, which is advantageous for pattern recognition tasks involving spatiotemporal data (Humble et al., 2012). As discussed in Section 2.4.1, learning spike information in both longer and shorter periods is biologically plausible and increases the ASNN algorithm's learning capability. However, maintaining firing homeostasis in the network is important to achieve these benefits in ML tasks with STDP learning (Watt & Desai, 2010). Moreover, the lack of understanding of what makes biologically plausible systems perform better at the fundamental level of spikes (S. Thorpe et al., 2001) makes controlling spiking activity under STDP challenging.

Except for the eSNN algorithm discussed previously, none of the multi-spike algorithms in this section has considered the structural adaptability of the ASNN for learning. Structural adaptation is important for online learning and handling concept drift found in complex data types such as spatiotemporal data because structural adaptation can be used effectively to retain already acquired knowledge and discard obsolete knowledge (Lobo et al., 2018; Minku & Yao, 2012). Investigating the impact of structural adaptability on offline and online ML performance are two key areas of this research. Some key literature that has used structural adaptation in offline and online learning algorithms has been discussed extensively in Chapter 6 .

[10] Shorter time durations between spike occurrences causes exponential increase of weights
[11] Longer time durations between spike occurrences causes exponential decrease of weights

Apart from synaptic weight modification, neuron threshold modification according to a teacher signal has been found in some methods discussed (Legenstein et al., 2005; Ponulak & Kasiński, 2010; Ruf & Schmitt, 1997). This is an important strategy that needs to be further investigated since the use of threshold alteration with STDP learning has been found to help maintain spiking homeostasis, which is crucial for network performance stability (Chen et al., 2013). Firing homeostasis refers to self-regulation of the firing activity of a neural circuit where neurons do not become 'too dominant' or 'too dormant'. This is a state of spiking equilibrium often tends to follow power law distribution (Beggs & Plenz, 2003).

Table 2.1

Summary of ASSN algorithms using single neuron architecture

Algorithms	Architecture	Neuron spike processing	Major learning technique	Learning type suitable
SHL (Ruf & Schmitt, 1997)	Single neuron	Multi-spike input/single-spike output	Supervised Hebbian	Offline
SpikeProp (Bohte et al., 2002) *(Ghosh-Dastidar & Adeli, 2007; McKennoch et al., 2006; Shrestha & Song, 2015)	Single neuron	Single-spike input/single output	Output layer spike time backpropagation	Offline
SPAN (Mohemmed et al., 2012)	Single neuron	Multi-spike input/analogue output	STDP and anti-STDP	Offline
ReSuMe (Ponulak & Kasiński, 2010)	Single neuron	Multi-spike input/multi-spike output	STDP and anti-STDP with Widrow-Hoff rule	Offline/online
Chronotron (Florian, 2012)	Single neuron	Multi-spike input/multi-spike output	Modified Victor & Purpura rule measuring spike train difference between input and desired output	Offline/online
Tempotron (Gütig & Sompolinsky, 2006)	Single neuron	Multi-spike input/single-spike output	Gradient-decent	Offline

Note. * These studies are improved versions of SpikeProp

Table 2.2

Summary of ASSN algorithms using single or multiple layers of neurons (feed-forward)

Algorithms	Architecture	Neuron spike processing	Major learning mechanism	Learning type suitable
Multiple SPAN (Mohemmed et al., 2013)	Single layer of neurons	Multi-spike input/analogue output	STDP and anti-STDP	Offline
eSNN (Audiovisual & Recognition, 2008)	Single layer of neurons	Multi-spike input/single-spike output	RO	Offline/online
SODA_SNN (Pham et al., 2008)	Single layer of neurons	Single-spike input/single-spike output	Hebbian type delay learning	Offline/online
SHL (Legenstein et al., 2005)	Single layer of neurons	Multi-spike input/multi-spike output	Supervised form of STDP	Offline
deSNN (N. Kasabov et al., 2013)	Single layer of neurons	Multi-spike input/multi-spike output	Spike driven synaptic plasticity (SDSP) and RO (with adaptive structure)	Offline/online
SpikeTemp (J. Wang et al., 2017)	Three-layer feed-forward	Multi-spike input/multi-spike output	Enhanced RO (with adaptive structure)	Offline/online
SRESN (Dora et al., 2016)	Two-layer feed-forward	Multi-spike input/multi-spike output	RO and Hebbian like learning (with adaptive structure)	Offline/online
OeSNN (Lobo et al., 2018)	Two-layer feed-forward	Multi-spike input/multi-spike output	RO and adaptive structure	Online

2.5 Summary

The literature reviewed provided support for the idea that combining the computational cost efficiency of the LIF neuron with the ability of STDP to learn over different time scales would be suitable for efficient EEG pattern recognition and stress classification. Moreover, using STDP supports the underlying theme of developing biologically plausible learning techniques for spatiotemporal data classification.

Furthermore, the structural plasticity in eSNN and its importance in online learning made it a potential option to be used as a classifier. Since multilayer multi-spike processing ASNNs performed well with complex temporal data, the decision was made to develop a three-layer, feedforward ASNN architecture that consisted of LIF neurons, STDP learning and a structurally adaptive classifier. The previous SNN algorithms used for ML applications do not discuss the impact of the number of neurons in the hidden layer operating with STDP. Therefore, the common approach is to set up the hidden layer arbitrarily. Here, arbitrary settings can cause overutilisation or underutilisation of computer resources. Chapter 4 presents the preliminary study of a series of studies designed to address the knowledge gap stated.

Chapter 3 Electroencephalography and Spike Encoding in SNNs

This chapter is intended to introduce EEG data and analogue to spike encoding methods. Since the experiments of this thesis use, EEG data for SNN evaluation and knowledge extraction, background understanding regarding EEG helps in results interpretation. Furthermore, a comparison between analogue to spike conversion protocols is discussed to justify the use of Address Event Representation (AER) in the experiments presented in the following chapters.

In this chapter, Section 3.1 provides a basic introduction to spatiotemporal data, and Section 3.2 highlights the advantages and disadvantages of EEG compared to other brain imaging techniques. Section 3.3 introduces EEG electrode placement, and Section 3.4 introduces spike conversion techniques for analogue signals. Then the chapter concludes with a brief discussion of other EEG data encoding techniques that are found in SNN literature (refer Section 3.5).

3.1 Spatiotemporal data and electroencephalogram

EEG is a form of spatiotemporal data with space and time dimensions. In comparison to classical data, the instances of a signal are correlated to each other. This property of spatiotemporal data is known as *autocorrelation*. Another property of spatiotemporal data is *heterogeneity* which highlights different data sources having different data distributions over time and space. Therefore, the functions of a data source change over time dynamically, both spatially and temporally. Thus, events between EEG channels and within the same channel can correlate (autocorrelation). Moreover, it cannot be assumed that the signals acquired through electrodes would maintain the same distributions over time or space (Heterogeneity). Hence, applying probabilistic assumptions made in mining classical data, such as Independent Identical Distribution, is invalid (Clauset, 2011). Consequently, such traditional assumptions on spatiotemporal data are expected to produce low accuracies and poor interpretability (Eklund et al., 2016). The solution would be to develop processing techniques that do not disturb the said properties of spatiotemporal data.

The SNNs being a time-based processing technique, it's potential to better exploit spatiotemporal data over other ANN techniques has been highlighted previously (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019). Therefore, using SNNs for classification and pattern recognition of EEG can be viewed as an ideal use case. Since all the experiments of this thesis use EEG, the following section provides a background for EEG technology compared to the other brain imaging techniques for the uninitiated reader.

3.2 Background to EEG technology

In a functional brain, an electrical potential is generated by clusters of activated neurons. This generated potential/voltage is captured over time by EEG using electrodes mounted to the surface of the scalp. Typically, the voltages generated from these active clusters of neurons are very low and recorded in microvolts. Operating in such low voltages, EEG faces challenges in acquiring noise-free signals. Facial muscle movements and signals generated from peripheral equipment can easily disrupt these low-voltage signals. The other challenge in EEG is source localisation (Grech et al., 2008), which increases from one participant to another due to differences in scalp surfaces. These issues can be avoided using Electrocorticography or Electrography, which uses direct signals from cortical surfaces or sub-cortical regions using depth probes. However, these invasive methods require specialised equipment and skilled clinicians to operate.

Compared to Positron Imaging Tomography (PET), Single Photon Emitted Tomography (SPECT), Functional Magnetic Resonance Imaging (fMRI), and Magnetoencephalography (MEG), EEG provides a more direct method of measuring neuronal activity that has a greater spatial resolution (Lystad & Pollard, 2009). Table 3.1 compares several common brain data collection methods against a selection criterion. Of these technologies, EEG alone expanded even into the consumer industry due to being non-invasive, cost-effective, and simple to use (LaRocco et al., 2020; Sawangjai et al., 2020). However, compared with consumer-grade EEG headsets, the research and medical-grade EEG headsets have a higher number of electrodes (typically silver chloride), which means that medical-grade EEG machines can acquire data with higher precision. The motor movement-related dataset used in this study is collected from a consumer-grade EEG headset (Taylor et al., 2014), whereas the human emotion-related dataset is from a medical-grade EEG headset (Koelstra et al., 2012).

Table 3.1

Comparison of brain imaging techniques

	PET	SPECT	fMRI	EEG	MEG
Neuronal activity measurement	Indirect	Indirect	Indirect	Direct	Direct
Biological process measured	Hemodynamic response	Hemodynamic response	Hemodynamic response	Neuroelectrical potentials	Neuromagnetic field
Invasive	Yes	Yes	No	No	No
Confined space	Yes	Yes	Yes	No	Yes
Radiation	Yes (0.5-2.0 mSv)	Yes (3.5-12.0 mSv)	None	None	None
Temporal resolution	1-2 min	5-9 min	4-5 s	<1 ms	<1 ms
Spatial resolution	4 mm	6 mm	2 mm	10 mm	5 mm

Note. In the temporal resolution, less than 1 ms is considered "Excellent" and above 1min as "Poor". For spatial resolution, 0-2 mm is considered "Excellent", and 8-10mm as "Reasonable" (Lystad & Pollard, 2009).

3.3 EEG electrode placement

EEG electrode placement on the scalp is a crucial step in the process of brain data acquisition. The correct placement allows for proper analysis, comparison, and reproduction. The 10-20 system is an internationally recognised procedure of placing scalp electrodes (Klem et al., 1961) which has been used in all EEG datasets discussed in this thesis. This method standardises the relationship between the electrode location and the underlying brain regions. The term *10-20* stems from the fact that adjacent electrodes are placed either 10% or 20% apart from each other. The aim is to ensure a consistent and equal electrode placement across the skull and that the electrode placements are proportional to the skull (see Figure 3.1).

The reader of this thesis needs to have a general understanding of the electrode placement on the scalp and the corresponding brain areas. This will enable the reader to understand the findings presented using exploratory analysis of network behaviours. These findings include heightened or lowered activations in certain inputs or correlations between inputs that can be discussed in terms of the brain area covered by the said inputs.

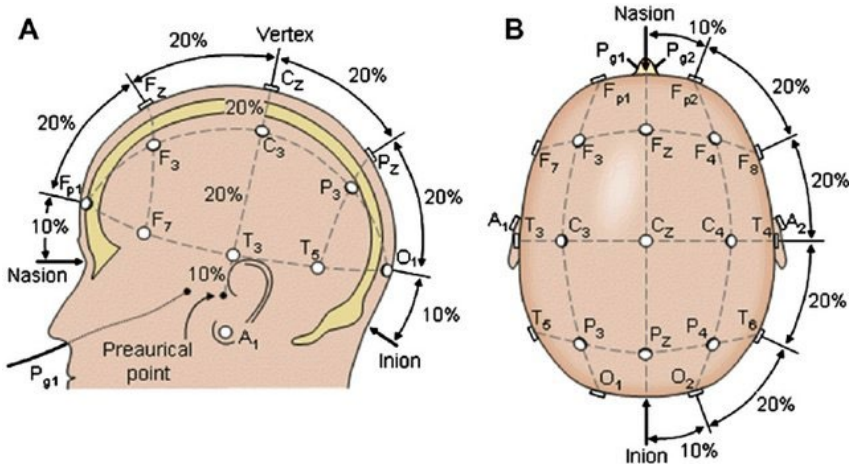


Figure 3.1: Electrode placement according to international 10-20 system. The proportionate distance between each electrode from Nasion to Inion left side view (Left). Electrode placement top view (Right). Electrode naming and corresponding brain areas: Fp (pre-frontal), F (Frontal), T (Temporal), P (Parietal), C (Central) and O (Occipital). Electrodes of the midline sagittal plane (FpZ, Fz, Cz,Oz). Odd and even numbers followed by letters refer to left-hemispherical and right-hemispherical electrodes respectively. The A1, A2 (also denoted as M1, M2) are connected to the mastoids (behind the outer ear) to provide contralateral reference to other electrodes.

Note. From "American Electroencephalographic Society guidelines for standard electrode position nomenclature", by Sharbrough, F., Chatrian, G.E., Lesser, R.P., 1990, Journal of Neurophysiology, 8, 200-2, PMID: 2050819, Copyright 1991 by American Clinical Neurophysiology Society. Reproduced with permission.

In order to handle EEG data in an SNN, the EEG signals must first be encoded into a spike format suitable for input into a spiking neuron. In the next section, methods from the literature are considered, and justification for using Address Event Representation (AER) in the SNN architectures developed for this thesis is provided.

3.4 Spike encoding

EEG signals are analogue voltage signals that require spike conversion. This spike conversion is an important step that should be carried out with a minimum loss of information. Loss minimisation can be achieved by testing reproducibility, and parameters can be optimised accordingly (Petro et al., 2019). Before optimisation, the most suitable spike conversion methods should be selected. Different spike encoding algorithms and their advantages and disadvantages are outlined in Table 3.2.

Table 3.2

Advantages and disadvantages of different encoding algorithms

Algorithm	Advantages	Disadvantages
Address event representation (Delbruck & Lichtsteiner, 2007)	Adjustable sensitivity for amplitude fluctuations Noise resistant Produces positive and negative spikes	The threshold factor needs manual tuning
Hough Spike Algorithm (HSA) (Hough et al., 1999)	Fast response Sensitive to fluctuations of unipolar data	Susceptible in encoding fast changes and small amplitudes closer to zero
Ben's Spiker Algorithm (Schrauwen & Van Campenhout, 2003)	Suitable spatial and temporal feature encoding Provides smoother output compared to HSA	Produces positive spikes only

3.5 Analogue EEG to spike conversions found in SNNs

3.5.1 Standalone conversions

NeuCube is an SNN that mimics the 3D architecture of the brain (N. K. Kasabov, 2014). Researchers have used the NeuCube for EEG modelling with different temporal spike encoding algorithms such as AER (Capecci et al., 2015; Doborjeh et al., 2019; Kumarasinghe et al., 2021; Taylor et al., 2014), BSA (Hu et al., 2014; N. Kasabov et al., 2013) and Adaptive Threshold Based (ATB) (Tu et al., 2017) methods to convert the analogue EEG to spikes. The sensitivity of AER to amplitude differences in analogue signals and its simplicity, which suits continuous data processing streams in real-time, has led to its use in many EEG applications. One of the disadvantages of AER is the need to set a spike threshold factor manually. This has been automated in ATB. However, the delay in processing makes it less suitable for real-time learning applications (Tu et al., 2017). The BSA technique is also used with EEG since the transformation of frequency, and amplitude features are smoother than HSA (Hu et al., 2014). However, BSA produces only excitatory spikes, making it unsuitable for an SNN that can process excitatory and inhibitory spikes (N. Kasabov et al., 2013). This limits the analogue EEG representations to two states of spike and non-spike, where AER can produce spike, non-spike and negative spike states resulting in an increased encoding capacity.

3.5.2 Conversions using network neurons

Other studies encode the analogue EEG signal using neurons that produce a rate-coded output (Chaturvedi & Khurshid, 2011; Ghosh-Dastidar & Adeli, 2007) or population-coded output using receptive fields (Dora et al., 2016; J. Wang et al., 2017). In the former, the rate of spikes produced is varied based on the input's salience or intensity. However, temporal-based coding is most suitable for SNNs that use spike time-dependent learning algorithms such as STDP. Population coding is an encoding method that can be used in the time domain. However, a single input signal in population encoding is represented by multiple spike trains. Therefore, multiple input neurons are required for a single-input EEG channel representation. This can add complexity to the network when working with a large number of EEG input channels.

According to the aim of this thesis of developing biologically plausible learning techniques, selecting a temporal encoding method was mandatory. Here, the population encoding method was a possible option but was avoided due to the added complexity it brings into the input layer of the SNN. Considering the unipolar spike production capability, noise resistance and fast response, which is suitable for real-time processing, the AER encoding algorithm was selected for the experiments of this study.

Chapter 4 Preliminary Studies on Brain Data Modelling using SNNs

This chapter introduces a three-layer SNN algorithm to process multiple spikes across multiple layers and classify spatiotemporal brain data. Although the literature review revealed the importance of structural adaptation of the output layer (final layer), none of the studies looked at the structural adaptation of the hidden layers operating with STDP. In a three-layer architecture, the single hidden layer maps the input data to a higher dimensional space to enable better pattern separation capability. Maintaining an appropriate number of neurons in the hidden layer is expected to support pattern separation capability better while effectively managing the computational cost. Therefore, this chapter contains the preliminary studies conducted to understand the impact of the number of neurons in the hidden layer on ML performance. For all the experiments discussed in this thesis, electroencephalogram data (EEG) is used to evaluate ML performances.

Two benchmark EEG datasets are used. The first dataset is related to motor movements (Taylor et al., 2014), and the other is to human emotions (Koelstra et al., 2012). These data sets were chosen because voluntary motor movements are found to be strongly associated with cortical regions (Sanes, 2003) and emotions with sub-cortical regions of the brain (Damasio et al., 2000). It was hypothesised that this would pose two different challenging environments for the SNN architecture used since the brain regions involved are different and the used different numbers of channels:14 (Taylor et al., 2014) and 32 (Koelstra et al., 2012).

4.1 The SNN learning algorithm

The importance of moving SNNs from fixed to adaptive structures was discussed in Chapter 2 Chapter 2 Network adaptivity as a part of learning is beneficial in acquiring new knowledge, sustaining already acquired knowledge and forgetting obsolete knowledge (Lobo et al., 2018). However, most SNNs have fixed structure methods, and only a handful of SNNs exist with adaptive capability as a method of learning (Draelos et al., 2017; N. Kasabov et al., 2016; Kheradpisheh et al., 2022; Lobo et al., 2020b; Schliebs & Kasabov, 2013; J. Wang et al., 2017; Wysoski et al., 2008). These attempts to make SNN structures adaptive include adding brand-new neurons or replacing existing neurons based on new knowledge availability at the classifier layer (Draelos et al., 2017; N. Kasabov et al., 2016; Kheradpisheh et al., 2022; Lobo et al., 2020b; Schliebs & Kasabov, 2013; J. Wang et al., 2017; Wysoski et al., 2008). However, these studies do not investigate structural adaptation in layers working with STDP.

In addition to being able to adapt the structure, it is important to note that maintaining the structure at an optimal level to lower the computational cost associated with the SNN involves keeping the number of neurons in the network as low as possible.

Structural adaptation through proliferation and pruning is another method that is found in the SNN literature (Dutoit et al., 2009; Rathi et al., 2019; Shi et al., 2019). Here, an overpopulated network is pruned after training is complete instead of applying structural adaptation as a part of learning (data-driven). However, these methods focus solely on reducing the computational cost by reducing computer memory usage by pruning synapses and neurons at the cost of performance. These methods are reviewed in-depth in Chapter 6 .

The implications of adding new neurons (with synaptic connections) or removing existing neurons from a hidden layer in an SNN (feedforward) employing STDP-like biological learning are less known. Therefore, setting up an SNN requires educated guesses, and typically the correct structure is developed through trial and error. Therefore, to investigate the effects of adding new neurons to a hidden layer, as part of this thesis research, a novel SNN learning framework was developed. The framework was developed using biologically plausible learning mechanisms, assuming it would produce better ML capability.

4.2 SNN Framework

This learning framework stands as the foundation for all the studies discussed in this thesis, and a detailed description can be found in the methodology sections of chapters that are publications[12] (A detailed explanation about the basic framework with associated equations can be found in Chapter 6). The ability to learn from complex temporal data, online learning readiness and biological plausibility in learning were key factors that were made during this development. The basic architecture of this SNN framework is illustrated in Figure 4.1. In brief, the algorithmic framework is as follows:

1. All the neurons of the SNN are LIF neurons (Gerstner & Kistler, 2002). LIF was selected over other artificial neurons considering biological behaviour, computational efficiency and the current understanding of how LIF functions with learning algorithms. Using a neuron with many neurocomputational features without the knowledge of how such features contribute to

[12] The foundational framework is furthered developed throughout the thesis and experiment results are compiled into journal articles

pattern separation may increase the functional complexity unnecessarily. Moreover, the frequent use of LIFs in similar studies makes functional knowledge of LIFs readily available. This helps set the network's hyperparameters and get the neuron to function in desired regimes.

2. The input layer processes multi-channel input spikes that are inhibitory and excitatory. Analogue EEG signals are converted to inhibitory and excitatory spikes using Address Event Representation (AER) algorithm (Delbruck & Lichtsteiner, 2007) before feeding the input layer of the SNN. This algorithm is used in silicon-retina applications, and multiple advantages are found that make it suitable for converting EEG signals to spikes. The ability of AER to adapt to amplitude fluctuations is a key advantage since special events in EEG are marked by amplitude fluctuations. Moreover, the ability to produce excitatory and inhibitory spikes and the noise-resistant supported the decision to select AER (The advantages and disadvantages of AER is summarised in Table 3.2).
3. The input layer connects to the hidden layer all-to-all. These synapses are trained unsupervised using the STDP rule (Bi & Poo, 1998) that processes inhibitory and excitatory spikes. The synapses are initiated with random weights between -0.1 and 0.1 following the Gaussian distribution. The use of unsupervised STDP enables efficient learning over short and long period time scales. This makes the learning process temporal and biologically plausible.
4. The hidden layer connects to an adaptive output layer through excitatory synapses (structurally adaptive output layer). The fundamentals of the output layer are adopted from the deSNN algorithm that uses both the RO rule (S. Thorpe & Gautrais, 1998) to initiate synaptic weights and Spike-Driven Synaptic Plasticity (SDSP) (Fusi, 2003) to update weights after initiation. When a sample is passed through the network, a new neuron is evolved with synapses and trained using RO and SDSP during the training phase. Each sample's Euclidean distance (ED) is calculated after that using the corresponding synaptic weight vector. During the testing phase, the same process is followed. However, the output neuron is deleted (as the testing sample processing is over) and only the Euclidean distance value is maintained. Based on the closeness of the testing samples (classes unknown) to training samples (classes known) calculated via EDs, classes are allocated to testing samples. This makes the classifier algorithm different to deSNN which uses a K-Nearest Neighbour (KNN) algorithm to cluster the classes at the output layer. This strategy was found to perform better with EEG data in the experiments done during the SNN development.

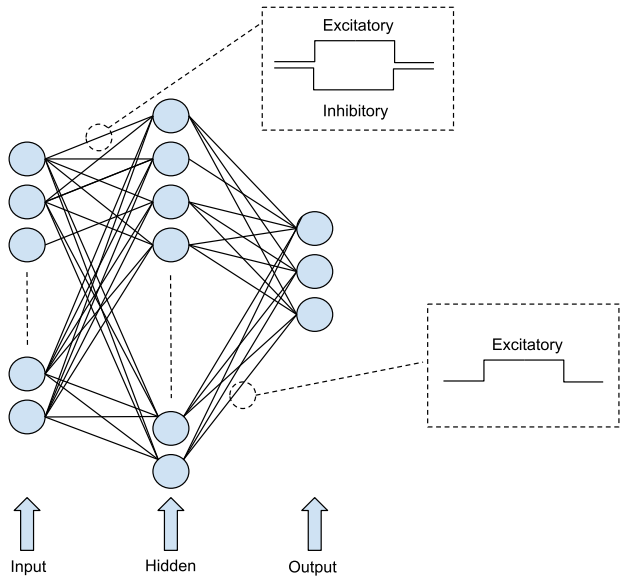


Figure 4.1: The basic SNN architecture developed

The advantage of using the output weight vectors for classification instead of output spiking patterns is that it avoids processing the same sample multiple times (This is discussed as the convergence issue in Chapter 2). The use of RO acknowledges the idea of spike ranking correlating to saliency in biologically plausible learning (S. Thorpe & Gautrais, 1998). The addition of SDSP increases the learning capability by considering every spike arrival after RO is applied. Finally, the deletion strategy of neurons during the testing cycle saves computational memory and maintains the output neuron repository constant but not forgetting the representation (ED of the testing sample).

This architecture is constructed based on the analytical discussions of feedforward SNN architectures in Chapter 2 This chapter highlights the importance of muti-layer, multi-spike processing capability to handle complex data. Incorporating both short-term (coincidence detection) and long-term spike correlation learning through STDP while accommodating excitatory and inhibitory activity has increased the unsupervised learning capacity of the architecture. Moreover, weight-based classification with the structurally adaptive output layer of the SNN makes it online learning ready. However, the challenge is to understand the role of hyperparameters (refer Table 4.1) and the size of the hidden layer neuron population required for better pattern separation capability.

Although it is possible to set hyperparameter values according to previous computational studies (Song et al., 2000), since input stimuli are different, the network may not spike. Therefore, the hyperparameter values given in Table 4.1 were selected using trial and error to produce spikes for all samples in the datasets discussed. All experiments discussed in this chapter use these hyperparameters.

Table 4.1

Hyperparameter settings of the experiment

Algorithm	Hyperparameter	Value
AER Encoder	f	0.5
LIF Neuron	v_{thresh}	0.5
	t_{ref}	5
	v_{rest}	0
	R	1
	C	10
STDP	A_+	0.001
	A_-	0.001
	τ_{pos}	10
	τ_{neg}	10
	Weight boundaries	0.1 to -0.1
Classifier	D_+	0.001
	D_-	0.001
	mod	0.8
	α	1

4.3 The EEG datasets

4.3.1 Wrist dataset

The wrist dataset (the dataset related to motor movement) is openly available (https://github.com/KEDRI-_AUT/neucube-cloud-sample-_file/blob/master/wrist_movement_eeg.zip) and has been used in a previous study (Taylor et al., 2014). The data were collected originally for a study that investigated how well an SNN could classify motor execution of wrist movement using EEG data. The data was from three healthy participants performing three different wrist positions: flexion, extension, or rest.

The EEG was collected by Taylor et al. using the 14-channel Emotiv Epoc Neuroheadset (Emotiv, 2021), and the electrodes were placed according to the international 10-20 system. Each recording lasted for 20 seconds at a sampling rate of 128 Hz. All data collections were done in an eyes-closed state minimising the artefacts, and no additional data pre-processing techniques were performed. This dataset contains 60 different EEG samples, 20 for each wrist movement category (flexion, extension, and rest)

performed by three healthy participants. A sample of this dataset is a 128 × 14 (i.e., time steps × channels) matrix. Before the SNN evaluation, "flexion" data samples were annotated as class one, "extension" as class two and "rest" as class three. All the samples were saved in .csv file format.

4.3.2 DEAP dataset

The DEAP dataset contains EEG readings of human emotions stimulated using music videos (Koelstra et al., 2012). The entire dataset consists of recordings from 32 participants collected while watching one-minute music videos. Forty different music videos were shown to each participant. The videos were selected to trigger emotions across the full valance arousal scale (Bradley & Lang, 1994). After watching each video, every participant rated the video according to valence and arousal using a self-assessment manikin. Each EEG sample was tagged with this valance and arousal score.

Each EEG recording was conducted using 32 channels at a 512 Hz sampling rate. The experiments of this chapter used the pre-processed DEAP dataset (publicly available at https://www.eecs.qmul.ac.uk/mmv/datasets/deap/download.html). In the pre-processed dataset, the signals had been downsampled to 128 Hz and filtered using a bandpass filter (4Hz to 45Hz). The EOG (electrooculogram) artefacts caused by eye movements and blinking had also been removed in the pre-processed dataset. A sample of this dataset is an 8064 by 32 (i.e., time steps by channels) matrix.

From the pre-processed data, a set of EEG data was extracted using the emotional stress definition by equations (6) and (7) adopted from Hosseini et al. (2010). This extraction was used since mental stress recognition is a part of the third research question in this thesis (refer 1.2), and these experiments provided information regarding the feasibility. The sample extraction process produced 125 and 104 EEG samples representing stressed (annotated class one) and relaxed states (annotated class two), respectively. Furthermore, To increase the processing speed and reduce the abnormalities of the input data, we used an averaging window (Golmohammadi et al., 2019) with a window size of 32. This processing resulted in a 32 by 252 matrix sample size, representing EEG channels and time points, respectively[13].

[13] The same sample extraction method is used in journal papers found in Chapter 5 and Chapter 6

$$Stressed = (Arousal > 5) \cap (Valence < 3) \quad (6)$$

$$Relaxed = (Arousal < 4) \cap (4 < Valence < 6) \quad (7)$$

4.4 Experimental Procedure

Before feeding the network with EEG data, the AER spike encoding algorithm (Delbruck & Lichtsteiner, 2007) was used to convert the analogue EEG signals into spike trains.

In each experiment, the number of hidden layer neurons is increased by one after completing a full evaluation phase until a performance saturation is reached. The point of saturation was determined based on a range of accuracy performances during five consecutive evaluation phases. If the accuracy range did not exceed 5% during five past consecutive phases, the search was terminated. Each full evaluation phase consisted of 30 train-test cycles. In each cycle, 70% of the data were used for training and 30% for testing. The sample selection was made randomly after encoding the spikes using AER. For each cycle, the network weight initiation was pseudo-random using a uniform distribution. All weights were restricted between -0.1 to 0.1. The sample selection and input mapping were also randomly determined for each train-test cycle.

The hyperparameter settings used are provided in Table 5.1. The starting number of neurons for both datasets was set to 47 following the results of an initial experiment. The initial results indicated an accuracy range of more than 40%. Therefore, searching below this point was ineffective.

4.5 Evaluation Measures

Classification performance was evaluated using *accuracy* and *robustness* metrics. Accuracy is defined by the ratio of correct classifications to total classifications (Urbanowicz & Moore, 2015). Robustness is defined as the network's resilience to withstand external perturbations and internal noise (Navlakha et al., 2018) induced by the randomisation of input mapping, sample selection, and weight initiation. Since a single evaluation consisted of 30 train-test cycles, a performance distribution is obtained. Here, accuracy is presented as an average accuracy over the 30 cycles and robustness is represented by the interquartile range (a higher interquartile range indicates lower robustness). The performance metric sample size of 30 was selected considering the rule of thumb related to statistical significance (Hogg et

al., 1977). By keeping the hyperparameters of the network and spike conversion techniques constant for both datasets, these experiments were designed to demonstrate the impact of adding new neurons on classification performance exclusively.

4.6 Results: Wrist dataset

The network showed a significant difference in terms of average accuracy and robustness at 102 neurons. This can be observed by analysing the boxes in Figure 4.2 where a single box represents the classification results over 30 cycles of training and testing using the same number of neurons.

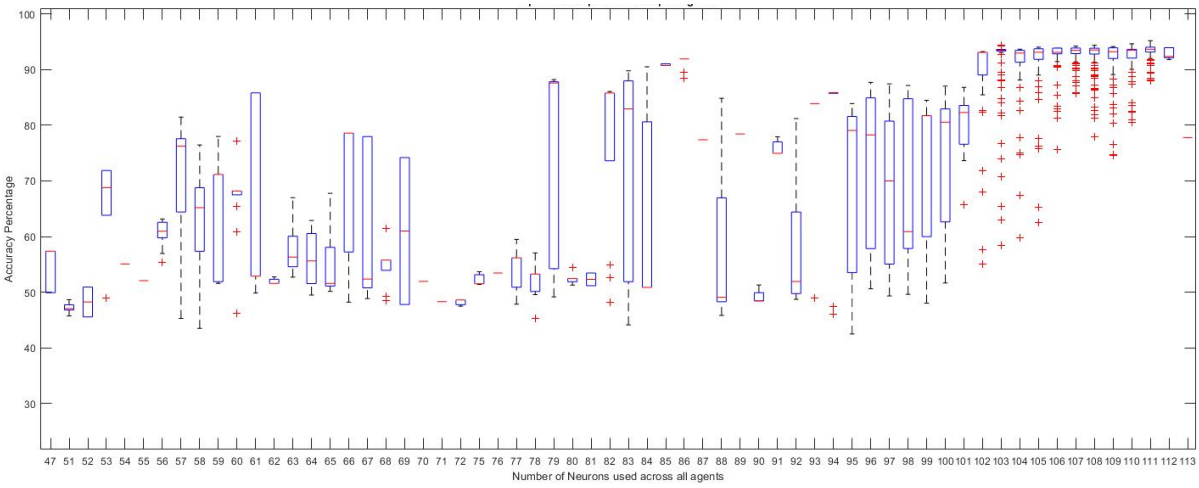


Figure 4.2: Average accuracy and robustness (interquartile range of a box) with increasing number of neurons – Wrist Dataset. Each box represents the results of 30 train-test cycles.

The average accuracy increased from 82% to 91.2% when the network size increased, by one neuron, from 101 to 102 neurons. The robustness of the network increased, indicated by the reduction of the interquartile range. The highest average accuracy was 93.2%, with an interquartile range of 0.7 for the network with 111 neurons. This accuracy is higher than that reported for a 1481- neuron SNN using the same data, which achieved an accuracy of 80% (Taylor et al., 2014). The SNN used by Taylor et al. used a small-world connectivity algorithm (Watts & Strogatz, 1998) to connect the LIF neurons and a 50/50 split for training and testing. Unlike the initial experiments reported in this chapter Taylor et al. employed a manual hyperparameter optimisation.

4.7 Results: DEAP dataset

The classification performance of the DEAP data was similar to that of the Wrist data in terms of a gradual overall increase in accuracy and robustness being observed as the number of neurons increased. However, the growth rate in classification accuracy was lower for the DEAP dataset than for the wrist dataset.

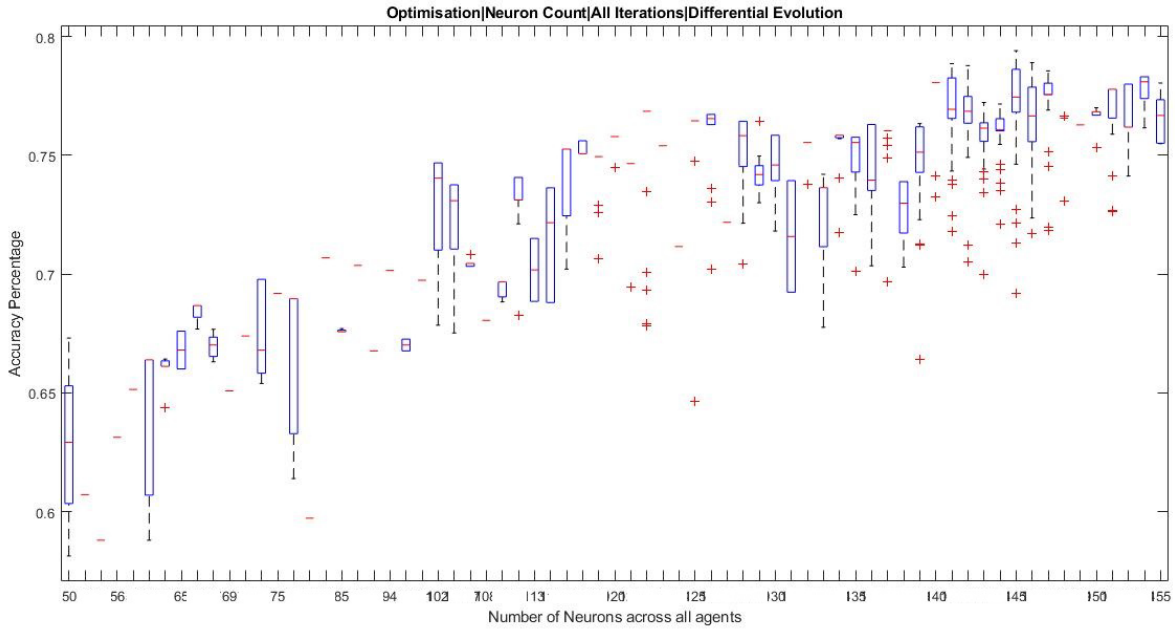


Figure 4.3: Average accuracy and robustness (interquartile range of a box) with increasing number of neurons – DEAP Dataset. Each box represents results of 30 train-test cycles.

The highest accuracy was 77.3%, with an interquartile range of 1.8 for 154 hidden layer neurons (see Figure 4.3). However, the average accuracy and robustness were reduced with 155 neurons in the hidden layer (76.5% average accuracy, 2.6 interquartile range). Although this is an interesting observation, the reasons behind these observations are not investigated in this chapter. How and why certain network structures perform better are questions that are analysed in terms of spiking activity in Chapter 6 .

The original study that used the same dataset reported 70.1% accuracy under five-fold cross-validation using a KNN algorithm with manual feature extraction (Hosseini & Khalilzadeh, 2010). The highest average accuracy achieved using this study's foundation SNN was 77.3% without manual feature extraction. Some other studies that also used the DEAP data achieved accuracies of 71.8% using KNN and three-fold validation (Shon et al., 2018) and 81.3% (García-Martínez et al., 2017) using a support vector machine (SVM) and ten-fold validation respectively. The apparent ability of the SNN to automatically detect features and perform better than a manual feature extraction method for emotional stress recognition is promising.

4.8 Conclusion

There are some limitations to this work. The observations above are restricted to the range of the number of neurons used for the testing (set by the search termination criteria). It is important to note that the performance may vary if the testing range was extended beyond the point where it was decided saturation had been reached. However, in an ML setup, the aim is to achieve the best possible performance level with the smallest network structure. When comparatively analysing the performance of the SNN experiments presented in this chapter, with the SNN algorithm presented by Taylor et al. for wrist movement data classification, it is evident that a bigger network does not always guarantee better performance. The performance seems to saturate or even reduce slightly as the number of neurons is increased in the hidden layer with STDP learning. This illustrates the importance of maintaining the minimum required number of neurons to ensure optimal performance.

Since this thesis also focuses on mental stress[14] recognition, testing the feasibility of the proposed architecture for stress recognition is a key step. Therefore, the emotional stress classification capability of the proposed algorithm is compared with the previously conducted studies using the same dataset. The results indicate that the SNN presented could perform at par with the previously proposed methods that used traditional ML techniques such as KNN and SVM with manual feature extraction methods.

[14] Mental stress can be separated into chronic, episodic or acute where the time frame of stress prolongation varies. The datasets used in this thesis are related to *acute* stress where stress is induced using audio and/or video stimuli for a short period of time.

Chapter 5 Emotional Stress Classification using Spiking Neural Networks

This chapter is comprised of a peer-reviewed journal article submitted to and published by *Psychology and Neuroscience.*

5.1 Reference:

Weerasinghe, M.M.A., Wang, G., Parry, D. (2022). Emotional stress classification using Spiking Neural Networks. *Psychology and Neuroscience*

5.2 Author contribution:

MMAW: 90%, GW: 5%, DP: 5%

M.W. developed the learning algorithms, designed the experiments, analysed the data, and wrote the manuscript. G.W. and D.P. supervised the research. All authors reviewed the manuscript and provided feedback.

5.3 Preamble:

The work reported in this chapter builds on the foundational feed-forward SNN architecture presented in Chapter 4 . The experiments in this chapter evaluate the ability of the SNN architecture to classify stressed and relaxed brain states using electroencephalogram (EEG) data and compare the performance of the architecture with stress classification methods found in the literature. Moreover, the network behaviour between stress and relaxed states is compared to identify spatial activation patterns.

Chapter 3 investigated how the number of neurons in the hidden layer impacted ML performance on models of two EEG datasets. In that chapter, it was found that networks with 154 neurons performed the best, with a classification accuracy of 77%. This can be attributed to a better spike representation from input to hidden layer with 154 neurons (The specifics of the spiking behaviours are discussed in Chapter 6). The performance metric of average accuracy and robustness were clearly affected by the number of neurons. The SNN showed an upward trend of average accuracy while having groups of peak performances. The variance of performance also followed the same trend. However, in these experiments, the other hyperparameters that directly affect the spiking of the SNN, related to LIF

neurons and the STDP rule (see Table 4.1), had fixed values. So how these hyperparameters affect performance and how they interact with the number of hidden layer neurons is unknown. Since classification accuracy depends on the spike encoding from input to the hidden layer, the hyperparameters could influence ML performance. To address this limitation, the experiments in this chapter use Differential Evolution (DE) to optimise the hyperparameters. Unlike in Chapter 4 , the optimisation was automated since six hyperparameters needed to be considered to improve classification performance. For these experiments, it was assumed that having optimum hyperparameter settings with fewer neurons would lead to better ML performance (in comparison to bigger networks with unoptimised hyperparameters). Therefore, instead of using the SNN with 154 neurons that delivered the highest performance in Section 4.5, a smaller network of 130 neurons was used based on the assumption that a smaller network would improve performance with optimised hyperparameters.

Approaches reported in previous literature for emotional stress classification require manual feature extraction. However, this is a challenging process that often requires subject expertise. Even then, one feature extraction method may not suit a different dataset, making the method's generalisation difficult. Hence, emotional stress classification methods with automatic feature extractions are highly desired. This chapter is based on a published journal article presenting an SNN that operates without manual feature extraction. The performance is compared with other methods found in the literature.

.

5.4 Abstract

This study examined the data modelling capability of spiking neural networks (SNN) in classifying stressed versus relaxed brain states using electroencephalogram (EEG) data. The spatiotemporal input dynamics were explored to obtain further knowledge regarding the two-brain states. A publicly available EEG dataset for emotion analysis using psychological signals (DEAP) collected from 32 participants (50% females) with an average age of 26.9 is used in this study. Firstly, data extraction is performed using a criterion that defines stress and relaxation states using self-reported valence and arousal scores. Two hundred eight such extracted samples were selected to train and evaluate a novel three-layer feedforward SNN. This SNN consisted of leaky-integrate and fire neurons and learned from incoming data using spike-time dependent plasticity (STDP) and dynamically evolving SNN algorithms. The SNN performance was evaluated using both fivefold cross-validation and a 60:40 training testing split. To explore spatiotemporal input dynamics, a specialised SNN architecture for brain data processing named NeuCube was used. The highest-performing model of the novel SNN algorithm produced 88% average accuracy (F1 score: 86.21%, Matthew's correlation coefficient: 0.78). This SNN outperformed traditional machine learning (ML) techniques without the use of manual feature extraction. Moreover, the input dynamics revealed higher prefrontal activation during relaxation states compared to stress states. The results showed the capability of the SNN algorithm to recognise stressed and relaxed states of the brain, using temporal learning techniques. Furthermore, the findings obtained from NeuCube suggested a potential approach for brain data analysis, setting SNNs apart from black box approaches used for brain data processing.

5.5 Introduction

Human emotional stress recognition is a growing area of research in psychology, neuroscience and computer science. The term stress represents the body's response to a perceived mental, emotional or physical distress that causes changes in the autonomic nervous system (Reisman, 1997). Therefore, the impact of stress on physical and mental health is inevitable. Several studies have shown the impact of stress on chronic diseases (Pickering, 2001), immunity (Lawrence, 2000) and cognitive capability (Decker et al., 1996). Therefore, many studies explore the possibility of recognising mental stress using physiological signals in combination with computational methods. Electroencephalography (EEG) is one such physiological signal type that is widely used for stress recognition (Katmah et al., 2021).

EEG signals are complex and carry a vast amount of data on brain activity. Therefore, using modern machine learning (ML) techniques to recognise brain activation patterns corresponding to mental stress states is a viable approach. From a variety of such ML techniques, the Support Vector Machine (SVM)

has demonstrated high accuracy in classification tasks of human emotions using EEG signals (T. Xu et al., 2018). Moreover, the recent success of artificial neural networks (ANN) in complex function approximation has also inspired ANN adaptation to brain data modelling (K. Roy et al., 2019). In a previous study, a deep neural network (DNN) and a convolutional neural network (CNN) were used separately to classify human emotions (i.e., Valence and Arousal) using EEG (Tripathi et al., 2017). These models demonstrated state-of-the-art accuracy for the classification of two (High and Low) and three (High, Normal and Low) classes of emotions. Moreover, Long-Short Term Memory (LSTM) for feature learning coupled with a dense layer was used for binary classification of emotions and reported higher average accuracy performances over traditional ML methods (Alhagry et al., 2017). The use of LSTMs in this study for temporal feature learning using raw EEG is an important step towards automated human emotion recognition.

However, the majority of the research studies conducted in this area require different feature extraction methods before the application of ML. This creates two implications, a) a tendency to lose information on spatial and temporal relationships due to data transformations (Al-Fahoum & Al-Fraihat, 2014) and b) limitations in generalisability due to the variety of feature extractions used (Katmah et al., 2021). In this study, we present a novel approach to classifying brain states of stress and relaxation using brain-inspired Spiking Neural Networks (SNN) without feature extraction. The validity of the proposed approach is tested using a publicly available DEAP EEG dataset (Koelstra et al., 2012). Moreover, we visualise network behaviour under training separately for the two brain states using NeuCube SNN architecture (N. K. Kasabov, 2014) to interpret the inferences made by the model. In this article, we use the term stress in general to describe emotional stress induced by external stimuli.

5.5.1 Electroencephalography and Spiking Neural Networks

EEG is becoming an increasingly popular method of attaining functional brain information due to advantages such as non-invasiveness, high temporal resolution and cost-effectiveness (Lystad & Pollard, 2009). The information in EEG represents the integrated voltage fluctuations occurring due to neuron activations in the brain from a scalp level. Therefore, EEG signals originating from different scalp regions carry both spatial and temporal information. Such spatiotemporal signals consist of autocorrelative and heterogeneous properties (Shekhar et al., 2015). SNNs process information using spikes and can be incorporated with biologically plausible learning techniques such as Spike Time Dependent Plasticity (STDP) (Taherkhani et al., 2020) to learn temporally, allowing these properties to be fully explored (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019).

5.5.2 Stress recognition using EEG

Studies conducted on emotional recognition used different physiological signals such as EEG, Electromyography, heart rate, Galvanic skin response, respiration rate and/or combinations (Alarcao & Fonseca, 2019). Our aim in this section is to only summarise the studies that used EEG and the same labelling of stress given in equations (8) and (9). From those selected, we have separated studies that utilised the DEAP EEG dataset and others. However, the performance comparisons will only be made with the priory mentioned. It is worth mentioning at this stage that all the methods discussed in this section have used feature extraction methods to classify stress data.

Hosseini presented emotional stress recognition using multimodal physiological signals, which included EEG (Hosseini & Khalilzadeh, 2010). The International Affective Picture System (IAPS) image database was used for stimuli with EEG channels located in FP1, FP2, T3, T4 and Pz. Wavelet coefficients and fractal dimension by Higuchi's algorithm and correlation dimension were used as features. A classification accuracy of 82.7% is reported using the Elman classifier. A similar study was conducted by Pomer-Escher to test the feasibility of detecting stress levels using EEG on stress-induced using the IAPS database (Pomer-Escher et al., 2014). The study does not present validation in terms of accuracy. However, it illustrates the possibility of detecting stress using alpha asymmetry in the frontal cortex from channels FP1, FP2, F3 and F4.

Bastos-Filho used power spectral density (PSD) and high order crossings (HOC) as statistical features with K- nearest neighbour (KNN) classifier for stress classification(Bastos-Filho et al., 2012). Experiments resulted in the best accuracy of 70.1% and 69.59% for the two feature extraction methods, respectively. In another study, Garcia-Martinez employed time series feature extraction methods, namely, permutation entropy (PEn), amplitude-aware permutation entropy (AAPEn) and quadratic sample entropy (QSEn) to produce a classification accuracy of 81.31% with SVM classifier (García-Martínez et al., 2017). Additionally, this study only used P3 and P4 channel data only. Moreover, an accuracy of 71.76% using the KNN classifier was reported in another study that used a multitude of feature extraction techniques. The accuracy mentioned was obtained using a genetic algorithm (GA) based feature selection method (Shon et al., 2018).

5.6 Method

In this section, we first introduce data preparation steps followed by the SNN architecture used for classification and experimental framework. Here we intend to develop a system that can automatically extract global temporal patterns of stressed and relaxed brain states. Secondly, we introduce the NeuCube SNN architecture (N. K. Kasabov, 2014) used to visualise network behaviour.

5.6.1 Data preparation

DEAP dataset consists of EEG recordings of 32 healthy participants, collected in an isolated room while watching 40 one-minute video clips, consequently annotated for valence and arousal using a self-assessment manikin (SAM). The recordings were originally conducted using 32 channels at a 512Hz sampling rate (Electrode connections to the scalp followed the international 10/20 system) (refer (Koelstra et al., 2012) for more details on the data collection process). Our experiments used the pre-processed version (downloaded from https://www.eecs.qmul.ac.uk/mmv/datasets/deap/download.html), which contained 32 channel data, downsampled to 128Hz, filtered with a 4Hz to 45Hz bandpass filter and artefacts removed (see Figure 5.1). Each EEG file sample of this dataset contained 8064-time points. To increase the processing speed and reduce the abnormalities of the input data, we used an averaging window (Golmohammadi et al., 2019) with a window size of 32. This processing resulted in a sample size of 32 by 252, representing EEG channels and time points, respectively.

$$Stressed = (Arousal > 5) \cap (Valence < 3) \quad \textbf{(8)}$$

$$Relaxed = (Arousal < 4) \cap (4 < Valence < 6) \quad \textbf{(9)}$$

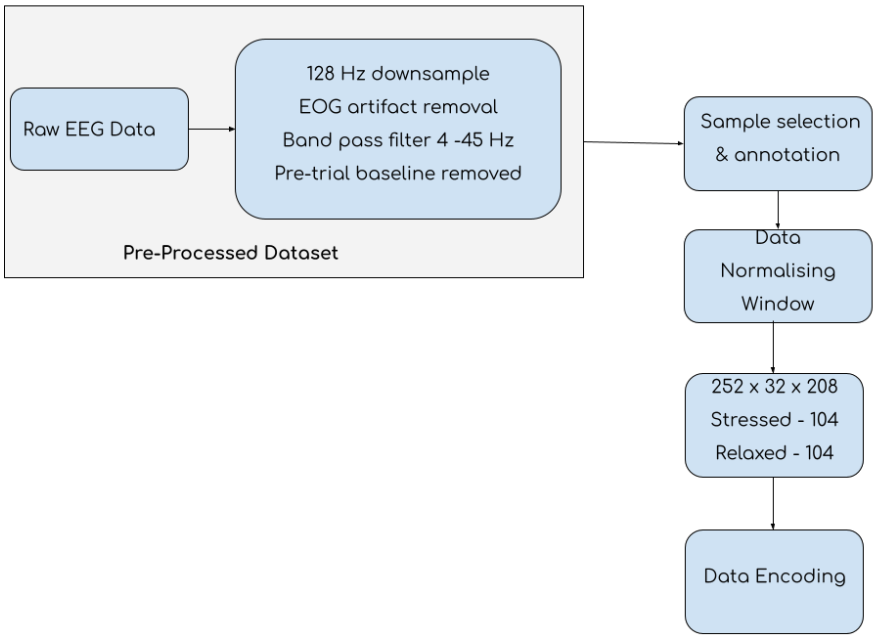


Figure 5.1: Overview of EEG sample preparation

Based on personal SAM ratings of videos, EEG samples were extracted and annotated according to (8) and (9) (Hosseini & Khalilzadeh, 2010). Therefore, each EEG sample label resembles a perceived level of valance and arousal. This sample extraction process produced 125 and 104 EEG samples representing stressed and relaxed states, respectively (Only 27 participants out of 32 indicated SAM ratings that were within the criterion. For the other participants, none of the videos prompted a SAM rating within the mentioned criterion). To balance the dataset, the stressed state was randomly under-sampled to 104, resulting in a balanced data set with 104 samples for each class.

Since SNNs process and communicate information with sparse and asynchronous binary spikes (Maass, 1997), all data samples were converted to their spiking equivalent using Address Event Representation (AER) algorithm (Delbruck & Lichtsteiner, 2007) before feeding the network. AER uses a temporal contrast procedure to define a threshold given by **(10)**.

$$Threshold = Median\,(dif) + f * Standard\,(dif) \quad \textbf{(10)}$$

Here, dif represents signal amplitude contrast between adjacent time points, and f is a user-defined factor. We set f to 0.5 during all our experiments based on pilot studies carried out.

5.6.2 SNN Architecture

We used a three-layer SNN architecture (see Figure 5.2) to classify the EEG data (This architecture was developed based on the JNeuCube framework, publicly available at https://github.com/Auckland-University-of- Technology/NeuCube-java). The first layer of the SNN consists of pairs of nodes (i.e., allocated per channel) that can propagate both excitatory and inhibitory spikes. The input nodes propagate spikes to the hidden layer of Leaky Integrate and Fire (LIF) neurons (Gerstner & Kistler, 2002). Input to hidden layer weights is updated using STDP, a biologically plausible, unsupervised learning algorithm (Song et al., 2000). The weight update strategy for the output layer was inspired by dynamically evolving SNN (N. Kasabov et al., 2013) that evolves with every sample propagated through the network.

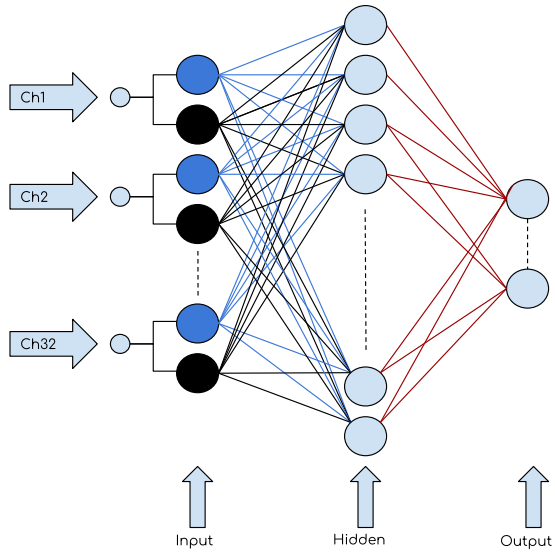


Figure 5.2: The spiking neural network used for the classification of EEG data

The function of a LIF neuron can be modelled using a resistor-capacitor (RC) circuit. Therefore, it can be mathematically represented as:

$$\frac{dv_t}{dt} = \frac{[RI_t - (v_t - v_{rest})]}{\tau_m} \quad \textbf{(11)}$$

In this equation, $\frac{dv_t}{dt}$ represents the change in membrane voltage v_t with respect to time t. Resistance is given by R, and current at time t is given by I_t. Resting voltage is denoted as v_{rest} and, τ_m represents the refractory time which is a constant. In an RC circuit, this is the multiplication of resistance and capacitance. When the instantaneous membrane potential v reaches a defined threshold value v_{rest} at time x, and if the previous spike was emitted at $(x - 1)$, the neuron emits a spike δ_x denoted by:

$$\delta_x = \begin{cases} 1 \quad if\ v = v_{thresh}\ and\ x > [\tau_m + (x-1)] \\ 0\ otherwise \end{cases} \tag{12}$$

The first stage of learning in the SNN uses STDP, which updates weight between post and pre-synaptic neurons based on the temporal synchrony of the spikes produced by each neuron. The mathematical formulation of the algorithm can be given as:

$$F(\Delta t) = A_+ exp^{(-\Delta t/\tau_{pos})}\ \Delta t > 0 \tag{13}$$

$$F(\Delta t) = -A\_ exp^{(\Delta t/\tau_{neg})}\ \Delta t > 0 \tag{14}$$

$$\Delta W_{ij} = \sum_{a}^{b} \sum_{p}^{q} F\ (t_i^m - t_j^n) \tag{15}$$

In (13) and (14), the function $F(\Delta t)$ refers to long term potentiation (LTP) and depreciation (LTD) respectively. LTP takes place if the post-synaptic neuron fires after the pre-synaptic neuron where the time gap between the firings, Δt becomes positive. It becomes negative if the firing sequence happens and vice versa, leading to LTD. Both τ_{pos} and τ_{neg} denotes positive and negative time windows for the synaptic modification. This was held at 10ms for our experiments. Positive and negative modification factors are given by A_+ and $A\_$ respectively. The cumulative weight change is denoted by ΔW_{ij} where post-synaptic firing is considered from a to b and pre-synaptic from p to q. Firing at a given time m by post-synaptic neuron is given as $t_i^m$and, pre-synaptic neuron firing at time n by t_j^n.

5.6.3 Experimental framework

During the training phase of the network, samples are randomly divided according to a 60:40 split, with 60% of samples for training and 40% for testing. Moreover, the input to hidden layer synapses of the SNN is initiated with random weights between -0.1 and 0.1, according to Gaussian distribution. In the training phase, all the samples are propagated through the network, and STDP weights are updated. This allows the network to learn all the temporal patterns in the data. Thereafter, each training sample is propagated over the network to train the deSNN classifier, where neurons are evolved for each sample. Since we have used 130 hidden layer neurons in this experiment, each sample will be

represented with a classifier weight vector of 130 elements with labels known. Thereafter the classifier seeks the closest training sample in terms of Euclidean distance and gives the same label to the testing sample. This method proved to be better than the clustering approach of deSNN since each of the brain signals was coming from a different participant with distinct variations.

5.6.4 Network optimisation

Once the network is initiated, we use Differential Evolution (DE) (Storn & Price, 1997) (i.e., DE/rand/2/bin version) to tune six hyperparameters of the network. We tuned the following hyperparameters: Threshold voltage(v_{thresh}) and refractory time (t_{ref}) of LIF neurons, which governs the number of spikes generated by the network; positive and negative synaptic weight modification factors of STDP (A_+ and A_-), that governs the degree of each synaptic weight update; positive and negative drifts (D_+ and D_-) of deSNN algorithm, that affects the classifier vector representation of a given sample (The tuned hyperparameter values and search ranges are given in Table 5.1). For this optimisation, we defined an objective function that considered both 60:40 split testing and five-fold cross-validation (only on 60% of training data used) accuracies. We initiated the network ten times and calculated the mean of average cross-validation and 60:40 split testing accuracies, which were set as the objective function. Therefore, the objective function considered both training data fit and generalisation capability. The maximisation of the objective function or testing of 5000 combinations was set as the termination criteria for the optimisation process. Once we obtained the tuned parameters, the network was tested (same split ratio) for 30 cycles with pseudo-random weight initiations and sample selections to demonstrate the performance statistically.

5.6.5 NeuCube architecture

NeuCube is a specialised SNN architecture for spatiotemporal data processing tasks. It consists of a spike encoder, 3D SNN reservoir, classifier and gene regulatory network(optional) modules. This architecture is also suitable for network dynamics visualisation, particularly due to its capability to map EEG into a network structure according to the Talairach brain template (Talairach & Tournoux, 1988) (see Figure 5.3). Here the 3D SNN reservoir consists of LIF neurons with connections established according to small-world connectivity (Watts & Strogatz, 1998) (i.e., inclusive of recurrent connections) and trained using STDP. In our experiments, we intend to investigate the differences in network dynamics under stressed and relaxed data separately. Therefore, we trained the reservoir with specific samples to extract information on input neuron clusters, connection formation and firing correlations between input neurons.

5.7 Results and Discussion

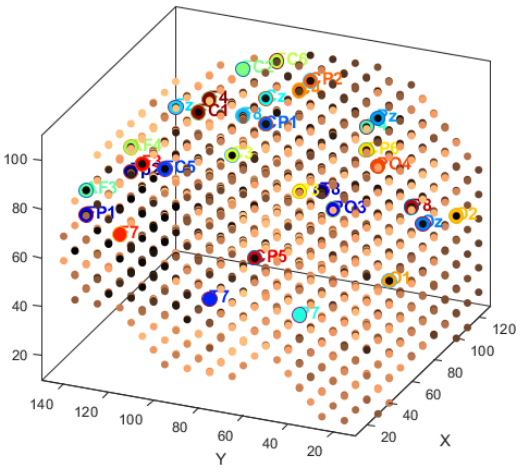


Figure 5.3: NeuCube spiking neural network architecture (N. K. Kasabov, 2014) mapped using Talairach brain template (Talairach & Tournoux, 1988) for network behaviour analysis (View from the left side of the 3D reservoir)

The model proposed in this study yielded 79.8% mean accuracy with 2.7 variance during 30 randomly initiated cycles of five-fold cross-validation for training data (see Figure 5.5). Moreover, the model produced 83% mean accuracy and 5.1 variance during 30 randomly initiated cycles of 60:40 split training testing conditions (see Figure 5.4). The maximum accuracy was recorded as 88%, and the minimum at 69%. The model that produced the highest accuracy (i.e., 88%) demonstrated an F1 score of 86.21% and a Matthews correlation coefficient of 0.78 (see Figure 5.9 for confusion matrix). The mean accuracy performance in testing is compared with previous studies in Table 5.2. Importantly, these studies used the same dataset and data labelling with different ML techniques (García-Martínez et al., 2017; Hosseini & Khalilzadeh, 2010; Shon et al., 2018). From the comparison, it is evident that the method introduced in this paper has performed better in terms of mean classification accuracy of 83%. Moreover, the five-fold cross-validation accuracy of 79.8% for the training dataset indicated the non-existence of overfitting conditions. One of the main advantages of our method is the ability to process data without manual feature extraction. The ability of SNNs to learn inherent spatiotemporal features has been discussed previously (K. Roy et al., 2019; Tavanaei et al., 2019). Moreover, Alhagry et al. demonstrated the capability of LSTM in successful temporal feature learning and reported accuracy levels beyond 85% in classifying emotions using raw EEG (2017). These findings highlight the possibility of emotion recognition without manual feature extraction, given the ML technique can exploit spatial and temporal features.

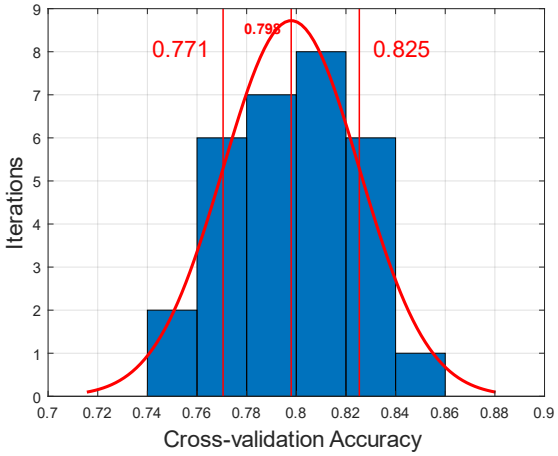


Figure 5.5: Five-fold cross-validation accuracy over 30 iterations

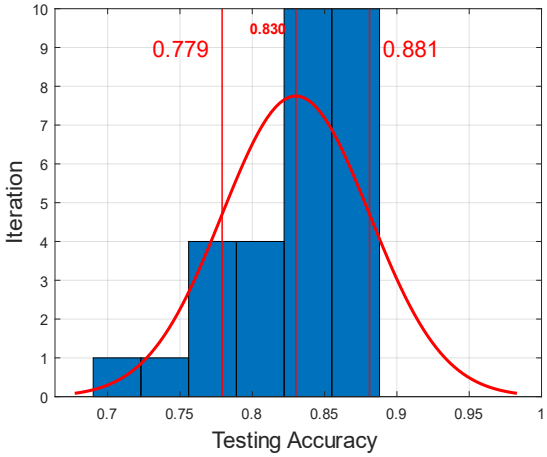
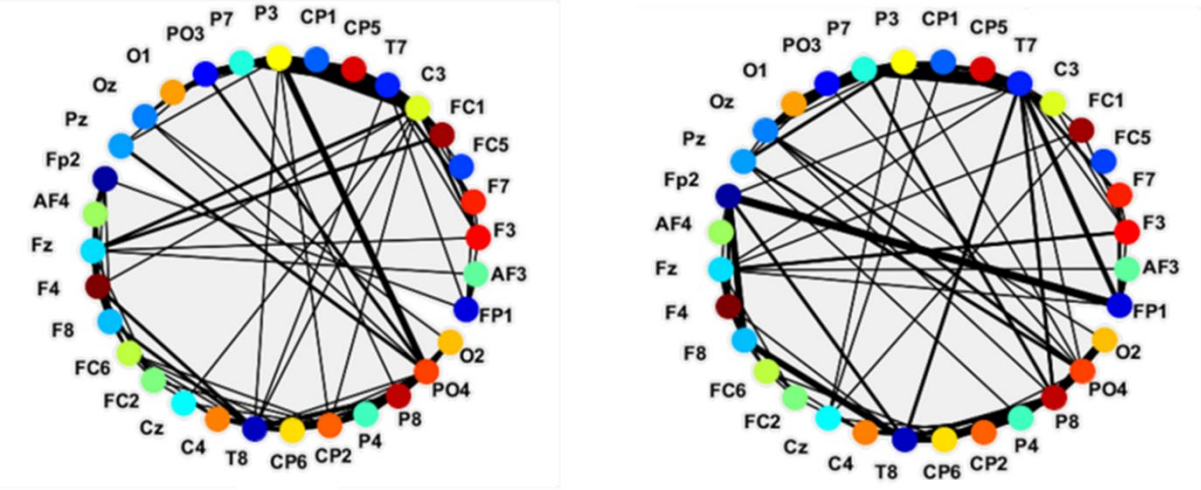


Figure 5.4: Split testing accuracy (60% training, 40% testing) over 30 iterations

We also analysed the network behaviour using stress and relaxed data separately using the NeuCube SNN architecture (N. K. Kasabov, 2014) with the Talairach brain template (Talairach & Tournoux, 1988). In Figure 5.7, brighter and darker neurons represent high and low spiking neurons in the network, respectively. The P7 cluster shows more activation in stressed than relaxed. In contrast, the P8 cluster demonstrates more activation under a relaxed state. In the occipital areas, higher activations are evident under stress. The top view indicates more activation in the left hemispherical neurons under stress. Moreover, the process of evolving connections for the two classes of data is given in Figure 5.8. Blue and red connections indicate excitatory and inhibitory synaptic connections respectively. In this figure, more synaptic connections are observed with stressed data than relaxed. Moreover, the thickness of the lines connecting input neurons in Figure 5.6 indicates the degree of firing correlation. Higher correlations are observed cross hemispherically between P3 and PO4 with stress data and, FP1 and FP2 with relaxed data.

Compared to the traditional ANNs, which are often not interpretable (i.e., black-box approaches), our method provided an opportunity to investigate differences in the network behaviour of stress and relaxed states. The SNN neurons corresponding to the left brain indicated more activation compared to the right brain under stress conditions and vice versa. However, psychological studies have suggested opposing activation patterns in the brain for stress and relaxed states (Coan & Allen, 2004). These anomalies are not fully understood and require further investigation. Higher prefrontal activations with a relaxed state over a stressed state is another observation of our study that has been proposed in neurobiological studies (Arnsten, 2009). Since this SNN used STDP as the primary learning algorithm, which takes the temporal synchrony of the incoming data, the interpretation of activation could be much closer to the actual activation patterns that exist. This network behavioural analysis not only provides



(a) **(b)**

interpretability but also could be used to generate interesting hypotheses.

Figure 5.6: Input neuron firing correlation analysis after training two networks separately with (a)stressed (b)relaxed EEG data samples

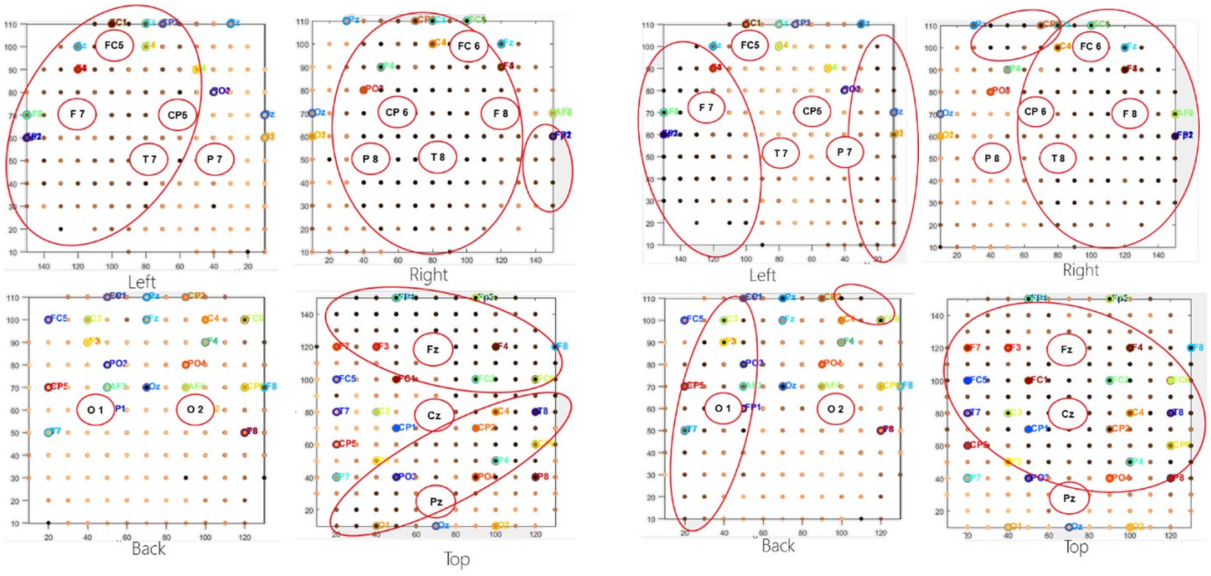


Figure 5.7: Input neuron cluster analysis using 2D views from left, right, back and top of the NeuCube reservoir after training separately using class-specific data (a) Stressed (b) Relaxed. Brighter neurons represent higher activations and darker neurons lower activation. Darker neurons are contoured in red for better visibility

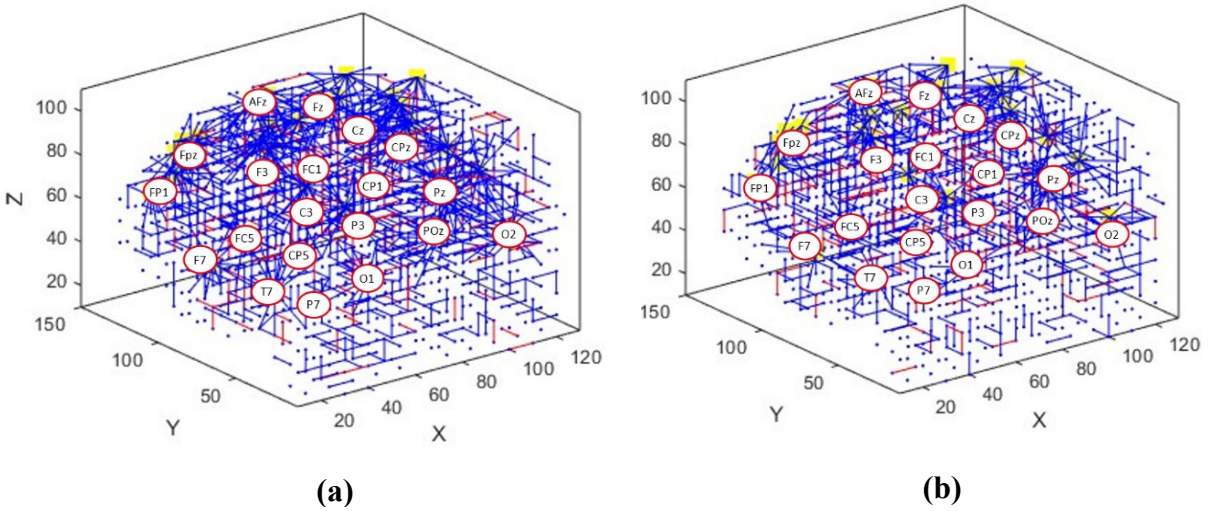


Figure 5.8: Synaptic connection formation according to small world algorithm(Watts & Strogatz, 1998) after training the network with class-specific incoming spike trains (a) Stressed (b) Relaxed

		Actual	
		Stressed	*Relaxed*
Predicted	*Stressed*	25	7
	Relaxed	1	35

Figure 5.9: Confusion matrix of the best model that produced 88% accuracy for 60:40 split testing

5.8 Conclusion

The application of ML algorithms in the psychology and neuroscience domain has gained much popularity in the recent past. Among them, studies on emotional stress recognition have particularly progressed due to their potential to enable better mental health. In this study, we used an immerging type of artificial neural network, namely, the Spiking Neural Network, for emotional stress recognition and achieved state-of-the-art classification accuracy with the use of a publicly available EEG dataset. In this work, we have also demonstrated the aptness of SNNs for exploratory data analysis using brain data. In terms of limitations, we have used a balanced set of data which limits us from commenting on simulations with unbalanced data. As a part of our future work, we foresee the possibility of developing SNN models for individuals with online learning capability for mental stress recognition using brain data and other physiological data such as galvanic skin response (GSR) and heart rate variability (HRV).

Table 5.1

Network hyperparameter values

Algorithm	Hyperparameter	Value	Search range
AER Encoder	f	0.5	-
LIF Neuron	v_{thresh} [a]	0.048	[0.01 - 0.5]
	t_{ref} [a]	5	[2 - 10]
	v_{rest}	0	-
	R	1	-
	C	10	-
STDP	A_+ [a]	0.030317	[0.001 - 0.05]
	A_- [a]	0.000958	[0.001 - 0.05]
	τ_{pos}	10	-
	τ_{neg}	10	-
	Weight boundaries	0.1 to -0.1	-
deSNN	D_+	0.025753	[0.001 - 0.05]
	D_-	0.0099	[0.001 - 0.05]
	mod	0.8	-
	α	1	-

[a] Hyperparameters optimised.

Table 5.2

Performance comparison with previous studies

Study	Feature	Classifier	Accuracy
(Hosseini & Khalilzadeh, 2010)	PSD	K-NN	70.1%
	HOC		69.59%
	Statistical		69.59%
(Shon et al., 2018)	GA[a]	K-NN	71.76%
	All features[b]		67.08%
	PCA		65.03%
(García-Martínez et al., 2017)	QSEn, PEn and AAPEn	SVM	81.31%
This study	None	SNN	83%

[a] Genetic algorithm was used to select features from statistical features, PSD, HOC, Hjorth parameter and frontal asymmetry alpha

[b] All features mentioned in 'a' were used

Chapter 6 Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling

This chapter is comprised of a peer-reviewed journal article submitted to and published by *IEEE Access.*

6.1 Reference:

Weerasinghe, M. M. A., Espinosa-Ramos, J. I., Wang, G. Y., & Parry, D. (2021). Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling. *IEEE Access*, *9*, 117338–117348. https://doi.org/10.1109/ACCESS.2021.3099492

6.2 Author contribution:

MMAW: 82%, JIER :8%, GW: 5%, DP: 5%

M.W. designed the algorithm, ran the experiments, analysed the data, and wrote the manuscript. J.R supported the development of algorithms and scrutinised the methodology. G.W. and D.P. supervised the research. All authors reviewed the manuscript and provided feedback.

6.3 Preamble:

At the point that the experiments in this chapter were designed, the need for structural adaptation methods to operate in tandem with other hyperparameters (LIF thresholds, STDP learning factors and drift parameters) was evident. The results of prior experiments showed an 83% average accuracy for stress classification using tuned hyperparameters and a network of 130 neurons (Chapter 5). In contrast, the SNN with 154 hidden layer neurons produced a 77% average accuracy without hyperparameter optimisation (Section 4.5). Moreover, an SNN with 130 neurons without tuned hyperparameters produced only 75% average accuracy. These results suggest that using tuned hyperparameter values would produce better accuracy and enable performance with fewer hidden layer neurons.

Although the combined operation of an adaptive hidden layer and hyperparameters could bring better pattern recognition capabilities to an SNN, the type of network spiking behaviour that would bring

about a better performance is unknown[15]. The hidden layer of the SNN framework discussed in this thesis encodes incoming data into spike patterns using unsupervised learning and feeds it to the classifier. Therefore, the size of the neuron population in the hidden layer and hyperparameters involved with neuron excitability and STDP regulation influence the final classification accuracy. Understanding at a basic spiking level why a certain number of neurons and hyperparameter values help classification accuracy is critical to implementing data-driven structural adaptation for SNNs without relying on exhaustive tuning methods.

To address this lack of understanding, the journal article presented in this chapter introduces a Differential Evolution based structural plasticity (SP)[16] incorporation method. Here, the SNN structure adapts automatically and iteratively with six other hyperparameters (see Table 6.1), improving accuracy performance at each iteration. The optimised SNN is compared with overgrown and undergrown structures where the hidden layer neuron number is kept fixed, but the hyperparameters are tuned. This work is also expected to confirm whether structural adaptation is needed for better performance or whether hyperparameter optimisation would suffice. The spiking behaviour of the network and the values taken by hyperparameters are also analysed comparatively. The findings of this analysis resulted in spiking indicators that can predict optimal structures that can produce better performances.

6.4 Abstract

Structural Plasticity (SP) in the brain is a process that allows structural neuronal changes in response to learning. Spiking Neural Networks (SNN) are an emerging form of artificial neural networks that use brain-inspired techniques to learn. However, the application of SP in SNNs, its impact on overall learning, and network behaviour is rarely explored. In the present study, we use an SNN with a single hidden layer to apply SP in classifying Electroencephalography (EEG) signals of two publicly available datasets. We considered classification accuracy as the learning capability and applied metaheuristics to derive the optimised number of neurons for the hidden layer along with other hyperparameters of the network. The optimised structure was then compared with overgrown and undergrown structures to compare the accuracy, stability, and behaviour of the network properties. Networks with SP yielded ~94% and ~92% accuracies in classifying wrist positions and mental states (stressed vs relaxed), respectively. The same SNN developed for mental stress classification produced ~77% and ~73% accuracies in classifying arousal and valence states. Moreover, the networks with SP demonstrated

[15] Particularly when unsupervised STDP is in operation.
[16] In terms of neuroscience, structural plasticity is the structural adaptation of neuron circuits according to stimuli (Aimone, 2016). In this thesis, the terms represent automated changes in the structure due to influences of the input stimuli.

superior performance stability during iterative random initiations. Interestingly, these networks had a smaller number of inactive neurons and a preference for lowered neuron firing thresholds. This research highlights the importance of systematically selecting the hidden layer neurons over arbitrary settings, particularly for SNNs using Spike Time Dependent Plasticity learning and provides potential findings that may lead to the development of SP learning algorithms for SNNs.

6.5 Introduction

Spiking Neural Networks (SNNs), referred to as the third generation of neural networks (Maass, 1997), are capable of accommodating pattern recognition and function approximations with greater computational efficiency (Tavanaei et al., 2019). This can be credited to the approach of adopting biologically inspired information processing techniques in SNNs. Although this approach does not guarantee better accuracy in pattern recognition tasks, it is more likely to give detailed insight into brain-like computing (Pfeiffer & Pfeil, 2018), such as event-based spatiotemporal data processing.

However, the advantages of SNNs cannot be fully utilised in Machine Learning (ML) tasks due to the lack of robust and well-justified learning techniques (Grüning & Bohte, 2014; Pfeiffer & Pfeil, 2018; Tavanaei et al., 2019), which inhibits accuracy. Therefore, it is a common practice to make SNNs larger and deeper (Diehl & Cook, 2015; J. H. Lee et al., 2016). Additionally, SNNs operate with a large number of hyperparameters, which can make network implementation and optimisation more computationally expensive. To resolve these issues, it is important to better understand the unified function of neurons, synapses and hyperparameters in accurately recognising patterns.

The remarkable pattern recognition capability of the mammalian neocortex is achieved with a power consumption of 10 to 20 Watts (Javed et al., 2010). One key attribute that accommodates this phenomenon is structural plasticity (SP). This is the self-regulation capability of neuron circuits which was initially investigated by (Peter R., 1979). Moreover, the neuroscience literature (Aimone, 2016; Iglesias & Villa, 2007; Spiess et al., 2016; Yuan et al., 2019)further highlights the positive impact of SP on learning. Though brain-inspired learning concepts such as spike time dependent plasticity (STDP) (Bi & Poo, 1998; Song et al., 2000) and its variants are being applied for training SNNs, how SP works in an STDP learning environment is rarely investigated. Studies exploring the interplay between SP and computational properties of neurons and synapses, such as firing thresholds, are even rarer.

In this paper, we explore the impact of systematically varying the number of neurons in the hidden layer (denoted as η) based on prior learning on ML performance. Furthermore, we investigate the role of hyperparameters (i.e., intrinsic properties of neurons and synapses) under this process to provide better insights to further expand the knowledge in the area. For this purpose, we use metaheuristics with SNNs. To the best of our knowledge, this is the first time that the impact of η on hyperparameters and network performance has been analysed quantitatively and qualitatively. Our research used a 3 layered SNN architecture and two publicly available Electroencephalographic (EEG) datasets (Koelstra et al., 2012; Taylor et al., 2014).

This paper is organised as follows. Section 6.6 presents a summary of research on the application of SP in SNNs. Section 6.7 introduces the two EEG datasets, annotation and encoding procedures. Section 6.8 describes the SNN setup and learning procedures involved. In Section 6.9, we introduce the structural optimisation techniques and experimental framework. Section 6.10 presents results and observations which are discussed and concluded in Sections 6.11 and 6.12, respectively.

6.6 Structural Plasticity (SP) in SNNs

In this section, we intend to summarise SP methods found in SNN literature and highlight the research gap that we intend to contribute towards. Here, we commonly refer to all structural adaptation methods in SNN literature as SP. This includes both growth and pruning of neurons and/or synapses. When we consider SP in SNNs, all research in the area can be divided into two main categories. Use of SP as a learning mechanism introduced by (Dora et al., 2016; S. Roy & Basu, 2017; J. Wang et al., 2014; Wysoski et al., 2008) or as a method to increase efficiency (Rathi et al., 2019; Shi et al., 2019).

In previous research (Wysoski et al., 2008), a four-layered SNN was introduced with 2D maps of integrate and fire neurons specifically designed for visual recognition tasks. The first two layers filter incoming samples according to contrast and orientation. The third layer of this SNN learns using the rank order (RO) rule (S. Thorpe & Gautrais, 1998). Each time a sample is presented to the network, a new neuronal map is created in the third layer. The similarity of the newly created map with already existing maps is calculated by applying inverse Euclidean Distance measures between weight matrices. If the similarity exceeds a predefined similarity threshold value, maps are merged and, if not, maintained separately. For each of the maps in the third layer, a single neuron is maintained at the fourth layer. The synapse in this layer responds only to excitatory signals where weight is increased by +1 for each spike arrival. During inferencing, the class-specific neuron at the fourth layer spikes.

Another SP algorithm is introduced by (J. Wang et al., 2014), a three-layered SNN with population coding introduced in the first layer with multiple delays. The hidden layer consists of RBF neurons where the sample representative capability of a given input is determined by the time taken to produce a spike. Therefore, the neuron that spikes first in the hidden layer is considered the winner neuron. The time taken by the winner to fire, is compared with a predefined threshold to decide the new neuron addition (i.e., existing neurons is not representative enough). Similarly, if two winner neurons fire with a certain time gap which is below a predefined pruning threshold, one neuron is removed (i.e., two neurons demonstrate similar representative capability). In the case of neuron addition, scaled weights are assigned to the newly added neuron, making it fire first for the current sample. STDP and anti-STDP rules are used to update the weights between hidden and output layers. Rate coding is used at the output layer to determine the class of a given input.

Dora et al. introduced a two-layered SNN with SP applied at the output layer (2016). The input layer of the network is used for population coding, and the output neurons are leaky integrate and fire (LIF) neurons. The weights between the layers are updated using the modified RO rule. The class of a given input signal is determined by the neuron that produces a spike earliest at the output layer. When a new sample is propagated, if the time stamps of the first spikes produced by output neurons exceed a predefined threshold, a neuron is added, and if it is below the threshold, the sample is skipped. The time gap between the predefined threshold and the actual spike time for the current sample determines whether a new neuron should be added, the sample to be skipped, or weights should be updated.

An online learning mechanism based on adaptable connectivity is introduced in (S. Roy & Basu, 2017). In this study, a winner-takes-all network is implemented with a single layer of LIF neurons with multiple synapses. A separate connection matrix is maintained, which keeps track of the connections to synapses and input streams. The efficacy of a certain connection to make a neuron spike is being calculated using STDP inspired learning algorithm. This value of spike efficacy is then compared with a dynamic threshold value to allow the elimination of connections after each training epoch. Therefore, apart from weight leaning, synaptic rewiring takes place.

Pruning is another form of SP applied mainly to increase efficiency, often at the cost of learning capability. In (Rathi et al., 2019), researchers introduce a synaptic pruning method based on STDP weights. There, a pruning threshold is defined, and STDP weights below the threshold values are pruned. In the same research, the applicability of weight quantisation is introduced, which restricts all the values in the trained weight matrix to a predefined scale. Another form of synapse elimination is Soft-Pruning (Shi et al., 2019), where synapses selected to prune are set to their lowest value instead of

completely removing during training epochs. The total elimination of synapses is made once training is completed. Both methods have proven to have the capability of reducing the network parameters drastically while maintaining accuracy performance up to a certain extent.

As discussed (Rathi et al., 2019; Shi et al., 2019) are methods where SP can be implemented to increase resource efficiency, which is particularly important in hardware implementations of SNNs. In contrast, (Dora et al., 2016; S. Roy & Basu, 2017; J. Wang et al., 2014; Wysoski et al., 2008) introduced SP as a form of learning method where neurons are added and/or removed with predefined thresholds to obtain desired spiking patterns. However, these methods are not intended to explore the impact of η under STDP learning or the impact of the same on overall ML performance and intrinsic properties of the network.

6.7 Stress recognition using EEG

SNNs process and communicate information with sparse and asynchronous binary signals called 'spikes' (Grüning & Bohte, 2014). This method of operation, combined with unsupervised learning methods such as STDP, makes SNNs an ideal solution for exploring the dynamics of spatiotemporal data (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019). EEG is one such data type with properties of autocorrelation and heterogeneity built into the temporal signal. Recent studies (Chikara & Ko, 2019; Ko et al., 2020) have used feature extraction methods with traditional ML techniques (non-spiking) in exploring these properties. In this study, we have used an SNN architecture in classification tasks without such manual feature extraction.

6.7.1 EEG Datasets

1) Dataset 1- wrist Flexion dataset

We obtained this dataset from a study previously conducted to test the feasibility of an SNN architecture in detecting motor execution and motor intention (Taylor et al., 2014). EEG was collected from 3 healthy participants when performing three different wrist positions, namely flexion, extension, or rest. EEG was recorded using 14 channels Emotive Neuroheadset with international 10-20 locations, and each recording lasted for 20 seconds. Recordings were obtained under an eyes-closed state, minimising artefacts due to eye blinking and no additional data cleaning techniques were utilised except for the data encoding. The recordings were sampled at 128 Hz without additional data preparation steps. (Dataset is available at https://github.com/KEDRI-AUT/neucube-cloud-sample-file/blob/master/wrist_movement_eeg.zip).

2) Dataset 2- emotional stress dataset

We used an extraction of the DEAP dataset (Koelstra et al., 2012) to detect emotional stress. This dataset consisted of EEG recordings of 32 healthy participants collected while watching 1-minute video clips, consequently annotated by each individual for valence and arousal using a self-assessment manikin (Bradley & Lang, 1994). Each recording was conducted using 32 channels at 512 Hz. Our experiments used 32-channel data downsampled to 128Hz, filtered with a 4Hz to 45Hz bandpass filter, and EOG removed. (The main pre-processed dataset is publicly available at https://www.eecs.qmul.ac.uk/mmv/datasets/deap/download.html). We then normalised each sample to bring all values to a common scale, as per Figure 6.1. This process was conducted on all samples prior to data encoding.

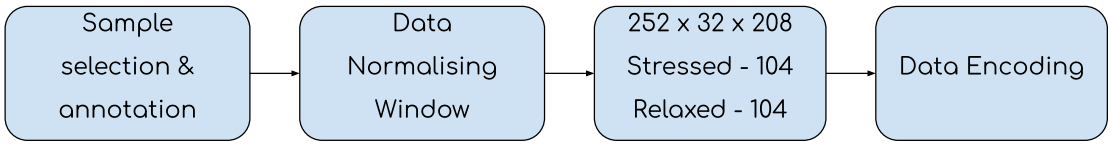


Figure 6.1:Data preparation process used for the second dataset

6.7.2 Data Annotation

Firstly, we converted all samples of both datasets into comma-separated values (CSV) file format. Therefore, each sample file represented the format of an m × n matrix with rows representing time points and columns representing EEG channels (i.e., Features/Attributes).

1) Dataset 1- wrist Flexion dataset

A sample of this dataset is a 128 × 14 (i.e., time steps × channels) matrix. The dataset comprised 60 samples in total, with 20 samples corresponding to each wrist position. We annotated wrist flexion as class one, extension as class two, and resting state as class three.

2) Dataset 2- emotional stress dataset

The extraction from the main DEAP dataset used the following equations:

$$Stressed = (Arousal > 5) \cap (Valence < 3) \quad \textbf{(16)}$$

$$Relaxed = (Arousal < 4) \cap (4 < Valence < 6) \quad \textbf{(17)}$$

The equations were obtained from (Savran & Arman, 2006). We assigned stressed samples as class one and relaxed as class two. A sample of the selected dataset is in the form of 252 × 32 (i.e., time steps × channels) matrix. A total of 208 samples were obtained using the above equations, with 104 samples for each class. This same data extraction method with DEAP dataset was carried out by (Bastos-Filho et al., 2012; García-Martínez et al., 2017) and (Shon et al., 2018), which enables performance comparison. However, it is important to note that it differs from the common classification approach used with all samples in the DEAP dataset that uses four classes for classification in studies such as (Delbruck & Lichtsteiner, 2007).

6.7.3 Data Encoding

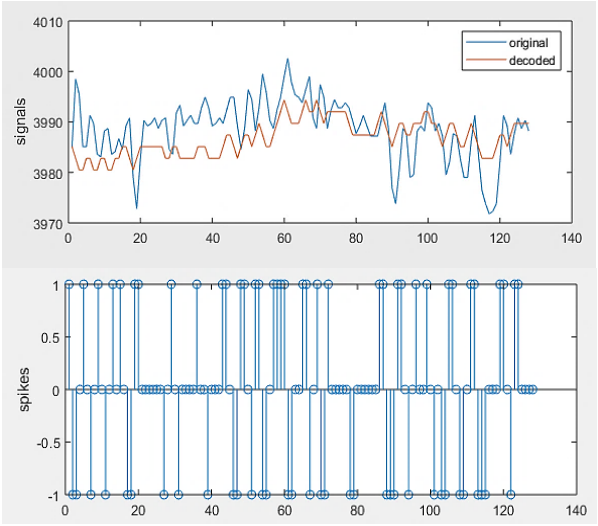


Figure 6.2: Spike representation of FP1 channel extracted from a stressed class sample. The top plot illustrates original signal (Blue) and reconstruction of the same using spikes generated (Red)

Before feeding the data into the SNN, a spike conversion takes place. For this, we used the Address Event Representation (AER) (Delbruck & Lichtsteiner, 2007) for both datasets considering the representation and noise filtering capability. Equation (18) provides the threshold calculation, where $M(dif)$ and $Std(dif)$ denotes the median and standard deviation of temporal difference signal dif. A dif of a particular EEG channel is calculated by subtracting amplitude at time t by the amplitude at $t-1$. The threshold factor f is set by the user (In our experiments this was set to 0.5). Once Tr is calculated according to (18), dif amplitudes at each time point are compared. If the dif amplitude exceeds Tr at time t, an excitatory spike is emitted and if the dif amplitude drops below Tr, an inhibitory spike is emitted.

$$Tr = M(dif) + f * Std(dif) \quad \textbf{(18)}$$

As shown in Figure 6.2, where peaks of the original signal are represented with a volley of spikes, this encoding algorithm gives prominence to amplitude fluctuations, enabling the preservation of salient events in the EEG signal. This method can filter out minute fluctuations caused by signal noise.

6.8 SNN Architecture

In our experiments, we used a 3-layered SNN with leaky integrate and fire (LIF) neurons, developed based on the JNeuCube framework, publicly available at https://github.com/Auckland-University-of-Technology/NeuCube-java.

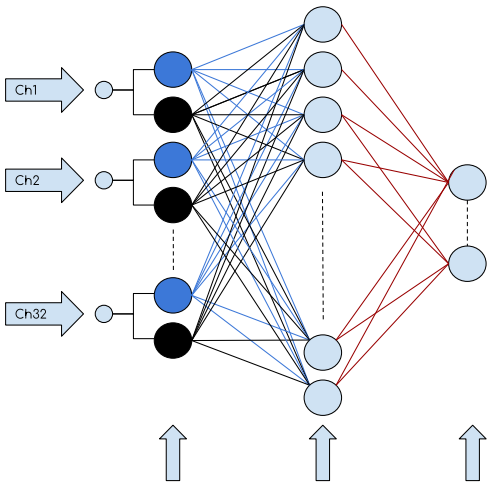


Figure 6.3: SNN architecture used for classification task using dataset 2 accommodating 32 inputs

The input layer consisted of pairs of nodes capable of accommodating both excitatory and inhibitory spikes. The number of pairs depended on the number of EEG channels. (i.e., for Dataset 1, 14 pairs and dataset 2, 32 pairs). We fully connected the input to the hidden layer with pseudo-random weight initiations following the gaussian distribution and an unsupervised learning strategy based on the spike time dependency (STDP) algorithm (Bi & Poo, 1998; Song et al., 2000) to update weights. The hidden to-output layer also followed fully connected criteria. The weight update strategy was inspired by deSNN algorithm (N. Kasabov et al., 2013).

6.8.1 Leaky Integrate and Fire neurons

For the neuron model selection, we focused on using a simple biological representation with tractability. Therefore, we used LIF model described in (Naud & Gerstner, 2012). The behaviour of a LIF is modelled using a resistor-capacitor circuit adhering to the following equations,

$$\frac{d}{dt}(v) = \left[RI_{(t)} - (v_t - v_{rest}) \right] / \tau_m \quad \textbf{(19)}$$

where v denotes membrane potential, R resistance, $I_{(t)}$ for instantaneous current, v_{rest} for resting voltage and τ_m the membrane time constant, which is calculated as,

$$\tau_m = RC \quad \textbf{(20)}$$

and when membrane potential v exceeds a defined threshold v_{thresh}, a spike denoted as s^{ti} is generated i.e.,

$$s^{ti} : v_{(ti)} \geq v_{thresh} \quad \textbf{(21)}$$

6.8.2 Spike Time Dependent Plasticity

STDP is a temporally asymmetric form of Hebbian learning which depends on the spiking time between pre and post-synaptic neurons to adjust the weight of the synapse (Song et al., 2000). Therefore, STDP is a local learning rule i.e., it does not consider the information of other synapses for weight update. The significance here is that STDP enables neurons to discriminate temporally distinct inputs and then integrate them to form a meaningful output (Watt & Desai, 2010). Hence by using STDP in a fully connected network, we ought to capture the patterns between spatially distributed yet temporally synchronised spiking activity. One of the drawbacks of using STDP for learning is the runaway synaptic potentiation, i.e. the synapse gets caught up in a potentiation loop that increases weight even if the information coming into the synapse is insignificant (Gerstner et al., 1996). For the experiments discussed in this paper, we have restricted the weight increase by defining boundaries (+0.1 and -0.1), and we have introduced inhibitory nodes in the input layer to balance the spiking activity.

The formalised mathematical model for STDP can be interpreted using (22) adapted from (Song et al., 2000). Depending on the coincidence between spiking activity within the learning window (i.e. the time window considered for weight adjustments between the pre and the post-synaptic spike), the long-term potentiation (LTP) i.e. synaptic strengthening by weight increasing or long-term depreciation (LTD) i.e. synaptic weakening by weight decreasing takes place according to (23) and (24).

$$\Delta w_{ij} = \sum_{k}^{y} \sum_{m}^{z} F(t_i^m - t_j^k) \qquad \textbf{(22)}$$

$$F_{(\Delta t)} = A_+ \, e^{(-\Delta t/\tau pos)} \qquad \Delta t > 0 \qquad \textbf{(23)}$$

$$F_{(\Delta t)} = -A_- \, e^{(\Delta t/\tau neg)} \qquad \Delta t < 0 \qquad \textbf{(24)}$$

As per (22), the weight adjustment Δw_{ij} is an accumulation of weight fluctuations calculated using the function $F_{(\Delta t)}$ over pre-synaptic spikes occurring from k to y and post-synaptic spikes occurring from m to z, where t_i^m represents post-synaptic spiking times and t_j^k represents pre-synaptic spiking times. In (23), Δt is the time gap between the two spikes and A_+ represents the positive modification

factor and *τpos* denotes the learning window for positive weight modifications. Vice versa, equation (24) denotes the negative weight modification with the relevant modification factor A_-, and learning window, *τneg*. During our experiments, Δt was held equal and constant at 10 ms for excitation and inhibition, according to previous experimentations (Song et al., 2000). It is also understood that inequality of the same would create biases causing misrepresentations. Moreover, A_+ and A_- were selected as parameters to be tuned with a starting value of 0.001. The selection of these values depended on the simulations conducted with the encoded data.

6.8.3 Classifier learning

We used a version of the Dynamic Evolving Spiking Neural Network (deSNN) (N. Kasabov et al., 2013) algorithm to create and adapt weights from the hidden to the output layer. This segment of the network considers the global activity of the hidden layer for learning and inferencing. Therefore, with STDP, it is a combination of local and global learning, a contrast to methods using only RO learning (i.e. global learning only) (Dora et al., 2016; Wysoski et al., 2008) .

The output layer (i.e., the sample classifying layer) is a setup that increases its number of neurons with each passing sample. Therefore, during the training cycle, each sample will be represented with a neuron at the output. This neuron is fully connected with the hidden layer. Initial weights of the hidden to output layer were established according to the rank order rule (S. Thorpe & Gautrais, 1998) as per **(26)**,

$$w_{ho} = \alpha . mod^{order(h,o)} \quad \textbf{(25)}$$

Where w_{ho} represents synaptic weight between hidden and output neurons. α is the learning parameter, the constant factor which decides the magnitude of the weight. For these experiments, we have set this to 1 based on previous simulation experience. The modulation factor *mod* takes the values between 0 and 1. Each synapse coming from the hidden layer is set with an initial weight based on the spike arrival order *order(h,o)*. Here, *mod* is set to 0.8, and the corresponding neuron synapse will get a weight of 1 since the *order(h,o)* would be set to 0. The spiking proceeded will receive weights from *order(h,o)* increased 0 onwards.

Once the initial weight of each synapse is set, using two parameters, namely *positive drift* and *negative drift*, weights are updated according to the spike arrival at each time step. The values of these parameters need to be tuned. If a spike arrives at a given time step, the new weight becomes w_{ho}+ *positive drift*, else w_{ho}- *negative drift*.

At the beginning of testing, the output layer consists of *N* number of neurons, each representing a training sample with the label being known. During testing, a sample is propagated, a new neuron is created, and weights evolve at each time step. Thereafter, the weight vector of that sample is used to calculate the Euclidean distance. According to the Euclidean distance, the model selects the closest training sample neuron and predicts by assigning the same label to the testing sample.

Considering the uniqueness of each participant's EEG data, we used this technique for the classification layer without neuron clustering in contrast to methods in (Dora et al., 2016; N. Kasabov et al., 2013; Wysoski et al., 2008).

6.9 Network Optimisation

When it comes to SP, it is not justifiable to only change η and evaluate the performance of the network since the six hyperparameters mentioned in Table 6.1 directly influence spiking activity and classification accuracy. Therefore, each network with a certain η requires appropriate hyperparameter values to be chosen. This section describes the process of selecting the hyperparameters.

Table 6.1

Hyperparameters of the experiment

Step	Hyperparameter	Range Tuned or Held Constant
AER Encoding	Spike Threshold	0.5
LIF	Reset Voltage	0
	Resistance	1
	Capacitance	10
	Voltage Threshold*	[0.01 - 0.5]
	Refractory period*	[2 – 10]
Local Learning	STDP Positive Modification Time Window	10 ms
	STDP Negative Modification Time Window	10 ms
	Random weight initiation boundaries	[-0.1, 0.1]
	Maximum weight bounds	[-0.1, 0.1]
	Positive Synaptic Modification*	[0.001 – 0.05]
	Negative Synaptic Modification*	[0.001 – 0.05]
Global Learning	RO learning parameter	1
	RO modulation factor	0.8
	Positive Drift*	[0.001 – 0.05]
	Negative Drift*	[0.001 – 0.05]

Note. Describes all hyperparameters used in the SNN algorithm. The hyperparameters used for tuning are noted with *, with search range.

6.9.1 Differential Evolution

We used Differential Evolution (DE) (i.e., DE/rand/2/bin version) as the optimisation algorithm to search for the most suitable parameters/hyperparameters that would produce the highest fitness. Amongst the advantages of DE, the capability to handle non-differentiable cost functions with minimum control variables, computational efficiency achieved through the independent generation of populations using stochastic perturbation and good convergence properties that have been experimentally proven were most important (Storn & Price, 1997).

In DE, the greedy criterion is used to determine the acceptance of a given vector solution, and an annealing process is introduced at trial vector generation to counter the search process getting trapped in a local optimum. A selected target vector solution performance is tested with a trial vector solution

developed based on a mutation and crossover. The two variables of the algorithm are the weight factor used to amplify the differential variation in mutation and the crossover constant that determines the probability of the component representation from the mutant vector. In these experiments, we set the weight factor at 0.1 and constant for crossover probability at 0.8 according to the heuristic findings of (Storn & Price, 1997).

6.9.2 Experiment Framework

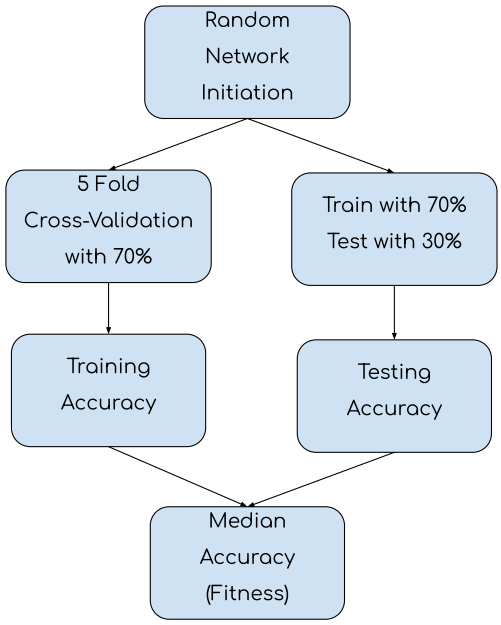


Figure 6.4: Fitness function of a single network initiation

A single trial solution is tested on 10 pseudo-random network initiations. For each network initiation, data samples were selected randomly based on a 0.7 split, with 70% of samples for training and 30% for testing. Each network initiation was validated for classification accuracy with 5-fold cross-validation and a testing round with unseen data. The fitness function of a given trial solution was obtained as the median of average accuracies of cross-validation and testing. (i.e. (average cross-validation accuracy over 10 initiations + average testing accuracy over 10 initiations)/2). Termination criteria were set to 100 iterations or maximisation of the fitness function. Each iteration consisted of 50 agent solutions. Enabling validation of a maximum of 5000 candidate solutions.

1) EXPERIMENT 1 – OPTIMISING HIDDEN LAYER NEURON COUNT AND HYPERPARAMETERS

In this experiment, we optimised η and the six hyperparameters mentioned in Table 6.1. We selected 30 to 200 as the search space for η. This range was selected following random performance testing conducted. We conducted the same experimental procedures to both dataset 1 and dataset 2. Figure 6.5 shows the sample candidate solution used for experiment 1. After completing experiment 1, we obtained structurally optimised networks with corresponding hyperparameter values that produced the highest fitness. This method enables the application of SP based on the network performance, where each iteration generates information on η and other network properties that would enable better performance. This information is used as feedback in regenerating candidate solutions.

2) EXPERIMENT 2 – OPTIMISING COMPARATIVE NETWORKS

If we consider the structurally optimised network to have *p* number of neurons in the hidden layer, we created 2 additional networks, with 1.5 times *p* and 0.5 times of *p* (i.e., A network with 50% more and 50% fewer hidden neurons). These two networks (referred to as overgrown and undergrown) were separately optimised using DE to find the best values for the 6 hyperparameters mentioned in Table 6.1. The search spaces and the number of candidate solutions remained the same. Therefore, we concluded Experiment 2, with three optimised networks, for each dataset. (Optimised parameter values for each of the network and dataset are appended as Table 6.7)

During the evaluation stage, we reinitiated a selected network 30 times with optimised parameters and pseudo-random weights that followed the Gaussian distribution. During each initiation, data was split randomly, with 70% for training and 30% for testing. This allowed a statistical evaluation of the performance.

LIF parameters		STDP weight factors		deSNN weight factors		Structural Plasticity
Threshold	Refractory time	Positive modification	Negative modification	Positive modification	Negative modification	Neuron count

Figure 6.5: Representation of a sample trial solution vector with 6 hyperparameters (given under headings highlighted in blue) and hidden layer neuron count (highlighted in red) which was only used in Exp. 1

6.10 Results

6.10.1 Testing Network Performances

1) DATASET 1

As shown in Figure 6.6, the SNN with 111 hidden neurons produced ~94% average classification accuracy for the 3-class classification task. The standard deviation of the network performance was recorded at 0.048.

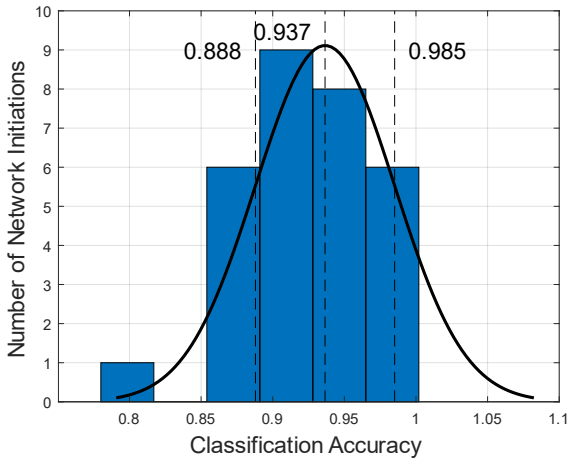


Figure 6.6: Statistical analysis of the performance for dataset 1 across 30 random initiations

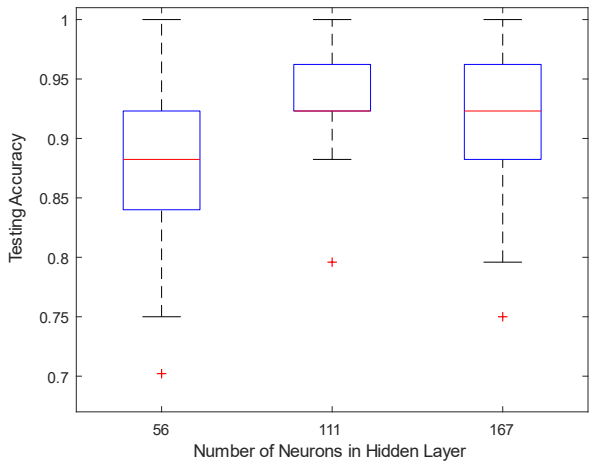


Figure 6.7: Comparative statistical analysis of the generalisation capability of the 3 networks for dataset 1

We compared this network performance with other optimised networks. Figure 6.7 shows the distribution of testing accuracy across the three networks. The network with 111 neurons produced a 7% better average accuracy than 56 neurons. Moreover, 111 neurons produced fewer performance variances comparatively. The average accuracy of 111 neurons and 167 neurons was almost similar.

Each of the networks produced a single outlier performance, with 111 neurons having the highest accuracy among them.

Moreover, we compared the performance of the three networks on Cohen's Kappa and F1s-score using the confusion matrices produced during iterative network testing. A statistical summary of the performance is given in Table 6.2. SNN with 111 neurons performed better than other SNNs in all measures.

Table 6.2

Network performance comparison dataset 1

Network (η)	Test accuracy	Cohen's Kappa	F1-score
56	0.865 ± 0.091	0.678 ± 0.129	0.817 ± 0.114
111	0.937 ± 0.048	0.819 ± 0.099	0.909 ± 0.068
167	0.922 ± 0.058	0.819 ± 0.122	0.889 ± 0.078

Table 6.3 compares the performance of this study with previous studies that used the same dataset for classification. Please note that the previous study used a 50:50 split for training and validation for the experiments with NeuCube architecture without hyperparameter optimisation (Taylor et al., 2014).

Table 6.3

Dataset 1 classification accuracies

Study	Method	Average Classification Accuracy
Taylor(Taylor et al., 2014)	MLP	55%
	SVM	62%
	NeuCube architecture	76%
This Study	3-Layer SNN	93.7 ± 4.8 %

2) DATASET 2

We obtained ~92% average classification accuracy for 2-class classification tasks used to determine between Stressed and Relaxed brain states using the SNN with 130 hidden layer neurons. As shown in Figure 6.8, the Standard deviation of the network performance was recorded at 0.025.

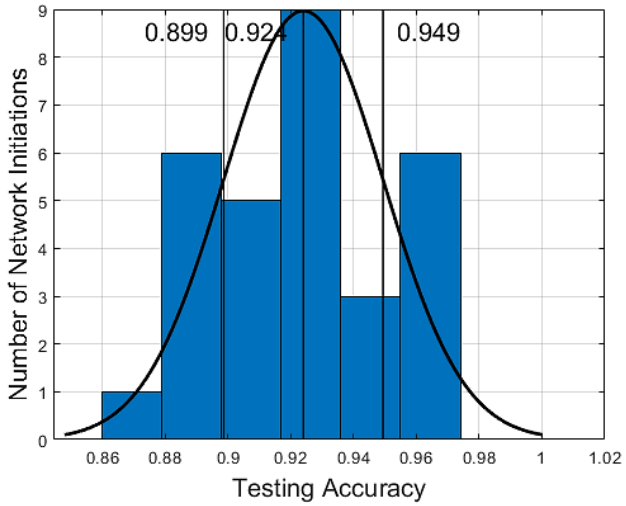


Figure 6.8: Statistical analysis of the performance for dataset 2 across 30 random initiations

Figure 6.9 shows the distribution of testing accuracy across the 3 networks. The network with 130 neurons produced 6% better average accuracy than 65 neurons and 1% better than 195 neurons. Moreover, 130 neurons produced comparatively smaller performance variance.

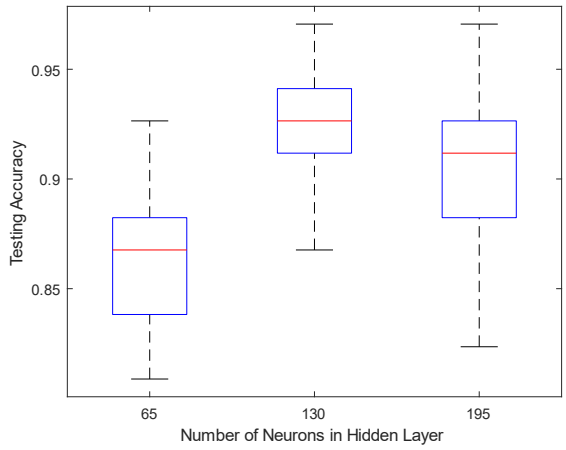


Figure 6.9: Comparative statistical analysis of the generalisation capability of the 3 networks for dataset 2

Table 6.4 presents the Cohen's Kappa and F1-score calculated over testing iterations where the structurally optimised network indicated to have better performance in all measures.

Table 6.4

Network performance comparison dataset 2

Network (η)	Test accuracy	Cohen's Kappa	F1-score
65	0.860 ± 0.033	0.615 ± 0.076	0.8 ± 0.048
130	0.924 ± 0.025	0.823 ± 0.073	0.909 ± 0.039
195	0.910 ± 0.036	0.631 ± 0.086	0.796 ± 0.053

Table 6.5presents a comparison of the studies that used the same data previously for classification tasks with different machine learning techniques and feature extraction methods.

Table 6.5

Dataset 2 classification accuracies

Study	Method	Average Classification Accuracy

Bastos(Bastos-Filho et al., 2012)	K-NN with feature extraction	70.1%
Garcia(García-Martínez et al., 2017)	SVM with feature extraction	81.31%
Shon(Shon et al., 2018)	K-NN with feature extraction	71.76%
This study	3-Layer SNN without feature extraction	92.4 ± 2.5 %

We tested the same SNN with hyperparameters used for stress state classification, for emotion recognition based on valence and arousal dimensions with 10-fold cross-validation to compare with the results summarised in (Chao et al., 2018). Table 6.6 presents this performance comparison.

.

Table 6.6

Dataset 2 classification valence and arousal dimensions

Method	Valence		Arousal	
	Accuracy	**F1 Score**	**Accuracy**	**F1 Score**
GNB(Koelstra et al., 2012)	57.6%	56.3%	62%	58.3%
DBN(D. Wang & Shang, 2013)	60.9%	-	51.20%	-
CNN/RNN (X. Li, Song, et al., 2017)	72.06%	-	74.12%	-
DBN-SVM(X. Li, Song, et al., 2017)	58.40%	-	64.20%	-
DNN(Tripathi et al., 2017)	75.78%	-	73.13%	-
CNN(Tripathi et al., 2017)	81.41%	-	73.36%	-
DBN-GC-based ensemble DNN(Chao et al., 2018)	76.83%	70.15%	75.92%	69.31%
This study	73.10 ± 1.5%	69.5 ± 1.8%	76.86 ± 1.3 %	70.92 ± 1.7%

6.10.2 Inactive Neurons and LIF Threshold

With the differences in performance observed, we analysed the average inactive neurons to the total number of hidden layer neuron ratios for the three networks for each of the dataset. We obtained the average inactive neuron count from the iterations used for statistical performance evaluation in part A. The results of this analysis are summarised in Figure 6.10 for both datasets. We observed a reduction in the ratio for the optimised network structures for both datasets. In other words, in comparison to undergrown and overgrow networks, the network with the optimised number of neurons had a fewer number of inactive neurons compared to the total number of neurons.

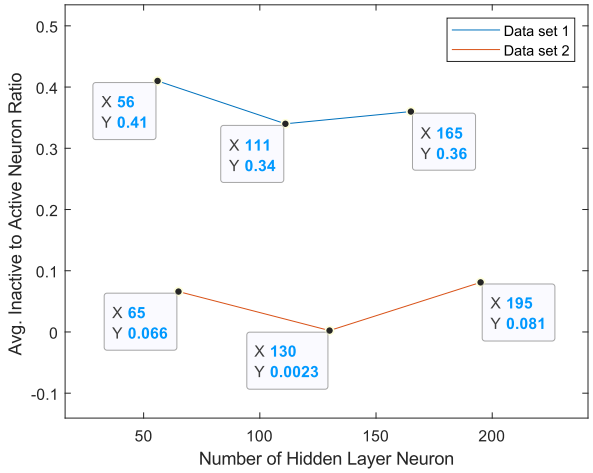


Figure 6.10: Graphical representation comparing inactive neurons to total neurons of different structures tested with dataset 1(Blue) and dataset 2(Red)

In terms of the LIF threshold values, for 56, 111 and 165 η values, we recorded 0.081, 0.064 and 0.067, respectively, for dataset 1. Similarly, for dataset 2, for 65, 130 and 195 η values, LIF threshold was recorded at 0.049, 0.013 and 0.059. This indicates a drop in LIF threshold for the structurally optimised network for both datasets. Figure 6.11 graphically represents this drop for dataset 2.

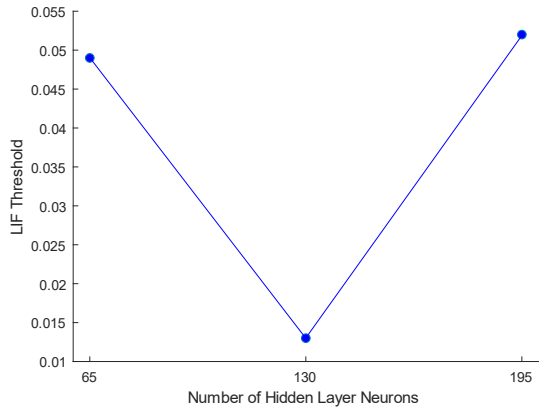


Figure 6.11: LIF Threshold selections for the SNNs with different η values for dataset 2

6.11 Discussion

One of the key findings of this study is the level of performance and stability obtained using SP, reported in Table 6.2 and Table 6.4. For all performance measurements, namely, accuracy, F1-Score and Kappa, the network with SP produced comparatively higher mean and lower standard deviation. Therefore, in addition to having higher accuracy, the SP network performed well in terms of adaptability(i.e., performance stability under changing inputs and/or environments) (Navlakha et al., 2018). This speculation was made with the randomisation we introduced in sample selection, weight initiations and input mapping during network testing. This indicates the importance of considering SP in ML tasks, particularly when using STDP for learning which aligns with neuroscientific findings (Aimone, 2016; Iglesias & Villa, 2007; Spiess et al., 2016; Yuan et al., 2019). Moreover, it is also clear that for SP to provide this performance, the corresponding hyperparameters need to be optimised.

Investigating the better performance of SP revealed lowered LIF thresholds for both datasets in the optimised structure, which results in a higher probability for the neurons to fire. Figure 6.11 presents this finding for dataset 2. However, this spiking was sparser amongst neurons since the relative number of active neurons of the optimised networks was increased (i.e., inactive to the total number of neurons decreased) as per Figure 6.10. This relates to most neurons having balanced firing rates, preventing a selected set of neurons from over-activation, which is desired in SNNs (Diehl & Cook, 2015). However, this desired state is not achieved in the bigger networks presented in this paper. For instance, Figure 6.13 (appended) indicates the firing activity of the trained network for dataset 1, where an overgrown network with 167 neurons produced 5 over-activated neurons. These over-activated neurons can be attributed to excessive exposure to similar patterns in training data and/or runaway synaptic potentiation (Watt & Desai, 2010) caused by STDP. In either case, this will have a negative impact on generalisation capability, as observed. Since the classifier presented here is based on RO and spike counts, over-activation restrains pattern separation. This spiking domination can be potentially minimised by introducing self-regulation methods such as inhibitory neurons (Yuan et al., 2019), lateral inhibition and homeostatic intrinsic plasticity (Diehl & Cook, 2015) or reducing training simulation time (Diehl et al., 2015). As per the smaller networks presented, the level of activation is insufficient to capture the temporal dynamics in the data.

Furthermore, we have obtained state-of-the-art classification accuracies of ~94% and ~92% for datasets one and two, respectively. Moreover, the same SNN developed with SP for mental state recognition yielded 76.86% and 73.1% in classifying arousal and valence, respectively, under 10-fold cross-validation. As per Table 6.6, this result is comparable with contemporary deep learning methods tested using DEAP dataset. The performance of this method can be attributed to the capability of SNNs to

learn from temporal spiking sequences, especially favouring data that are spatiotemporal by nature (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019). Responsiveness of STDP learning in recognising temporal patterns in the data automatically (Guyonneau et al., 2005; N. K. Kasabov, 2014; Rekabdar et al., 2014) may further enhance the said capability. Interestingly, (Chao et al., 2018) also demonstrate the importance of fusing temporal and frequency characteristics of EEG for affect recognition. However, in SNNs, the formulation of an algorithm for salient pattern recognition continues to be challenging. In this study, we have empirically studied the importance of SP in the said formulation process, and the findings correlate with foundational works in SNNs (Maass, 1997), showing the ability to form function approximation with fewer computational units (i.e., neurons).

In terms of the STDP learning algorithm, we could not observe particular patterns of hyperparameters common to both datasets (Figure 6.12 appended). However, larger values for A_+ and $A\_$ seemed to push synaptic values towards the boundary values quicker (i.e., with fewer training samples), thereby missing valuable information in the data during training. It is important to experiment further on STDP weight fluctuations with time and spiking activity of the network during training to understand behaviour patterns of A_+ and $A\_$ that correlates with SP.

6.12 Conclusion and Future Work

The "tuning" of SNN structures is often seen as a matter of expert opinion and may not always be reported in sufficient depth to reproduce existing results. There is a danger that such a "black box" approach may reduce credibility and acceptability. While this work does not solve this problem completely, it offers some potential approaches to move forward in the area.

The findings of this study a) challenge the common practice of having larger η values to produce better accuracies in SNNs and discourage arbitrary setting of the same, but do not suggest that there is one perfect neuron structure for a given data modelling task, b) highlight the importance of SP in achieving higher accuracy and adaptability of the network to increase generalisation capability, and c) present the common observations of reduced inactive neurons and LIF threshold in structurally optimised networks producing sparser spiking.

The methods introduced in this paper apply SP based on evolutionary knowledge of network performance passed on over iterations. Structural adaptation using STDP learning only can make SP more efficient. Extending this research, we foresee the possibility of developing autonomous SP algorithms using the inactive to the total number of neurons ratio as a measure of network learning controlled by η and LIF threshold. This can then be applied as an extension of STDP learning.

6.13 Appendix

Table 6.7

Optimised hyperparameter values

Step	Hyperparameter	Exp. 1	Exp. 2	
Dataset 1	Voltage Threshold	0.064	0.081	0.067
	Refractory period	5.5	5.6	6.1
	STDP Positive Synaptic Modification	0.0027	0.0015	0.0038
	STDP Negative Synaptic Modification	0.0029	0.0018	0.0041
	Positive Drift	0.0372	0.0005	0.0114
	Negative Drift	0.0257	0.0071	0.0129
	Number of neurons	111	56*	167*
Dataset 2	Voltage Threshold	0.013	0.049	0.052
	Refractory period	5	5	5
	STDP Positive Synaptic Modification	0.079	0.0009	0.059
	STDP Negative Synaptic Modification	0.001	0.001	0.001
	Positive Drift	0.040	1.05	0.025
	Negative Drift	0.001	0.001	0.001
	Number of neurons	130	65*	195 *

Note. Presents optimised values of the hyperparameters used in the SNN algorithm. The hyperparameters with * indicate values set manually.

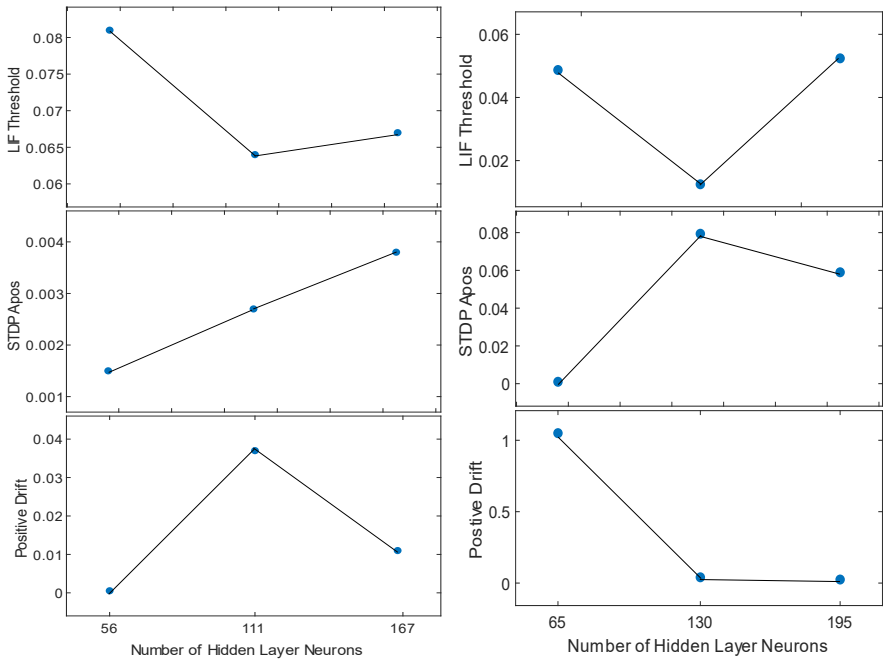


Figure 6.12: Hyperparameter (i.e., Neurons & synaptic properties) preferences over 3 SNN structures compared for dataset 1(Left) and dataset 2 (Right)

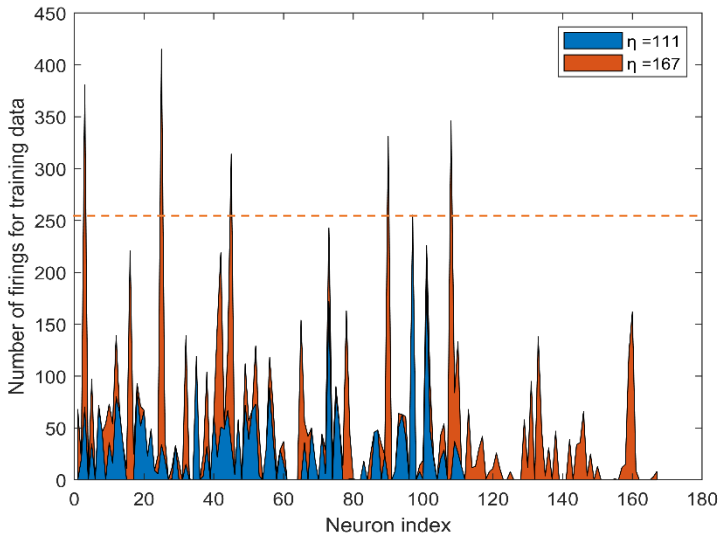


Figure 6.13:Comparison of firing count of each neuron after training with dataset 1 for network with 111 neurons (Blue) and 167 neurons (Orange)

Chapter 7 Brain Plasticity and Learning in SNNs

ML applications of ASNNs[17] often use biologically plausible learning techniques. Among them, plasticity techniques are an important aspect that has the potential to improve the learning and efficiency of ASNNs. The Structural Plasticity (SP) methods discussed in Chapter 6 performed well and provided insights into spiking behaviours that could lead to better performance. However, the biological plausibility of the method of implementation is relatively low. The use of Differential Evolution for hyperparameter selections and final accuracy-based optimisation reduces biological plausibility. This increases the energy and memory requirements due to its exhaustive iterative approach.

The human brain synergises multiple plasticity techniques to achieve remarkable learning capability with limited resources. However, the knowledge of how different plasticity techniques can effectively be combined for ML applications is less known. To develop an SP technique that is more efficient and brain-inspired, this chapter investigates brain plasticity techniques from a macro perspective and discusses the relationships between them. Furthermore, ASNN applications that have already used biologically inspired plasticity techniques have been reviewed. A special emphasis is made on the synergistic application of STDP learning, neuron excitability regulation (a part of Intrinsic Plasticity) and SP for robust and efficient learning.

7.1 Plasticity in the Brain

Neuroplasticity is a general term used to describe changes in the brain, both functionally and structurally, in response to experience (Mateos-Aparicio & Rodríguez-Moreno, 2019). In terms of ASNNs, the changes in synaptic weights according to incoming training samples (i.e., learning) hold a resemblance to neuroplasticity. However, the change of synaptic efficacy is only a single adaptation method among many in the brain (James et al., 2017). Plasticity techniques found in the brain can be divided into different spatial and temporal scales (see Figure 7.1). The implications of such plasticity techniques on data processing, memory formation, learning and system maintenance have been discussed in many neuroscientific studies (Abraham et al., 2019; Bi & Poo, 1998; Chechik et al., 1999; Desai et al., 1999; Eriksson et al., 1998; Frémaux & Gerstner, 2015; Frick & Johnston, 2005; Markram et al., 2011; Moreno-Jiménez et al., 2019; Mozzachiodi & Byrne, 2010; Song et al., 2000; Spiess et al., 2016; Watt & Desai, 2010; Yamaguchi & Miura, 2015). Due to the intricate nature of the brain, the underlying interaction between different plasticity techniques and their coupling processes is yet to be

[17] Artificial Spiking Neural Networks. The term "Artificial" is specifically used to differentiate from biological spiking neural networks also found in this chapter.

understood fully (Aimone et al., 2014). However, researchers have identified how some combinations of plasticity techniques can enhance the functions of the brain.

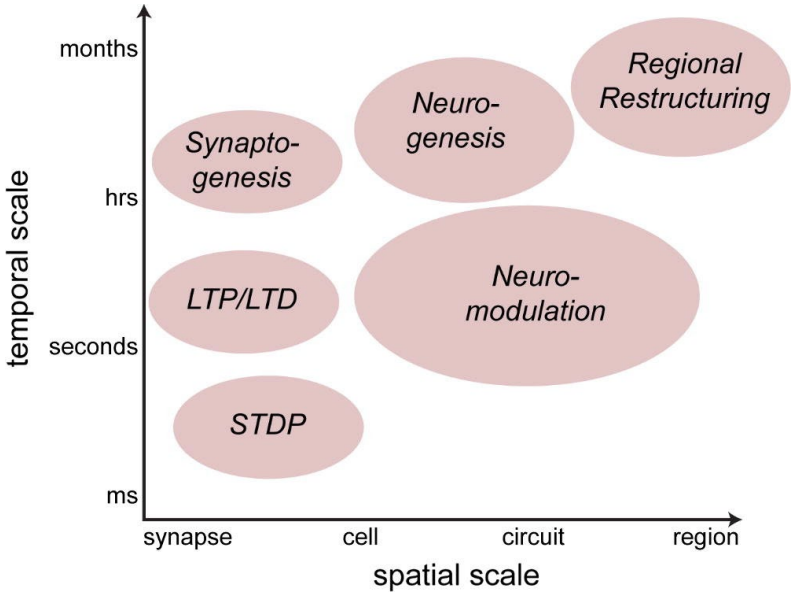


Figure 7.1: **Plasticity methods associated with computation in neurobiological systems**

Note. From "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications", by James, C. D., Aimone, J. B., Miner, N. E., Vineyard, C. M., Rothganger, F. H., Carlson, K. D., Mulder, S. A., Draelos, T. J., Faust, A., Marinella, M. J., Naegle, J. H., & Plimpton, S. J., 2017, *Biologically Inspired Cognitive Architectures*, 19, 49–64, https://doi.org/10.1016/j.bica.2016.11.002, Copyright 2016 by Elsevier. Reproduced with permission.

7.1.1 Synaptic level plasticity

The idea of synaptic efficacy between two neurons changing due to firing sequences was suggested by Donald Hebb in his learning postulate, commonly coined as *neurons that fire together wire together* (2005). This idea of activity-dependent efficacy change was tested in vivo using rat hippocampal brain cultures, and the researchers reported the spike timing between presynaptic and postsynaptic neurons influencing the extent of the synaptic strength change (Bi & Poo, 1998). These findings were further investigated by Song et al. in a computer simulation that produced a mathematical model for this plasticity mechanism, termed Spike Time Dependent Plasticity (STDP) (Song et al., 2000).

STDP refers to the nature of synaptic efficacy change caused by spiking, and if the spike sequences last a longer period, the synapse goes through long-term potentiation (LTP) or long-term depression (LTD) otherwise (refer to Section 5.6.2 for equations and further explanation). Suppose the presynaptic neuron activation causes a repetitive spiking in the postsynaptic neuron within a certain time frame (20

milliseconds in rat hippocampal culture); in that case, the synapse in between goes through an LTP phase and if the spiking sequence is reversed, an LTD is caused (Bi & Poo, 1998).

The function of STDP being dependent only on local activity makes it an efficient data processing technique compared to mechanisms that require feedback for synaptic weight modification (Song et al., 2000). Moreover, the ability of STDP to produce irregular postsynaptic firing whilst remaining highly sensitive to the input can be advantageous in complex data encoding and in maintaining sensitivity to low-intensity inputs (Song et al., 2000). However, it is highly unlikely that STDP works alone, primarily because it is hard to maintain the stability and functionality of neuron circuits without additional plasticity mechanisms to prevent extreme spiking or weight modifications (Watt & Desai, 2010). Exploring further into STDP, using simulations, researchers have found the STDP to favour LTP, even neglecting precise spike timing at instances (Froemke et al., 2006). This preference for LTP would potentially push neurons from optimum functional regimes (Tsodyks, 2002), causing potentiation loops to distort the information representation. Therefore, STDP alone could cause *runaway synaptic potentiation* or over-excitation of neurons (Chen et al., 2013). However, such dysfunctional activities are not observed in healthy brain functions, implying internal homeostatic mechanisms exist.

7.1.2 Cellular level plasticity (Intrinsic Plasticity)

Intrinsic Plasticity (IP) is the ability of a neuron to regulate its excitability using voltage-regulated conductances as a response to incoming signals, a phenomenon found in biological systems (Desai et al., 1999; Turrigiano & Nelson, 2004; W. Zhang & Linden, 2003). This mechanism helps neurons to be sensitive to a wide range of small and large synaptic inputs, which is a vital requirement in learning and development (Desai et al., 1999). This is also argued as a memory-storing method, which is non-synaptic and operates over larger temporal scales(Abraham et al., 2019).

In a neuron plasticity study, Desai et al. reported increased excitability of rat visual cortical neurons after an input deprivation period of two days (Desai et al., 1999). In other words, they became more sensitive when the neurons were blocked from input signals. In the same study, authors reported a drop in excitability after a constant stream of inputs. This implies that one of the methods by which neurons remain sensitive to different input levels is by changing their current to firing rate curve using internal electrical properties (Watt & Desai, 2010). Although the exact input-to-output function in IP is not known, one idea suggests neuronal coding to operate in such a way that information transmission is maximised (Laughlin, 1981). This means that if neurons use all their response levels with the same probability, the entropy of output firing distributions in a given neuron (or circuit) should maximise (i.e., should produce an exponential firing rate distribution under a fixed mean firing rate) (Triesch,

2007). Neurobiological studies have reported that neurons produce exponential firing rate distributions (Brons & Woody, 1980).

IP is advantageous in maintaining network stability and energy efficiency (Watt & Desai, 2010). Maintaining network firing stability is vital, especially in an environment where plasticity techniques such as STDP are in operation; STDP by itself is prone to drive networks into potentiation or depression loops that produce runaway synaptic dynamics (Chen et al., 2013). In extreme cases of such dynamics, the information representation becomes inaccurate while producing spikes needlessly, exerting more energy (Watt & Desai, 2010). However, biological organisms such as the brain operate in strict energy budgets, firing not too high or not too low but at an optimum rate to transfer maximum information to remain both energy efficient and accurate (W. Zhang & Linden, 2003). One of the ways this is achieved in IP is to increase or decrease the threshold when the input stimulation is too high or too low.

7.1.3 Circuit and regional-level plasticity

The overall processes of neuron development and reduction are often discussed separately under the topics of synaptogenesis, synaptic pruning, neurogenesis, and apoptosis (i.e., programmed cell death). Synaptogenesis and neurogenesis are the two mechanisms related to the generation of new synapses and neurons, respectively (Butz et al., 2006), whereas synaptic pruning and apoptosis refer to the elimination of synapses and neurons (Iglesias & Villa, 2006). In the process of network development, these processes are collectively operating in the mammalian brain (Rakic et al., 1986).

The process of proliferation, which is the overgeneration of neurons and synapses, is active in the human brain roughly until the age of two and then goes through a 50-60% synaptic pruning throughout adulthood (Peter R., 1979). Presumably, this is a viable option for network development, considering the amount of genetic information required to construct a specifically designed network structure. Moreover, the idea of a fixed structural plan would be a hindrance to experiential learning (Aimone et al., 2014). Apart from early development, these processes are active throughout life (Butz et al., 2006). The existence of adult neurogenesis, in particular, has been argued. Still, the growth of new neurons (and synapses) has been observed in hippocampal areas assisting the process of learning novel information (Eriksson et al., 1998).

The benefits of nerve cell development and deletion are two folds. Firstly, the simplification and accommodation of the network initiation process (Navlakha et al., 2018). Secondly, energy efficiency enhancement and improved learning (Deng et al., 2010). Therefore this form of structural plasticity is crucial in energy-efficient learning and the formation of new memory (Aimone, 2016).

7.2 Biologically inspired plasticity for Machine Learning

As discussed in the previous section, biological plasticity can be separated into different levels of spatial and temporal scales (see Figure 7.1). Similarly, the plasticity techniques used in ML can be identified at synaptic, cellular and circuit levels. This section discusses these plasticity techniques found in ML and highlights development opportunities with inspiration drawn from biology.

7.2.1 Synaptic and cellular level plasticity

Diehl et al. proposed an unsupervised learning algorithm based on the exponential STDP rule (i.e., classical STDP found in (Song et al., 2000)) (Diehl & Cook, 2015). In addition to STDP, homeostatic adaptive threshold plasticity (i.e., intrinsic plasticity) and lateral inhibitions were incorporated in the proposed two-layer ASNN setup. The algorithm demonstrated 95% accuracy in digit recognition using the MNIST data set; the highest accuracy performance was demonstrated in an unsupervised setup during the time. Researchers used the average spiking rate of selected neurons to distinguish the class. The authors reported stability in learning over time without overfitting or excessive increase of weights, both attributed to the competitive learning accommodated by STDP and IP. Kheradpisheh et al. used a similar learning mechanism for object recognition tasks in a deep ASNN setup and reported progressive learning of salient features (Kheradpisheh et al., 2018). The authors also tested image recognition with the MNIST dataset and reported an accuracy of 98.4%. However, this method used an additional Support Vector Machine (SVM) module as the classifier in the last step of processing. In comparing these methods, one can argue that the use of SVM is somewhat biologically implausible.

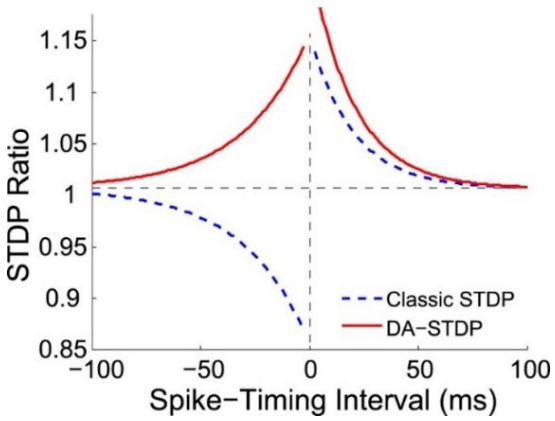


Figure 7.2: Classic STDP function (Bi & Poo, 1998) and Dopamine-modulated STDP (J. C. Zhang et al., 2009)

Note. "A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule", by Hao, Y., Huang, X., Dong, M., & Xu, B. , 2020, *Neural Networks*, 121, 387–395, https://doi.org/10.1016/j.neunet.2019.09.007 , Copyright 2019 by Elsevier. Reproduced with permission.

Hao et al. presented a biologically plausible supervised learning algorithm for ASNN and reported 96.7% accuracy with the MNIST dataset (Hao et al., 2020). This method differed from prior discussed methods because it used a different version of STDP, namely, Dopamine-modulated STDP (DA-STDP) (J. C. Zhang et al., 2009). The weight-changing function of this method is different from the classical STDP previously discussed (Bi & Poo, 1998) . Apart from DA-STDP, the authors used IP and synaptic scaling for learning. Synaptic scaling is a form of homeostatic plasticity which adjusts all synaptic weights to a common scale to help prevent unstable behaviour (Watt & Desai, 2010). This type of synaptic plasticity method has been found in the neocortex (Desai et al., 1999). Apart from these unsupervised methods, a supervised learning method was used for classification based on the average spiking of a specific neuron assigned to represent a class, similar to the method introduced by Diehl and Cook (2015). In their work, the authors also suggested two guidelines for an ASNN to be biologically plausible. Firstly, the neurons in the network should integrate temporal spikes and output temporally encoded spikes. Secondly, the computations and training within the network should be based on spikes completely.

The studies conducted by Diehl et al., and Hao et al., justify the use of biologically plausible learning techniques of STDP and IP in real-world machine learning by producing accurate results that are on par with the well-established deep learning methods (Diehl & Cook, 2015; Hao et al., 2020). The key drawback in these studies is the lack of discussion linking plasticity techniques, spiking behaviour and performance. Questions such as: what type of spiking behaviour is desired for better performance? How to accommodate such spiking behaviour through the unsupervised STDP, IP and synaptic scaling? What are the implications of energy consumption using such methods? Remain unanswered. Without answering these questions, IP tuning will have to rely on optimisation methods to improve the performance of the ASNNs suggested. A preliminary investigation to answer these questions was presented by Zhang et al. in which an IP rule based on the output spiking behaviour was tested (2019). This rule emphasised obtaining an exponential firing distribution with a fixed mean firing to ensure maximum information transfer from input to output, according to entropy maximisation (Bell & Sejnowski, 1995). However, this study does not consider biologically plausible techniques such as STDP for learning at a synaptic level. A detailed discussion of experimentation on this topic can be found in Chapter 7 .

To answer previously stated questions, neurobiological studies provide certain insights worth exploring. One of them is the exponential firing distributions observed in neuron circuits (Laughlin, 1981) which can be amalgamated with information maximisation theory to tune plasticity techniques (Bell & Sejnowski, 1995). Preliminary experiments using this observation have been conducted by trying to derive a transfer function between input and output spiking (W. Zhang & Li, 2019). However, this method assumes the input spiking rate to be constant, which is not the case in real-world data applications. Another observation in neuron circuits is the emergence of avalanche-like spiking demonstrating power-law behaviour that drives neurons to operate in criticality (Beggs & Plenz, 2003). These studies provide a potential foresight for a targeted spike regulation, in a way in which the plasticity techniques can be guided to produce.

7.2.2 Circuit level plasticity

Inspired by the structural adaptation phenomena of the brain, several ASNN algorithms have used structural plasticity techniques in ML applications (Dora et al., 2016; Draelos et al., 2017; Lobo et al., 2018; Schliebs & Kasabov, 2013; J. Wang et al., 2017; Wysoski et al., 2008). In these studies, the addition of new neurons and synapses are used to accommodate the preservation of new knowledge (Dora et al., 2016; Draelos et al., 2017; Lobo et al., 2018; Schliebs & Kasabov, 2013; J. Wang et al., 2017; Wysoski et al., 2008) whereas pruning or neuron merging is used for selective forgetting knowledge. Some studies overgrow the initial network and prune only the synapses that are not used frequently (Rathi et al., 2019). However, in such studies, the primary goal is to reduce the energy cost even at the cost of network performance.

Structural plasticity plays a vital role, particularly in continuous learning applications where discarding obsolete knowledge and preserving new knowledge are important (Lobo et al., 2018). Especially in applications dealing with non-stationary real-time data streams, where the representations of knowledge change over time (Minku & Yao, 2012), referred to as concept drift, ASNN models require fast adaptation to remain accurate under such input dynamics. Schliebs et al. presented a two-layer ASNN architecture for STD classification in which the output neuron repository could grow with each class sample presented (2013). This model used the RO learning rule (S. Thorpe et al., 2001), which gave prominence only to the first spike. Moreover, the repository of output neurons grew indefinitely as long as input signals were presented. This reduces the biological plausibility and causes memory issues at the hardware level. A solution for this indefinite growth of neurons was introduced where the neurons with obsolete knowledge were pruned, keeping the output neurons repository restricted (Lobo et al., 2018). A similar, two-layer ASNN was presented with an improved RO rule that relied on the exact time of spike to determine patterns related to a given class (Dora et al., 2016). However, this method

did not employ biologically plausible learning rules such as STDP that accommodates learning through spiking synchrony. The online learning algorithm proposed by Wang et al. used the STDP rule with structural adaptation on the hidden layer, which consisted of RBF neurons. This ASNN consisted of three layers of neurons and demonstrated accuracy results lower than non-spiking methods. However, the proposed method learned only through one-pass learning compared to multiple epochs used in non-spiking methods.

In summary, STDP and IP, together as plasticity methods, complement each other and can maintain firing stability. This has been found in both biological and artificial SNNs. However, the literature does not clearly identify guiding rules for the same. The information maximisation rules used by Zhang et al. for the ASNN do not use STDP for learning and define a transfer function between the input and output of the neuron with an assumption that the input spike rates would remain constant (2019). Moreover, the addition of new neurons is associated with new class learning in ASNNs that has been tested to be successful in the classification layer. However, neuron population regulation in the hidden layer is not discussed. Chapter 4 and Chapter 5 of this thesis observed how the size of the hidden layer neuron population functioning with STDP affects the classification performance. Therefore, a gap in knowledge exists in terms of how hidden layer neuron population size should be regulated for efficient and accurate learning.

Chapter 8 Ensemble Plasticity and Network Adaptability in ASNNs

This chapter is comprised of a peer-reviewed journal article submitted to *Neurocomputing Elsevier* (Under review).

8.1 Reference:

Weerasinghe, M. M. A., Parry, D., Wang, G., & Whalley, J. (2022). *Ensemble plasticity and network adaptability in SNNs*. http://arxiv.org/abs/2203.07039

8.2 Author contribution:

M.W.: 87.5%, J.W.:7.5%, G.W.: 2.5%, D.P.: 2.5%

M.W. designed the algorithm, ran the experiments, analysed the data, and wrote the manuscript. J.W supported the development of the methodology. G.W. and J.W. supervised the research. All authors reviewed the manuscript and provided feedback.

8.3 Preamble:

The experiments presented in Chapter 6 indicated a connection between the number of active neurons in the hidden layer, individual neuron excitability and ML performance. The SNNs that performed better had a more significant number of active neurons with increased individual neuron excitability. Although the individual excitability increased, the number of spikes produced by each neuron was fewer than for other networks (undergrown and overgrown networks), which suggested the spiking is distributed across the neurons. Although the SNNs with structural adaptations performed better, the DE algorithm was computationally expensive.

Of the six hyperparameters tuned in Chapter 6 , the LIF threshold proved to be a key contributor to improved performance. This parameter, along with the use of an optimal number of neurons in the hidden layer, ensured sparser firing activity. To incorporate this concept into a learning algorithm, the author investigated the plasticity methods found in the mammalian brain. Here, the spike regulation role of Intrinsic Plasticity (IP) (the ability to adapt neuron excitability according to incoming data) under STDP learning environments maintained the firing stability of the network. Therefore, it was

hypothesised that implementing IP in an STDP learning environment would bring spiking regularisation and activation-based neuron prioritisation[18]. Moreover, when it comes to structural plasticity, the brain follows a proliferation and prune method, where neurons are pruned according to their activity, following *use it or lose it* strategy.

To replace the computationally expensive DE method for structural adaptation, this chapter presents a brain-inspired learning method. In this method, pruning is applied to an overgrown structure that learns using STDP and IP. Experiments were conducted to compare the performances of SNNs that operate with STDP only, STDP+IP, and STDP+IP pruned. Further explorations are conducted to find spike patterns that enable better performance.

8.4 Abstract

Artificial Spiking Neural Networks (ASNNs) promise greater information processing efficiency because of discrete event-based (i.e., spike) computation. Several Machine Learning (ML) applications use biologically inspired plasticity mechanisms as unsupervised learning techniques to increase the robustness of ASNNs while preserving efficiency. Spike Time Dependent Plasticity (STDP) and Intrinsic Plasticity (IP) (i.e., dynamic spiking threshold adaptation) are two such mechanisms that have been combined to form an ensemble learning method. However, it is not clear how this ensemble learning should be regulated based on spiking activity. Moreover, previous studies have attempted threshold-based synaptic pruning following STDP to increase inference efficiency at the cost of performance in ASNNs. However, this type of structural adaptation, which employs individual weight *mechanisms, does not consider spiking activity for pruning which is a better representation of input stimuli*. In this study, we aimed to investigate spike regulation using ensemble learning in an information theoretic approach to optimise ASNNs for spatiotemporal data classification. We envisaged that plasticity-based spike regulation and spike-based pruning would result in ASSNs that perform better in low-resource situations. In this paper, a novel ensemble learning method based on entropy and network activation is introduced, which is amalgamated with a spike-rate neuron pruning technique, operated exclusively using spiking activity. Two electroencephalography (EEG) datasets are used as the input for classification experiments with a three-layer feed-forward ASNN trained using one-pass learning. During the learning process, we observed neurons assembling into a hierarchy of clusters based on spiking rate. It was discovered that pruning lower spike-rate neuron clusters resulted in increased generalisation or a predictable decline in performance. Moreover, the networks demonstrated

[18] In computational biology studies Savin et. al. demonstrated how STDP and IP could make neurons tune to respond to particular features, where level neuron activation correlated with its contribution to pattern separation

avalanche-like spiking activity under ensembled learning. The findings of this study draw attention to the ability of ensemble learning to push ASNNs towards criticality and of neuron pruning to allow ASNNs to 'forget' unimportant information. The results of this work illustrate how plasticity-based ensemble learning in conjunction with pruning can lead to increased robustness and efficiency of ASNNs in ML applications.

8.5 Introduction

The emergence of technologies such as IoT (Internet of Things) has highlighted the need for data processing techniques that are efficient under low resource situations (i.e. limited power consumption and/or memory) (Pfeiffer & Pfeil, 2018). Quite often, these techniques must deal with spatiotemporal data (STD) produced from a network of sensors. Therefore, such techniques need to consider the preservation and use of vital information stored spatially and temporally during the processing. Artificial spiking neural networks (ASNNs) process data using discrete temporal events called spikes. According to the literature, these ASNNs promise greater information processing power and lower resource consumption than those provided by traditional non-spiking artificial neural networks (ANNs) (Maass, 1997). ASSNs have also been found to be ideal for spatiotemporal data processing since they are equipped with an inherent ability to code inputs using spikes on a temporal axis (Pfeiffer & Pfeil, 2018; K. Roy et al., 2019; Tavanaei et al., 2019). In contrast, ANNs require specialised architectures, such as long short-term memory (LSTM) models, to process temporal data (Greff et al., 2017). These advantages of ASNNs have led to recent research efforts in the exploration and development of efficient ASNN algorithms for real-world Machine Learning (ML) applications involving STD (Dora et al., 2016; N. Kasabov et al., 2013).

Biologically inspired ASSN learning strategies, such as plasticity techniques, allow for the local adaptation of synaptic and neuron properties using asynchronous spikes. These plasticity rules can be much more efficient than methods that depend on synchronous adaptation and non-local transmission, such as error backpropagation used in ANNs. Spike time dependent plasticity (STDP) (Abbott & Nelson, 2000; Bi & Poo, 1998; Song et al., 2000) and intrinsic plasticity (IP) (Desai et al., 1999; Frick & Johnston, 2005; W. Zhang & Linden, 2003) are two such biologically plausible rules used in ASNNs for unsupervised learning. A handful of studies have used these plasticity rules ensembled in ML applications for static image data classification and reported performance at par with ANNs (Diehl & Cook, 2015; Hao et al., 2020). However, it is unclear how this ensemble learning should be regulated based on spiking activity to achieve robust performance with STD. Spike-based ASNN regulation is important in enabling efficient learning and can be vital in online and incremental learning applications where spike regulation based on final inference can be too time-consuming.

Neuron pruning has been identified as a key process that eliminates extra or unnecessary synapses and neurons in the biological brain that, and increase the network efficiency (Navlakha et al., 2018). Inspired by the biological findings, ASNNs have adapted pruning in resource-restricted applications (Shi et al., 2019). Past studies of neuron pruning have been following proliferate then prune approach for individual synapses (Rathi et al., 2019; Shi et al., 2019). However, comparing individual synaptic weights to a threshold may cause scalability issues. A probable solution to this would be neuron pruning based on spiking characteristics. Although bio-physical experiments have found such neuron pruning (i.e., apotheosis) to be helpful in network efficiency and pattern recognition (Iglesias & Villa, 2006), less is known about the effects of such mechanisms on ML applications with ASNNs.

As a result of exploring the literature on ASNNs two gaps in knowledge were identified:

1. The insufficiency of theoretically justified methods for spike regulation in ensemble learning
2. The lack of understanding in terms of neuron pruning to increase efficiency

These gaps led to the formation of two central hypotheses in terms of ensemble learning and network adaptability. Under the proposed ensemble learning method, if all the neurons are activated and entropy is minimised (i.e., the measure of uncertainty), it is hypothesised that the ASNN would perform better in terms of robustness and efficiency. This conjunction was based on the preliminary work of Lazar et al. (2007). Secondly, if neurons of less spiking rate are pruned, then the performance of the ASNN trained with ensemble learning will gradually reduce; as conjectured based on the work of Iglesias et al. and Savin et al. (2006; 2010) that demonstrated the hierarchical formation of neuron clusters based on input features. To test the hypotheses, we employed a feed-forward ASNN architecture with a hidden layer trained using a single epoch. The performance was tested in a rigorous ML framework to understand training and inferring performance separately. The algorithm is used to classify Electroencephalography (EEG) data, a form of STD. In this study, we make the following contributions:

- The introduction of a novel ensemble learning method based on information theory.
- The presentation of a novel neuron pruning method based on spiking rate.
- An exploration of avalanche-like spiking dynamics that may contribute to better pattern separation.

The rest of this article is organised as follows: Section 8.6 presents related work, Section 8.7 introduces the methodology and experimental framework, Section 8.8 presents the experimental framework, Section 8.11 discusses the results and relates them to previous work, and Section 8.11 concludes with limitations & future work. It should be noted that for the rest of the article, the terms plasticity and learning are used interchangeably.

8.6 Related work

8.6.1 Standalone IP

IP was initially investigated in vitro by Stemmler and Koch (Stemmler & Koch, 1999) and presented an unsupervised IP rule using Hodgkin-Huxley (HH) neuron (Hodgkin & Huxley, 1952). This application of IP involved adjusting voltage-dependent conductance via the properties of voltage-gated ion channels. The authors elucidated the impact of IP on output firing probability distribution and its utility in maximising mutual information. Since exponential distributions yields the highest entropy from all distributions for a non-negative random variable (i.e. given the mean is stationery), Triesch introduced an IP rule to reduce Kullback-Leibler (KL) divergence between desired exponential and actual firing probability distributions (2007). In a similar approach, Weibull distribution was considered instead of exponential (C. Li, 2011). Both these studies were based on non-spiking neurons and computer-generated data. Other researchers have extended the idea of information maximisation and studied the effect of IP on SNNs (C. Li & Li, 2013; A. Zhang et al., 2019; W. Zhang & Li, 2019). Real-world applications of image recognition and speech classification were presented by Zhang and Li (2019), in which IP was used with a Liquid State Machine (LSM). The synaptic weights of the LSM were fixed, and IP was tuned using R resistance and τm membrane time constant of each neuron to achieve the desired exponential firing distribution following the KL divergence method (Triesch, 2007). This method was modified to adjust the intrinsic properties of the neurons according to backpropagated error gradient (W. Zhang & Li, 2021). In these studies, the weights are set randomly and kept constant or changed according to backpropagation (W. Zhang & Li, 2021), where learning rules are only dependent on the spiking rate.

8.6.2 IP and STDP - Ensemble Learning

STDP is an unsupervised learning mechanism used for synaptic weight updates based on spike timing. The effects of applying STDP with IP were discussed by Savin, Joshi and Triesch (2010). The authors reported neurons adapting to be responsive to independent components from the input signal (i.e., ICA-like learning) and presented successful de-mixing of Gaussian signals. Another implementation of IP in a Liquid state machine (LSM) setup used Inter-Spike-Interval (ISI) for neuronal threshold tuning and

reported better performance with the ensemble method over standalone STDP (X. Li et al., 2018). A simplistic model of IP is presented in combination with STDP by Lazar (Lazar et al., 2007), which reported the advantages of such methods in time series prediction and highlighted the importance of neurons operating in criticality for better performance. In this study, IP was implemented using the equation,

$$T_i(t + 1) = T_i(t) + \eta_{ip}(x_i(t) - k/N) \quad \textbf{(26)}$$

This IP algorithm increases the threshold if a spike has occurred in the previous time step T_i and decreases if not. This forces every unit in the network to be sensitive to incoming stimuli. η_{ip} is the small IP learning rate, $x_i(t)$ denotes the state (i.e., 1 or 0) at time step *t*. The total number of neurons in the network is *N*, and *k* is decided based on the desired number of neurons that should be active at a given time step. In an ML setup, STDP and IP ensemble learning was discussed (Diehl & Cook, 2015; Hao et al., 2020), where authors reported state-of-the-art accuracy levels. However, these studies have not been implemented to discuss the effects of ensemble learning at a spike level nor to explore the impact of such methods at different stages of training and testing. This led us to the question of how ensemble learning should be regulated at a spike level for better performance in ML.

8.6.3 Structural adaptation in ASNNs

In this section, we define any form of network architecture change as a structural adaptation technique. This includes adding and/or pruning neurons and/or synapses. Biologically, the brain is believed to go through a pruning after proliferation strategy (Morris, 1999), and this process has been observed in-vivo (Turney & Lichtman, 2012). Presumably, it is much more likely for a brain circuit to get refined for better efficiency and robustness over time. However, most SNN applications in ML can be categorised as methods implemented either to enhance robustness (Dora et al., 2015; S. Roy & Basu, 2017; J. Wang et al., 2014; Wysoski et al., 2008)or efficiency (Rathi et al., 2019; Shi et al., 2019) separately.

One learning algorithm based on neuron growth was introduced specifically for performing image recognition (Wysoski et al., 2008). During the training period, at each passing of a sample, a new neuron map is evolved at the final layer. Researchers have not implemented a temporal learning method such as STDP. A similar approach involved adding neurons to the hidden layer in a winner-takes-all circuit to make a selected neuron fire first for a given class of sample during training (J. Wang et al., 2014).

STDP and anti-STDP rules were used between the output and hidden layers of the network. In another study, a two-layer SNN with a rank order rule (RO) for weight updating was used to add a new neuron at the output layer for each training sample (Dora et al., 2015). Synaptic rewiring under STDP was introduced in a separate study where a connection matrix is updated based on a dynamic threshold (S. Roy & Basu, 2017). Synaptic rewiring takes place after each training epoch. These methods have not been implemented to deal with multi-spike environments or with consideration of biological plausibility. This may affect the scalability and adaptability of the network (Navlakha et al., 2018) and restrict the algorithm from making use of different neuronal dynamics for efficient coding. Therefore, we intended to follow a pruning after proliferation strategy, which is biologically plausible.

Pruning-based techniques of structural adaptation in the literature focus on the elimination of synapses to increase processing and implementation efficiencies. Rathi, Panda and Roy presented threshold-based synaptic pruning where weights are updated according to the STDP rule (2019). The threshold is a fixed value determined after evaluating the accuracy of the SNN. The same study presents the effects of weight quantisation which adjusts all weights to a fixed range of values after training is completed. A different form of structural adaptation was presented by Shi et al., where synapses are assigned to their lowest value based on the efficacy decided by a threshold at each epoch (2019). This leaves the possibility for a neuron to get activated at a later epoch, and synaptic elimination takes place only at the end of a training session.

The dynamic spiking activity produced by a neuron represents the input stimuli. This representation is made possible by synaptic (i.e., STDP) and non-synaptic plasticity (i.e, IP) in an ensemble learning setup. Therefore, we propose the spiking activity of a neuron to be a better representation of the input stimuli rather than the magnitude of individual synaptic weight. Hence, in contrast to the methods discussed, we focused on pruning neurons instead of individual synapses, which is arguably more efficient in terms of computational cost. This form of programmed neuron cell death, known as apoptosis, is observed in biology (Iglesias & Villa, 2006).

8.7 Methodology and Data

In order to investigate ensemble learning and neuron pruning, we employed a feed-forward SNN architecture (see Figure 8.1). This network is equipped with threshold adaptable LIF neurons, a combination of STDP and IP for unsupervised learning, and a classifier based on semi-supervised learning. The network is trained under batch mode with one-pass learning.

8.7.1 Leaky Integrate and Fire Neuron

The hidden layer of the SNN consists of LIF neurons, the behaviour of which can be mathematically modelled using a resistor-capacitor circuit (Gerstner & Kistler, 2002, p. 95). We selected this model due to its processing efficiency and simplicity over more biologically plausible counterparts, such as HH model (E. M. Izhikevich, 2004). The model introduced by Izhikevich (2003) is also capable of demonstrating many types of spiking behaviour with low computational cost. However, this model consists of four independent parameters and two differential equations. Since our main objective was to investigate ensemble learning and network adaptability, we opted for the more simplistic, computationally efficient solution, which is the LIF. Moreover, unlike regular LIF use, since our implementation involves IP, we expected more dynamism in firing types (E. M. Izhikevich, 2004).

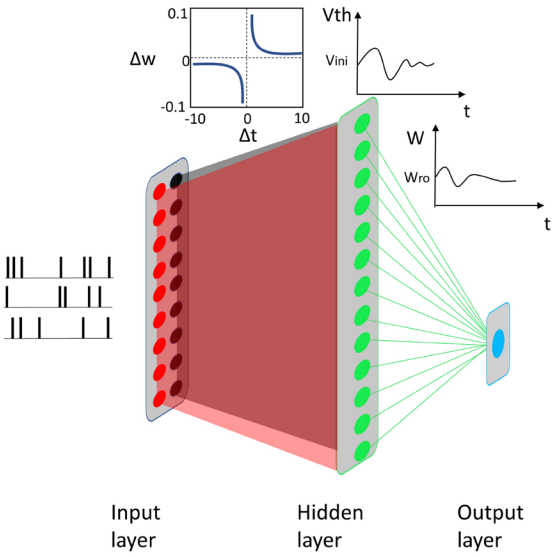


Figure 8.1: SNN algorithm used for experimentation with a single hidden layer. The input layer consists of LIF neuron pairs capable of propagating both excitatory and inhibitory spikes (Red-excitatory and black-inhibitory). Synapses between input and hidden layer

$$\tau_m \frac{dv_t}{dt} = v_{rest} - v_t + \mathrm{R}I_t \quad where\ \tau_m = RC \tag{27}$$

Here, the time constant τ_m is the multiplication of resistance *R* and the capacitance *C*. The membrane potential v_t change with the time dependent input current It, given that the neuron is not within the

refractory period. During refractory, the voltage resets to v_{rest}. Moreover, the spike generation of a neuron is given in the form of a Dirac delta function.

$$\delta(s_t) = \begin{cases} 1, \ if \ v_t = v_{thresh} \\ 0, \quad otherwise \end{cases} \quad \textbf{(28)}$$

A firing s_t in the neuron takes place if the membrane potential v_t reaches threshold voltage v_{thresh}. A firing is represented by 1, and a non-firing by 0.

8.7.2 Spike Time Dependent Plasticity

We used the standard form of STDP (Bi & Poo, 1998) to update weights between the input and the hidden layer. This unsupervised method of learning can be used effectively to learn from local spiking patterns (Chen et al., 2013). In the literature, we find other forms of STDP, such as anti-STDP (J. Wang et al., 2014), power law weight-dependent STDP (Rathi et al., 2019) and Dopamine modulated-STDP (Hao et al., 2020). We decided to proceed with the basic form of STDP, which is the foundation of all the other methods, enabling the findings to be more generalisable. One of the key implications of STDP plasticity is its tendency to produce *runaway synaptic potentiation* where synapses get caught up in potentiation or depreciation loop, which destabilises the network (Chen et al., 2013). The spikes produced under such situations are not truly representative of the input stimuli. Previous literature suggests IP as a remedial action against such destabilisation (Diehl & Cook, 2015).

$$F_{(\Delta t)} = A_+ \, exp^{(-\Delta t/\tau pos)} \qquad \Delta t > 0 \quad \textbf{(29)}$$

$$F_{(\Delta t)} = -A_- \, exp^{(\Delta t/\tau neg)} \qquad \Delta t < 0 \quad \textbf{(30)}$$

$$\Delta w_{ij} = \sum_{k}^{y} \sum_{m}^{z} F(t_i^m - t_j^k) \quad \textbf{(31)}$$

In (29) and (30), the function $F_{(\Delta t)}$ refers to long term potentiation (LTP) and depreciation (LTD) respectively. LTP takes place if the post-synaptic neuron fires after the pre-synaptic neuron where time

gap between the firings, Δt becomes positive. It becomes negative if the firing sequence happens vice-versa, leading to LTD. τpos and τneg denote the positive and negative time windows respectively, for the synaptic modification. Factors A_+ and A_- are used for positive and negative synaptic modification. The cumulative weight change is denoted by $w_{ij}$where post-synaptic firing is considered from *a* to *b* and pre-synaptic from *p* to *q*. Firing at a given time *m* by post-synaptic neuron is given as t_i^m and pre-synaptic neuron firing at time *n* by t_j^k.

8.7.3 Intrinsic Plasticity

Inspired by the IP implementation introduced by Lazar (Lazar et al., 2007), we took a similar approach to implement an IP learning mechanism to operate with STDP. However, instead of the single learning rate introduced in **(26)**, we used two separate learning rates θ_{pos} andθ_{neg}. This enabled us to increase or decrease $v_{thresh}(t)$ at different rates based on incoming spiking activity providing more flexibility around threshold adjustment.

$$v_{thresh}(t) = \begin{cases} v_{thr}(t-1) + N\theta_{pos}v_{init}\ , s(t-1) = 1 \\ v_{thr}(t-1) - N\theta_{neg}v_{init}\ , \quad otherwise \end{cases} \qquad \textbf{(32)}$$

In equation (32 **)**, the current voltage threshold $v_{thresh}(t)$ is dependent on the previous time step's spike event. If the inter-spike-interval (ISI) (Gerstner & Kistler, 2002) is 1ms, then the threshold is increased by a factor of the initial threshold of the neuron v_{init}. The factor is decided by multiplying learning rate $\theta_{pos}$by number of neurons in the hidden layer *N*. If a spike does not occur, the threshold is reduced by a learning rate of θ_{neg} increasing the spiking probability of the neuron. Both learning rates are small allowing IP to take place throughout the training period avoiding early saturation.

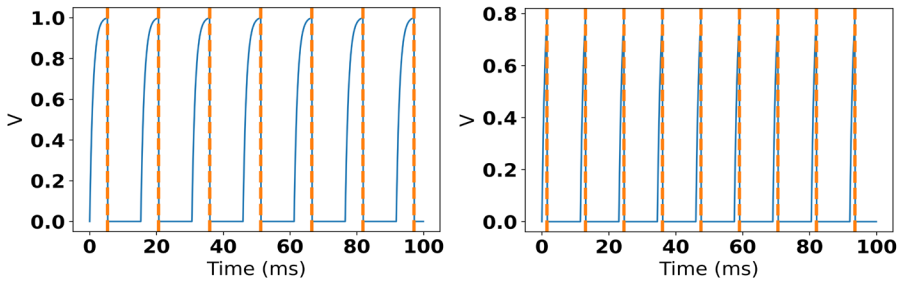


Figure 8.2: Spiking demonstration of the LIF with multiple threshold levels. Membrane potential is represented in blue and spike emissions are represented in orange dashed lines. a) Spiking activity at 1v threshold; b) Spiking activity at 0.8v threshold

In order to adjust $\theta_{pos}$and θ_{neg} (i.e., the learning rates controlling the threshold), we formulated a guided method with two dependent parameters, namely information entropy (C. E. Shannon, 1948) and active neurons (Lazar et al., 2007). We hypothesised the classifier to make fewer errors when the uncertainty is at its lowest, given that the maximum number of neurons were active (see Figure 8.4). In information theory, entropy is a measure of uncertainty or the average amount of information stored in a random variable (C. E. Shannon, 1948), whereas a high number of active neurons contributes towards better pattern separation (Lazar et al., 2007). Entropy was calculated based on the spiking probability distribution produced during training. This method is different to SpiKL-IP tuning method introduced by Zhang and Li (2019), where output spiking is adjusted using reciprocals of LIF membrane time constant and resistance to produce an exponential firing distribution. Our method is more in line with (Lazar et al., 2007) and (X. Li, Chen, et al., 2017) using direct threshold adjustments.

$$H(X) = \sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i) \qquad (33)$$

For the entropy equation (33), we calculated the probability of spiking rates $\mathrm{P}(x_i)$, according to spiking rate probability density function (PDF) of the entire network. This PDF was obtained after a training batch was processed. In other words, our method evaluates the PDF of the entire neuron population to make changes to the excitability of all neurons. This consideration of the global activity is influenced by underlying local modifications of STDP.

8.7.4 Neurons Pruning

Neuron pruning is found to be activity dependent in biological systems (Navlakha et al., 2018) and considered as an integral part of neuronal development (Yamaguchi & Miura, 2015). Iglesias and Villa investigated apoptosis in a large-scale ASNN trained using STDP (2006). This study reported network stabilisation and the emergence of cell assemblies producing recurrent spiking sequences. The decision for apoptosis was based on neuron firing rate.

We adopted a similar approach where the decision to prune a neuron was based on the spiking rate, calculated after completing the training cycle (i.e., STDP+IP only). With the introduction of IP, which is found to be capable of learning independent features from the input (Savin et al., 2010), we hypothesised the spiking rate of a given neuron(i.e., after training) to be proportional to its contribution towards pattern separation. It is important to note that the computational implementation of this method is a deactivation rather than a complete pruning. The computational cost of this approach is lesser compared to individual synaptic pruning, which requires checking each synaptic weight against a threshold (Rathi et al., 2019; Shi et al., 2019). Moreover, this method does not require the classifier to be trained since the pruning is solely based on spiking activity rather than network performance.

$$C(x) = \sum_{i=0}^{n} [\![S_i = x]\!] \text{ if } x = 1 \quad \textbf{(34)}$$

$$P(x) = \frac{C(x)}{n} \quad \textbf{(35)}$$

Considering x to be a neuron in the hidden layer, our algorithm maintains a count of firing during the training period $C(x)$. In the equation, n is the final time point of the training simulation. If the spiking rate is higher than a threshold decided based on the firing rate distribution, it is preserved while the rest is pruned.

8.7.5 Classifier learning

The classifier of the network is inspired by (N. Kasabov et al., 2013) and follows a semi-supervised approach for sample classification. Every training sample is represented by an output neuron which is evolved at each propagation. All the hidden layer neurons are connected to this output layer via

excitatory synapses where rank order (S. Thorpe & Gautrais, 1998) rule is used for weight initiation and spike counting is used to update the weight.

$$w_{ij} = \alpha \, . mod^{order(i,j)} \quad \textbf{(36)}$$

$$w_{ij}(final) = w_{ij}(initial) + \sum_{i=0}^{t=n} \mathrm{d} \quad \textbf{(37)}$$

When *i* pre-synaptic neuron is connected to *j* postsynaptic neuron, the weights are initiated according to the exponent of the modulation factor *mod*. The learning parameter is denoted by α. For the first spike arrival $order(i,j)$ is 0, therefore, it gets the highest weight, and other weights are allocated in a decreasing manner. Following the weight initiation, a drifts parameter d is used to increase or decrease the initial weight according to spike arrival at each time step *t*. At the inference stage, the similarities between neurons evolved are calculated using euclidean distance to label the testing sample.

8.8 Experimental Framework

8.8.1 Data sets used

To test the methods discussed, we used electroencephalogram (EEG) data in a classification setup. We selected two datasets to cover EEG signals related to motor movements and human emotions. For both datasets, samples were stored in comma-separated values (CSV) files. The format of a file is represented by an *[m,n]* matrix where m and n refer to EEG channels and time points *(t ε N),* respectively. Each value in the data represents the instantaneous voltage (x_t ε *Q*) recorded in microvolts. The first data set used was from a study conducted previously to test the feasibility of SNNs in recognising wrist positions (Taylor et al., 2014). EEG was collected from three healthy participants performing wrist flexion, extension, or rest. Each position was held on command for two seconds with the participants' eyes closed to collect EEG which was sampled at 128 Hz. A 14-channel EEG cap was used to collect data from locations defined under international 10-20 locations. No additional artefact removal or filtering process was carried out. The final dataset consisted of 60 samples in a 14 by 128 matrix with 20 samples for each class (i.e., wrist flexion, extension, or rest). This data set will be referred to as the Wrist dataset hereon.

The second dataset is widely used for emotion recognition tasks, commonly referred to as the DEAP dataset (Koelstra et al., 2012). The DEAP data set consists of EEG samples collected from 32 participants while watching 40 one-minute video clips. A 32-channel EEG cap was used with electrodes placed according to international 10-20 locations. Each participant rated the video on scales of valance, arousal, dominance and liking using a self-assessment manikin. We used the pre-processed dataset, which is free of artefacts caused by eye movement and filtered using a 4 Hz to 45 Hz bandpass filter and downsampled to 128 Hz. We removed six non-EEG data fields and applied an averaging window on the time vector as a data reduction process to expedite data processing during experimentation (Golmohammadi et al., 2019). The window size was set to 32, which enabled a single experiment step of 5-fold cross-validation and 70/30 split testing to complete within 30 minutes (Experiments were conducted in a core i7 machine with 16GB RAM). The Final pre-processed dataset consisted of 1086 EEG samples labelled into two classes: *Low arousal = arousal ≤ 5 and high arousal = arousal > 5*. Each class consisted of 543 samples in a 32 by 252 matrix.

8.8.2 Data encoding

EEG is a time-varying analogue signal where instantaneous voltage amplitude is recorded. Since artificial spiking neurons are restricted to spiking or non-spiking states only (i.e., one or zero computational representation), the input voltage amplitudes of EEG signals should be converted to a spiking equivalent. In this study, we used Address Event Representation(AER) encoding since it operates based on the rate of voltage change with a high processing efficiency (Delbruck & Lichtsteiner, 2007). The rate of voltage change is a strong indication of neuronal events, and the user-defined threshold in AER allows sensitivity adjustments to such rate changes. This threshold was set to 0.5 according to the findings of our previous study (Weerasinghe et al., 2021).

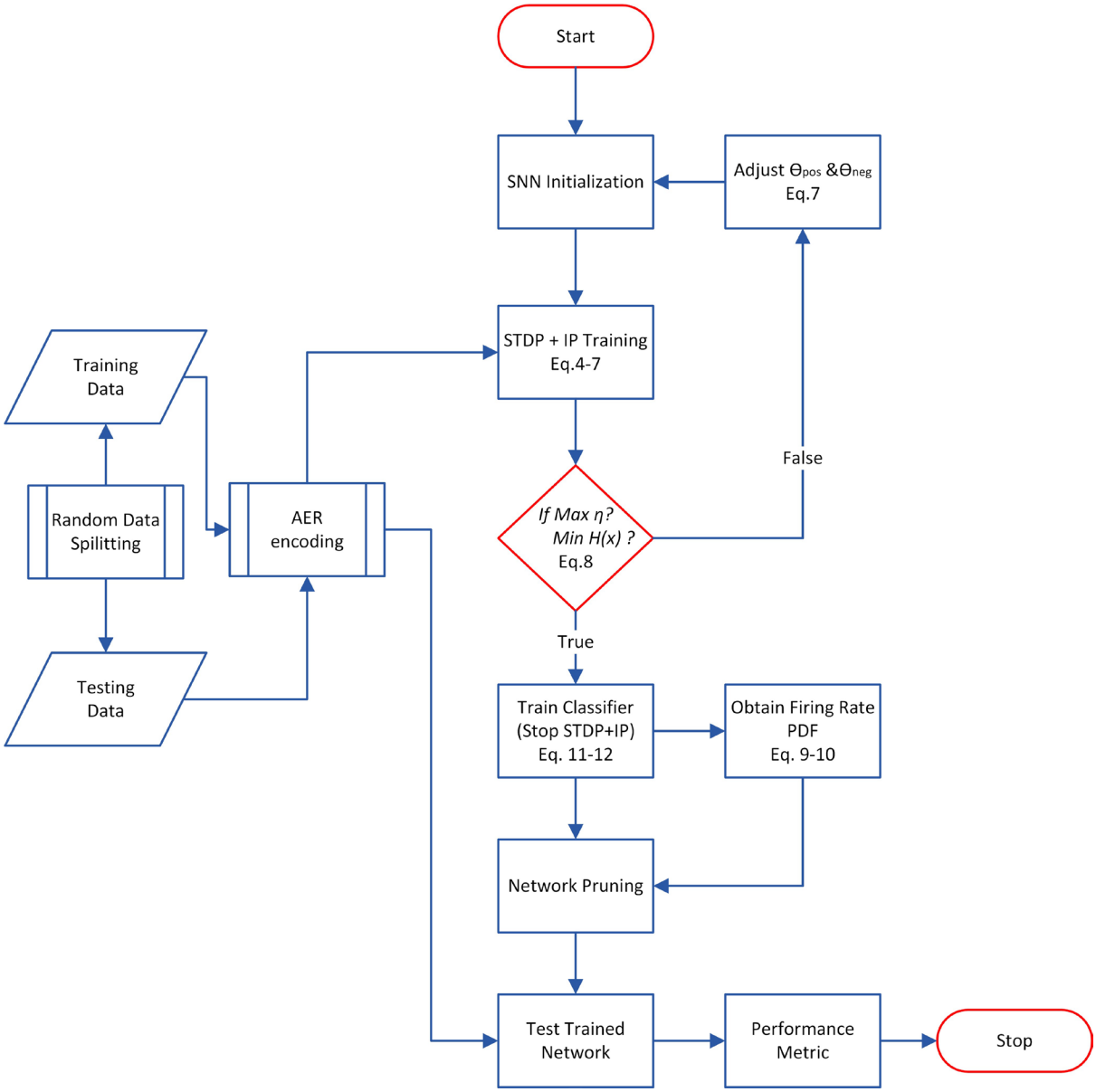


Figure 8.3: Flow chart representation of the ensembled learning and pruning process. Analogue EEG signals are converted to spikes using Address Event Representation (AER) (Delbruck & Lichtsteiner, 2007) encoding method given in Figure 8.13. STDP (Equations (29),(30) and (31)) and IP (Equations (32)) implemented as an ensemble learning method to operate simultaneously. If average active neurons η is maximised, entropy of the spiking PDF is evaluated (Equation (33)) to adjust IP learning rates, θ_{pos} and θ_{neg}. Once the classifier is trained (Equations (36) & (38)), network pruning is applied (Equations (34) & (35)).

8.8.3 Performance evaluation criteria

To evaluate the performance of each method, we selected three metrics, namely: classification accuracy, F1-score, and Cohen's Kappa. We evaluated results using these metrics under 70/30 training testing split and five-fold cross-validation (CV). Before training the network, data is split (i.e., 70% for training and 30% for testing), and CV is implemented using the training set only to analyse the training data fit. We selected fivefold cross-validation considering the size of the training data set. After CV, the network is reinitialised and retrained with training data, followed by testing with 30% of unseen data. Apart from the standard accuracy measurement, we included the F1-score to represent the precision and recall of the classifier and Kappa static to compare performance against random chance. We tested each method for both pattern separation capability(i.e., Robustness) and model performance under changing input and/or output environments(i.e., Adaptability) (Navlakha et al., 2018). The SNN introduced here learns through one-pass learning and we have run each simulation 30 times with randomisation incorporated in sample selection, weight initiation and input mapping. This challenges the model at each train-test cycle and would help assess the adaptability of the algorithm.

8.8.4 Experiment layout

Experiments were conducted under three comparable approaches. Firstly, both datasets were modelled using only STDP. In the second approach, IP was applied with STDP (i.e., ensemble learning). Thirdly, a model trained with ensemble learning was pruned based on the firing rates of the neurons of the hidden layer. The initial networks consisted of 200 and 300 hidden neurons for Wrist and DEAP datasets, respectively. We changed the initial network sizes due to the difference in sample sizes of the two datasets. Throughout the experimentation, non of the hyper-parameters (i.e. set before training) were changed, enabling a fair comparison between the approaches experimented. Only the IP learning rates were tuned to regulate spiking and pruning rate thresholds were changed to explore the effects comparatively. The IP learning rates were restricted between 1 to $1\times10-10$ after initial experimentation. These smaller values enable the network to produce balanced firing during the entire training period (Lazar et al., 2007). The pruning thresholds were decided based on the activation levels of clusters of neurons observed. All hyper-parameter values of the ASNN can be found in Table 8.3.

8.9 Results

8.9.1 Wrist dataset

In this section, we present the results obtained from different approaches experimented with: STDP only, ensemble learning (STDP+IP) and network trained with ensemble learning pruned. The raster

plots and firing distributions presented were obtained after propagating the training data set through trained networks.

STDP and IP combined learning

The key requirement for IP learning is the selection of values for θpos and θneg (According to the IP equations (32), here we have made reference to both values in terms of magnitude only). In general, if θpos increased above 0.001 with θneg set to zero, the spikes vanish towards the end of the training cycle (i.e., when 70% of data is used for training). If the conditions were reversed, the network became over-activated (see Figure 8.12a).

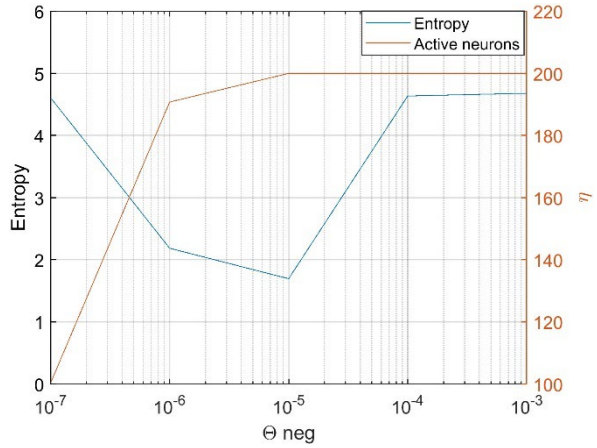


Figure 8.4: Result of sweeping θ_{neg} from 1×10^{-7} to 1×10^{-3} on η (Orange) and Entropy (Blue) while θ_{pos} is kept constant at 1×10^{-3}

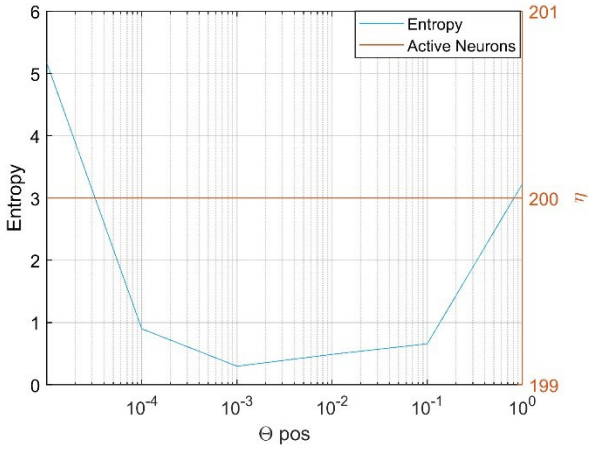


Figure 8.5: Result of sweeping θpos from 1×10^{-5} to 1 on η (Orange) and Entropy (Blue) while θneg is kept constant at 1×10^{-3}

As shown in Figure 8.4 when θpos & θneg were set to 1×10^{-3} and 1×10^{-5}, respectively, the ASNN produced minimum uncertainty under all neurons being activated at the hidden layer as per our

hypothesis. We used the same heuristic method for tuning IP learning rates for both data sets (See Appendix C for learning rates and other hyperparameter values).

When comparing the spike train differences between Figure 8.6a and Figure 8.6b, the sparseness of the latter is evident. This sparseness is observed both temporally (i.e., along the time axis) and spatially (i.e., among neighbouring neurons). Interestingly, within the sparsity observed in Figure 8.6b, scattered bursts of spiking emerged in each neuron, as seen in the zoom enlargement. Standalone STDP only seemed to operate the hidden layer neurons in either firing with high or low-frequency regimes, with occasional neurons maintaining firing rates in between, as per Figure 8.6c. This indicates a high number of neurons showing low spiking rates, with peripheral spiking rates being scattered, fitting into a decaying exponential. In contrast, Figure 8.6d shows a left-tailed normal distribution with ensemble learning.

In terms of spiking rates, STDP recorded 0.032 (±0.003, range = 0.01), whereas ensemble learning recorded 0.009 ($\pm\ 2 \times 10^{-4}$, range = 0.001) over 30 cycles of random testing conducted. This translates to better efficiency in terms of information encoding with ensemble learning: which utilises 3.6 times fewer spikes on average for information encoding compared to standalone STDP (*two-sample t-test*, $p < 0.05$).

Before pruning was applied, we tested the methods of STDP only and ensemble learning using the classifier presented in 3.4. We ran 5-fold cross-validation and a 70/30 split testing across the Accuracy, F1-Score and Kappa matrices. The ensemble learning demonstrated better robustness with increased average performance and fewer outliers compared to standalone STDP, as per Figure 8.7. However, when considering the results of cross-validation and split testing in ensemble learning showed a slight over-fitting.

Network Pruning

Based on the pruning hypothesis in Section 8.7.4, we selected three pruning rate thresholds from the network firing rate distribution. This selection enabled us to analyse network performance comparatively. The pruning spike-rate thresholds were 0.017, 0.052 and 0.087, which pruned on average 17%, 21% and 24% of the network, respectively. These three thresholds represent the lowest clusters of contribution according to our pruning hypothesis.

From the three pruning thresholds, the lowest threshold resulted in the highest, whereas the others yielded lower performances, as shown in Figure 8.8a, in terms of robustness and adaptability. This pattern was observed across all performance metrics. In particular, the higher thresholds dropped performance by 5% on average.

Interestingly, the pruned SNN (i.e., under the lowest threshold) demonstrated better performance comparison to standalone STDP and STDP+IP approaches as per Figure 8.8b. According to the performance summery of the Wrist data, an over-fit of 1-2% is present in standalone STDP, which had grown to an over-fit of 4-8% with STDP+IP across all matrices (refer Table 8.1). Pruning results indicate the countering effect that has elevated generalisation capability by 2-5%.

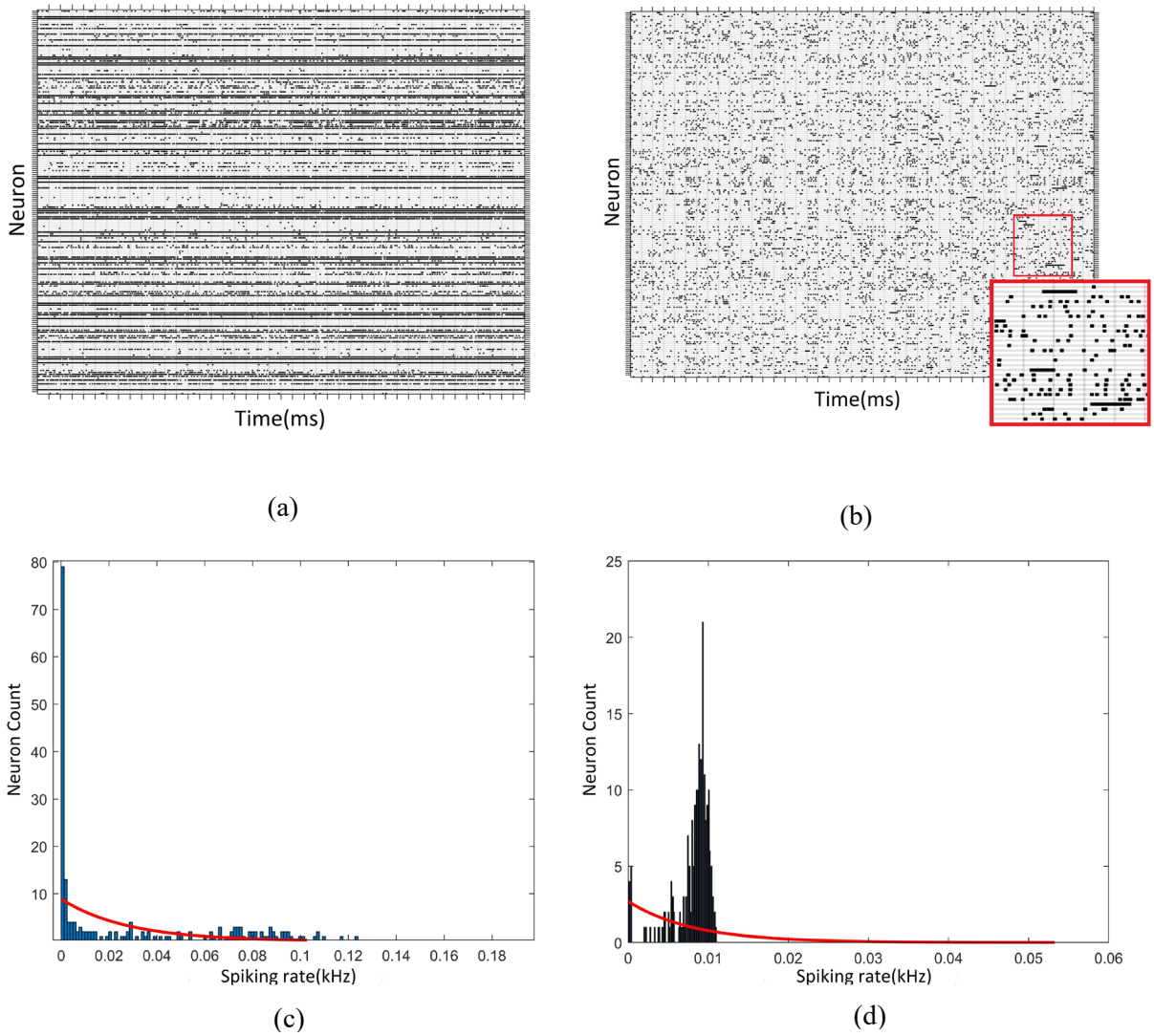


Figure 8.6: Spiking raster plots and firing distributions obtained after propagating the training data over the trained network a) Standalone STDP raster b) STDP+IP raster c) Standalone STDP firing distribution d) STDP+IP firing distribution

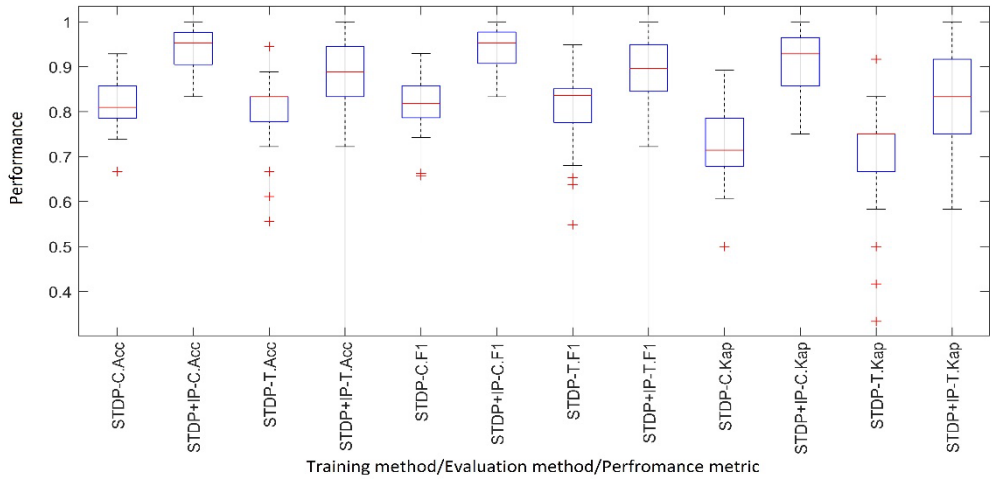


Figure 8.7: Performance comparison of STDP only and STDP+IP training. Cross-validation is denoted by *C* and split testing by *T*. *Acc* for Accuracy, *F1* for F1-Score and *Kap* for Kappa value.

Table 8.1

Average ML performances summarised for Wrist data

Metric	**STDP only**		**STDP + IP**		**Pruned***
	C.V	**Test**	**C.V**	**Test**	**Test**
Accuracy	0.814	0.796	0.936	0.880	0.917
F1-Score	0.818	0.807	0.937	0.889	0.924
Kappa	0.721	0.694	0.904	0.819	0.875

Note. * Testing results after 17% of the original network was pruned following STDP+IP training

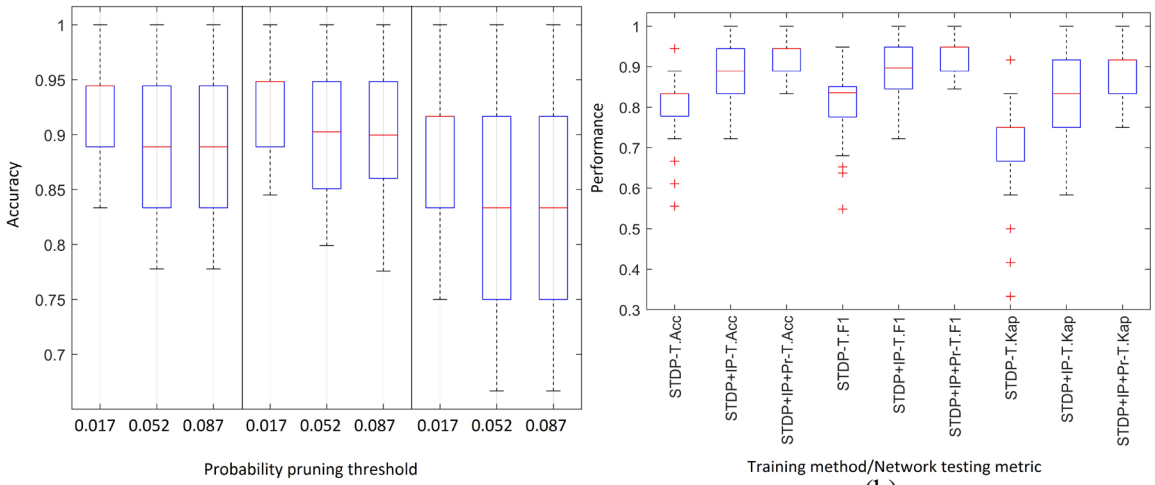


Figure 8.8: a) Pruned network accuracy performance compared under 70/30 train-test split b) Comparison of 70/30 train-split performance after STDP only, STDP+IP and STDP+IP Pruned (denoted as Pr) approaches. Split testing is denoted by T,Acc for Accuracy, F1 for F1-Score and Kap for Kappa value.

8.9.2 DEAP dataset

This section presents the results of performing the same experiments as detailed for the Wrist dataset with the DEAP dataset.

STDP and IP combined learning

With the DEAP dataset, we observed the sparseness of spiking with ensemble learning when compared with standalone STDP. Figure 8.9 illustrates sparse spiking after ensemble learning.

The zoom enlargement in Figure 8.9 shows the sizes of spike bursts observed. Two observations can be stated when these bursts are compared with that of the Wrist data experiment: a) bursts observed are much shorter, b) the diversity of burst sizes are minimal. Moreover, observing the entire raster plot, similar size bursts are aligned vertically, indicating minimum temporal diversity. On the contrary, the network trained with STDP only showed a large group of neurons constantly firing at high rates.

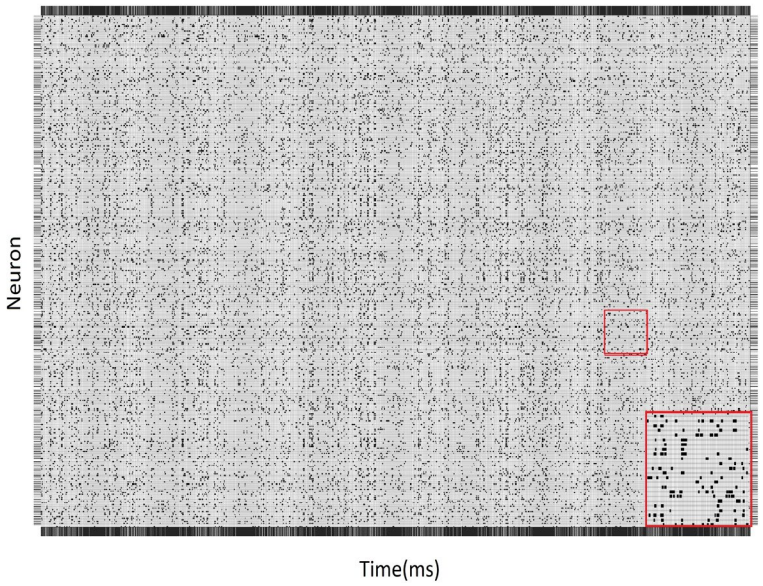


Figure 8.9: Spiking raster plot obtained after propagating the training data over the trained network using STDP+IP

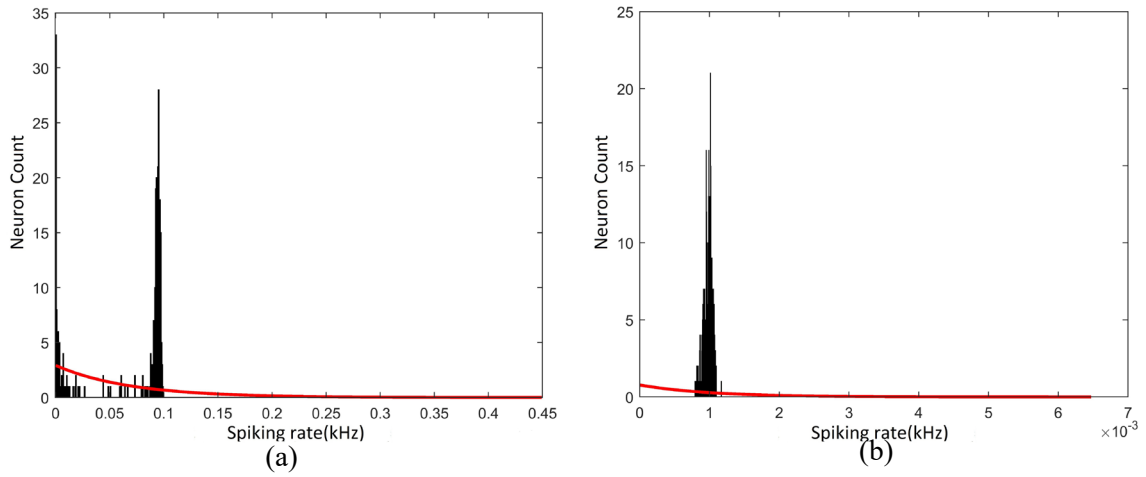


Figure 8.10: a) Spiking distribution under standalone STDP b) Spiking distribution under STDP+IP by T. Acc for Accuracy, F1 for F1-Score and Kap for Kappa value.

The spiking distributions in STDP produced a bimodal distribution where a large number of neurons were concentrated on high or low firing rates, as per Figure 8.10a. After applying IP, this firing rate diversity was further reduced and resulted in a normal spike distribution (see Figure 8.10b). When these spiking distributions are compared with Wrist data distributions, the lack of diversity in spiking rates is evident.

The spiking rates of STDP recorded 0.066 (±0.002, range = 0.009), whereas ensemble learning recorded 0.001(±3.05 × 10^{-7}, range = 1.7 × 10^{-6}) over 30 cycles of random testing conducted.

This translates to better efficiency in terms of information encoding with ensemble learning: which utilises 65 times lesser number of spikes on average for information encoding compared to standalone STDP (*two-sample t − test*, $p < 0.05$).

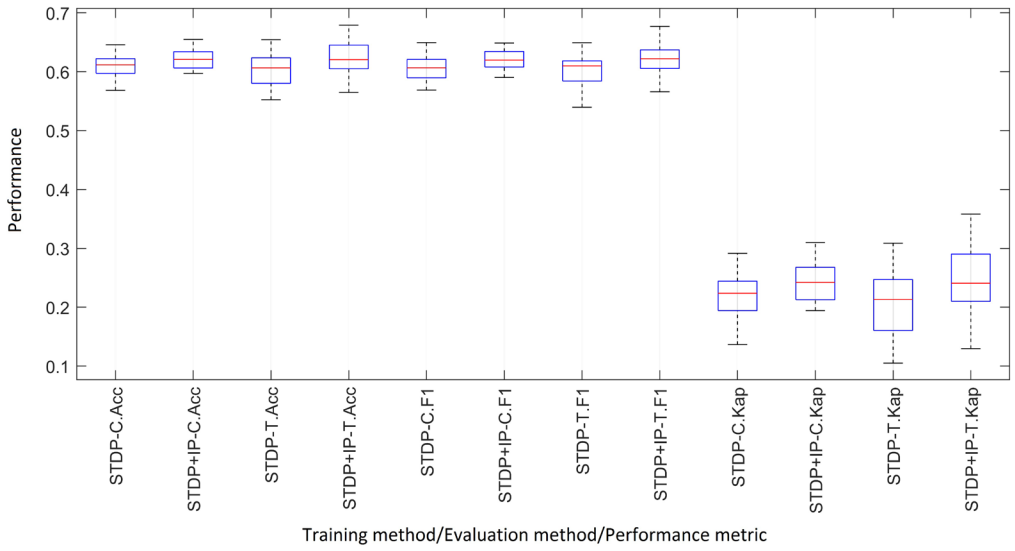


Figure 8.11: Performance comparison of STDP only and STDP+IP training for the DEAP dataset. Cross-validation is denoted by C and split testing

In terms of robustness, STDP+IP showed a 1-3% improvement in all matrices of measurement than standalone STDP. Adaptability was also better with IP, which is indicated by the lower dispersion of performance (Figure 8.11). We did not observe any over-fitting condition with the DEAP data, unlike with the Wrist data, but the overall performance in terms of pattern separation was low.

Network Pruning

We pruned the network trained using ensemble learning under three different pruning spike-rate thresholds of 8.2×10^{-4},8.8×10^{-4}, and 9.3×10^{-4}. These thresholds were assigned based on the three lowest spiking rates observed in the network. The average testing accuracy was recorded at 0.62 (±0.034, range = 0.11), 0.61 (±0.034, range = 0.12) and 0.61 (±0.031, range = 0.16) for 8.2×10^{-4}, 8.8×10^{-4} and 9.3×10^{-4} pruning thresholds respectively. These thresholds pruned the network on average by 26%, 29% and 36%. The initial pruning threshold, 8.2×10^{-4} yielded higher robustness than the standalone STDP, with 26% of the neurons removed. The performance summary under the three approaches is given in Table 8.2.

Table 8.2

Average ML performances summarised for DEAP data

Metric	STDP only		STDP + IP		Pruned*
	C.V	Test	C.V	Test	
Accuracy	0.606	0.601	0.622	0.624	0.617
F1-Score	0.605	0.597	0.617	0.618	0.612
Kappa	0.212	0.202	0.243	0.248	0.234

Note. * Testing results after 26% of the original network was pruned following STDP+IP training

8.10 Discussion

The results presented clearly indicate the superiority of the ensemble learning approach over standalone STDP in terms of robustness, adaptability and processing efficiency (See Figure 8.7 and Figure 8.11). Moreover, the application of neuron pruning after ensemble learning countered over-fitting conditions (i.e., in Wrist data) or gradually decreased inference performance. Interestingly, even if 25% of the network was pruned, the average inference performance remained better than the standalone STDP for both data sets.

After applying ensemble learning, we observed a cluster of low-firing neurons separated from the main firing distribution with Wrist data (see Figure 8.6d). This was not the case with DEAP data (see Figure 8.10b), where the diversity of spiking rates observed was minimum. The pruning of low-firing neurons rectified over-fitting in Wrist data and reduced the performance with DEAP data. In computational biology, Savin and Joshi proposed IP and synaptic scaling to complement STDP to enable single neurons to learn independent components from the input. Accordingly, we propose the low-firing neuron cluster to represent minute features causing over-fitting. Furthermore, the similarity of the input spikes may have caused less diversity in spiking distribution, which resulted in lower accuracy with DEAP data. The novel IP tuning method introduced in this paper uses two parameters to regulate spiking activity, namely, neuron activation and reduced entropy. These parameters are measured using the entire network activity, enabling *low but not too low* levels of spiking (C. Li & Li, 2013) and maintaining firing homeostasis while preventing runaway synaptic potentiation (see sparser firing in Figure 8.6b & Figure 8.9). From an ML perspective, runaway synaptic potentiation decreases efficiency due to the overpopulation of spikes. These findings keep in line with previous studies (Chen et al., 2013; Watt & Desai, 2010), which discussed the ability of IP to counter runaway synaptic potentiation.

Our method of IP adaptation is different compared to SPIKL-IP (W. Zhang & Li, 2019), which uses a singular objective of information maximisation without STDP. According to the information maximisation rule presented by Bell and Sejnowski (Bell & Sejnowski, 1995), maximising output entropy would increase the mutual transfer of information from input to output. They also pointed out that an exponential output firing distribution would correspond to maximum entropy. Interestingly, when comparing the performances(see Figure 8.7 and Figure 8.11) and firing distributions(see Figure 8.6d & Figure 8.10) of Wrist and DEAP data, we can observe the following: if standalone STDP produces an exponential firing distribution (i.e. maximum information transfer)(see Figure 8.6), proposed ensemble method would greatly enhance performance. However, if standalone STDP does not produce an exponential output firing distribution (See DEAP output firing distribution Figure 8.10b), the performance enhancement with ensemble learning is less (13% see Table 8.2). This translates to either STDP learning rates not enabling maximum information transfer or inadequacy of salient information from the input. Nevertheless, in both instances of modelling discussed in this paper, ensemble learning produced normal firing distributions. Therefore, according to our experiments, tuning learning rates to obtain the desired exponential (W. Zhang & Li, 2019) or Weibull (C. Li & Li, 2013) firing output distribution may not be essential when IP is applied with STDP in a multi-spike environment for better ML performance.

The method of network adaptability discussed here differs from previous studies (Rathi et al., 2019; Shi et al., 2019) due to the use of ensemble plasticity and spike-rate neuron pruning used instead of synaptic pruning. Since the neuron spiking is more representative of the temporal features, this method can amalgamate a soft-pruning mechanism (Shi et al., 2019)without removing the neuron completely, which can be beneficial if the robustness of the network is lower as per the instance with the DEAP data. Furthermore, since this adaptive mechanism is based on unsupervised learning, it can be used as a Neural Architecture Search (NAS) (Mellor et al., 2020) before complete training or hyperparameter optimisation. The combined method of ensemble learning and pruning can also be converted to online learning since IP learning rates and pruning thresholds are adjusted primarily based on spiking activity rather than the final inference result.

An important spiking behaviour was observed, which appears to push neurons to operate at a critical stage or “edge of chaos” (Bertschinger & Natschläger, 2004), which helps achieve higher performance and computational efficiency (X. Li, Chen, et al., 2017). These bursts of spikes that we observed (see Figure 8.6b) are often referred to as avalanches. High performance in the presence of avalanches is clearly evident when the spiking behaviour of ensemble learning is compared between Wrist data and DEAP data. This performance enhancement ability of avalanches has been proposed previously in

biological modelling (X. Li et al., 2018). This is an important area that needs further investigations that may open windows for ASNN to perform better since the information representation capability can be further enhanced beyond spike time, rate or rank.

8.11 Conclusion

In this work, we present an unsupervised ensemble learning method (by combining STDP and IP) with a spike regulation strategy developed based on information theory. This method is then amalgamated with a biologically inspired network pruning mechanism. We have evaluated these methods on a rigorous ML framework and attempted to interpret results at the fundamental level of spikes. The spike-based ensemble learning and pruning methods discussed in this paper can enhance the robustness and increase efficiency, particularly in multi-spike learning tasks. These methods can be adapted for batch learning, online learning or architectural searches before extensive training and hyper-parameter optimisations.

In terms of limitations, the data sets we have used in this study are balanced, which restricts us from discussing their performance on unbalanced data. Moreover, we used a common spike encoding mechanism for both tasks; this is a crucial point in modelling with obvious implications for ML performance. Apart from addressing the said limitations, we are motivated to a) explore further on ASNNs operating in *edge of chaos*, guided by ensemble learning and apoptosis, b) develop online ASNN algorithms using methods discussed, c) implement participant-specific ASNN models for neuropsychological hypothesis generation, as a part of our future work.

8.12 Appendix

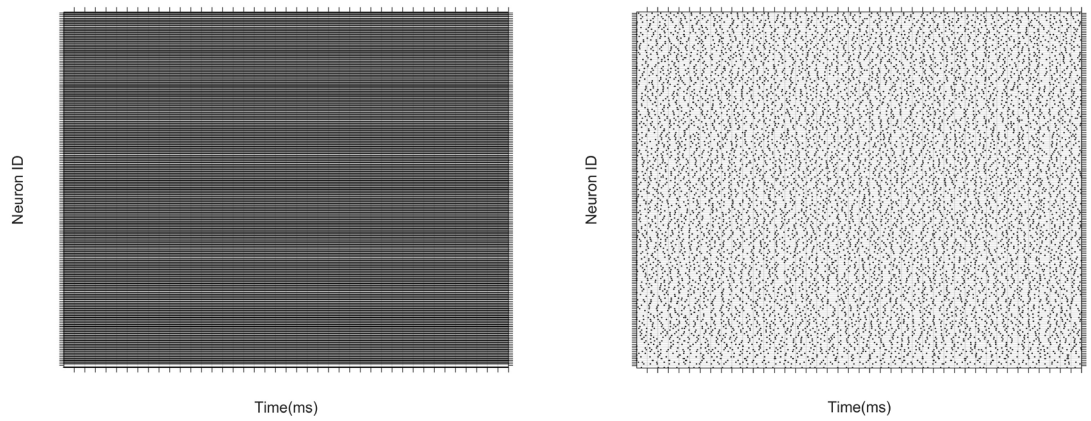


Figure 8.12: Spiking raster plots under different *θpos* and *θneg* conditions a) *θpos* <*θneg* b) *θpos* = *θneg*

Algorithm 1: AER encoding algorithm for spike conversion

function AER ($x_{(t)}$, f);
  Input : EEG channel data $x_{(t)}$ and factor f
  Output: Spike train $s_{(t)}$
for t = 1: $length(x_{(t)})$ do
  $tempdif\ f_{(t)} = x_{(t+1)} - x_{(t)}$
  return $tempdif\ f_{(t)}$;
  end
$tempdif\ f_{(end)} = tempdif\ f_{(end-1)}$
$threshold = mean\ (tempdif\ f_{(t)}) + (f * std\ (tempdif\ f_{(t)})$
for t = 1 : $length(x_{(t)})$ do
  if $tempdif\ f_{(t)} > threshold$ then
    $s_{(t)} = 1$
  else
    if $tempdif\ f_{(t)} < threshold$ then
      $s_{(t)} = -1$
    end
  end
  return $s_{(t)}$;
  end

Figure 8.13: AER encoding algorithm

Table 8.3

Hyper-parameter values used for experimenting

Processing step	Hyper-parameter	Value
AER Encoder	*factor*	0.5
LIF	v_{init}	0.05
	v_{rest}	0
	$t_{refractory}$	5
	R	1
	C	10
	η_{init}[a]	200
	η_{init}[b]	300
STDP	A_+	0.001
	A_-	0.001
	τpos	10
	τneg	10
	w_{max}	+0.1
	w_{min}	-0.1
IP	θpos[a]	1×10^{-3}
	θneg[a]	1×10^{-5}
	θpos[b]	7×10^{-5}
	θneg[b]	7×10^{-8}
Classifier	α	1
	mod	0.8
	drift	0.001

[a] For Wrist Flexion data set modelling [b] For DEAP data modelling

Chapter 9 Mental Stress Recognition on the fly using Neuroplasticity Spiking Neural Networks

This chapter is comprised of a journal article accepted by *Nature Scientific Reports.*

9.1 Reference:

Mahima, M., Weerasinghe, A., Wang, G., Whalley, J., Crook-Rumsey, M., & Whalley, J. (2022). *Mental Stress Recognition on the y using Neuroplasticity Spiking Neural Networks*. 0–32. https://doi.org/https://doi.org/10.21203/rs.3.rs-1841009/v1

9.2 Author contribution:

M.W.: 84%, G.W.:8.0%, J.W.:7.0%, M.C.R.: 1%

M.W. designed the algorithm, ran the experiments, analysed the data, and wrote the manuscript. G.W planned and executed EEG data collection with M.C.R. J.W. and G.W. supervised the research. All authors reviewed the manuscript and provided feedback.

9.3 Preamble

In Chapter 7, a repertoire of neuroplasticity methods found in the brain was explored. The SNN model presented in Chapter 8 used these methods (STDP, IP and data-driven pruning options) for batch mode learning. However, one of the key requirements for temporal data is to learn and infer online, and the model presented in Chapter 8 is not capable of online learning. Moreover, an online learning approach is appropriate for personalised mental health monitoring as it responds to new information as it is received. In this article, the idea of using online ANNs and EEG data for online learning for personalised mental health monitoring is explored.

One of the biggest challenges for online learning approaches is managing the dynamic change of output representations known as *concept drift.* Concept drift is predominantly present in spatiotemporal data collected from biological systems, and successful techniques for the handling of this phenomenon require forgetting obsolete knowledge without hindering the acquisition of new knowledge into the system.

In recent studies of SNNs, researchers have used selective neuron pruning and new neuron addition to forget obsolete knowledge and represent new knowledge, respectively. By further developing the brain-inspired ensemble learning method of STDP, IP, and pruning presented in Chapter 8, the following work introduces a new SNN algorithm that can learn from data on the fly (online). This SNN prunes neuron that represents obsolete features and adds new neurons if new knowledge is found. The experiments presented in this article test the model's ability to classify brain states in terms of stress, neutral, and positive. The EEG dataset used in this study was collected by providing auditory stimuli to participants, and each participant's perceived stress (PS) levels were collected using a standard questionnaire. This study is then extended to explore possible connections between acute and perceived stress using network behaviour and PS scores.

9.4 Abstract

Mental stress is found to be strongly connected with human cognition and well-being. As the complexities of human life increase, the effects of mental stress have impacted human health and cognitive performance across the globe. This highlights the need for effective non-invasive stress detection methods. In this work, we introduce a novel, artificial spiking neural network model called Online Neuroplasticity Spiking Neural Network (O-NSNN) that utilises a repertoire of learning concepts inspired by the brain to classify mental stress using Electroencephalogram (EEG) data. These models are personalised and tested on EEG data recorded during sessions in which participants listen to different types of narratives designed to induce acute stress. Our O-NSNN models learn on the fly producing an average accuracy of 90.76% (σ = 2.09) when classifying EEG signals of brain states associated with these audio comments. The brain-inspired nature of the individual models makes them robust and efficient and has the potential to be integrated into wearable technology. Furthermore, this article presents an exploratory analysis of trained O-NSNNs to discover links between perceived and acute mental stress. The O-NSNN algorithm proved to be better methods without structural plasticity for personalised stress recognition in terms of accuracy, efficiency, and model interpretability.

9.5 Introduction

9.5.1 Implications of mental stress

We often encounter stress in daily life with variations of intensity and prolongation. Stress is understood as the response of the human body to mental and physical stimuli that involves the nervous system and hypothalamus-pituitary-adrenocortical axis (Selye, 1965). According to the literature, stress is often classified as acute, episodic, or chronic (Epel et al., 2018). Many contemporary studies have found stress to have a major impact on human health and cognitive performance. In some cases, stress has been shown to have a direct connection to depression, anxiety, stroke, cardiovascular disease, cancer, speech, and cognition impairment (Crowley et al., 2011; O'Connor et al., 2021; Wu et al., 2019). The negative effects of stress on human cognition are associated with dysfunctional changes in the prefrontal cortex and amygdala activation (Arnsten, 2009; Wu et al., 2019), whereas the physical health effects of stress are related to detrimental changes in immunity and physical homeostasis (Lawrence, 2000). As the complexities of human life increase, the effects of stress have begun to burden nations and the globe at large (Epel et al., 2018; Seo & Lee, 2010), which highlights the requirement for more research in this area. Early detection of harmful stress can be crucial as a part of effective stress management to promote greater well-being.

9.5.2 Stress and electroencephalogram

Rapid development in sensor technologies and machine learning (ML) techniques have enabled research communities to begin to develop automated stress detection systems. These systems use invasive and/or non-invasive data acquisition methods. Stress recognition using invasive methods can be highly time-consuming and often require experts for data acquisition and processing (Jin, 1992; Lerner et al., 2007; Lundberg et al., 1994); this is not ideal for an automated system. The most common non-invasive methods include Electroencephalogram (EEG), heart rate variability, galvanic skin response, blood volume pulse, and electromyography for data acquisition. Of these non-invasive methods, EEG is used most extensively for stress recognition due to its: information richness, cost-effectiveness, and high temporal resolution (Saeed et al., 2020).

9.5.3 Stress recognition on the fly

Current methods for stress recognition use traditional ML techniques such as Linear Discriminant Analysis (Y. Zhang et al., 2020), Naive-Bayes (Subhani et al., 2017), Support Vector Machine (Betti et al., 2018), K-Nearest Neighbor (Khosrowabadi et al., 2011), and Multi-layer Perceptron (Saidatul et al., 2011). However, these methods are not capable of evolving and adapting to new information after

training, preventing them from being used in an online setup (Domingos & Hulten, 2003). Online learning typically uses real-world data that changes with time. Thus, the model is adaptive and learns as new data is fed into it over time. In contrast, most stress detection approaches presented in the literature use static data to train and test the model. They also typically employ interventions, to manipulate the data used to train and test the models, such as feature engineering methods. It is difficult to compare the performance of known stress detection models because the feature engineering and extraction approaches differ from one study to another. This lack of standards also means that the generalisability of the methods presented is questionable (Katmah et al., 2021). Moreover, these traditional methods require a high volume of labelled data for model training. Today, the emergence of wearable technologies has revealed the potential for personalised health applications, designed to detect stress. Such applications must meet certain conditions to be practical. Use of online learning allowing the model to adapt to change, capability to operate under low power and the need for low-resource utilisation are among them. This work focuses on finding solutions for the challenges posed by these conditions.

9.5.4 Data drifts and online learning

One of the challenges in online learning is handling what is known as the drift phenomena successfully. Drifts can be observed in spatiotemporal data such as EEG, and they can be defined in terms of input(s) and concept(s) (Lobo et al., 2018). The input(s) drift refers to the change of input data distribution over time without affecting the posterior probabilities of classes; concept drift refers to the change of posterior probabilities of the classes over time without any changes in the input distribution (Lobo et al., 2020a). The drift phenomena require ML techniques to be able to acquire new knowledge without forgetting the prior knowledge (i.e., avoiding catastrophic forgetting) and even to update prior knowledge based on that new or recently gained knowledge. Adding to the challenge are the restrictions posed by online learning, which demands the algorithm to use only a limited amount of pre-allocated memory, process a sample only once, use a consistent amount of time for processing, produce a valid model at each processing step, and perform in par with batch mode learning (Domingos & Hulten, 2003).

9.5.5 Spiking Neural Networks (SNNs)

SNNs are a class of artificial neural networks (ANNs) that are considered to be biologically plausible (Maass, 1997). They have proven to be highly efficient in terms of time and memory requirements for data processing compared to commonly used sigmoidal counterparts (Maass, 1997). The temporal dimension used in data processing is a major factor that contributes to their increased efficiency when

compared with traditional ANNs, which makes SNNs an ideal candidate for online learning (Zuo et al., 2017). Moreover, the unsupervised learning mechanisms in SNNs have demonstrated the capability for fast and data-efficient learning (Kheradpisheh et al., 2018; Masquelier & Thorpe, 2007; Panda & Roy, 2016). These attributes have led to the development of several online learning algorithms using SNNs with both supervised and unsupervised learning (Bohte et al., 2002; Dora et al., 2016; Florian, 2012; Gütig & Sompolinsky, 2006; Legenstein et al., 2005; Lobo et al., 2018; Pardey et al., 1996; Schliebs & Kasabov, 2013; J. Wang et al., 2014; M. M. A. Weerasinghe et al., 2021a; Wysoski et al., 2008). Of these methods, only a few algorithms use structural adaptation (i.e., evolving and pruning neurons and connections). Structural adaptation is crucial for learning new knowledge and forgetting irrelevant information in an online setup (Dora et al., 2016; Lobo et al., 2018; Schliebs & Kasabov, 2013; J. Wang et al., 2014; M. M. A. Weerasinghe et al., 2021a). However, some of these structurally adaptive methods are built for batch mode learning only (Schliebs & Kasabov, 2013; M. M. A. Weerasinghe et al., 2021a) or do not fully exploit the temporal dynamics through learning (Dora et al., 2016; Lobo et al., 2018; J. Wang et al., 2014).

The Online Neuroplasticity Spiking Neural Network (O-NSNN)

The O-NSNN introduced in this work uses mathematical abstractions of selected plasticity techniques found in brain functions to fully exploit spatiotemporal patterns present in the data. This does not mean that the model mimics the entire neurobiological process of the brain, but rather it uses selected concepts of signal encoding, propagating, processing, and learning found in the brain. This algorithm differs from the previous ASNNs (Dora et al., 2016; Lobo et al., 2018; Schliebs & Kasabov, 2013; J. Wang et al., 2014; M. M. A. Weerasinghe et al., 2021a) due to the inclusion of a full repertoire of plasticity techniques for temporal learning. These techniques are Spike Time Dependent Plasticity (STDP) (Bi & Poo, 1998), Intrinsic Plasticity (IP) (Desai et al., 1999), Neuron Evolving (neuron addition) (Aimone, 2016) and Neuron Pruning (neuron elimination) (Shi et al., 2019). We hypothesise that this algorithm will produce stable and faster pattern separation capability in the online classification of stress-related EEG by considering and handling the challenges associated with online learning.

The proposed O-NSNN consists of three layers of Leaky-Integrate and Fire neurons (LIF) (Gerstner & Kistler, 2002) (see Figure 9.1); a mathematical abstraction of a biological neuron that has demonstrated a greater balance between biological plausibility and computational tractability(E. M. Izhikevich, 2004). Before processing, the EEG signals are converted to their spiking equivalent using Address Event Representation (AER), a spike encoding algorithm used in the artificial retina(Delbruck & Lichtsteiner, 2007). Thereafter, the first layer of neurons propagates spikes to the second layer via excitatory and inhibitory synapses. During this propagation, the synaptic weights are updated using the

STDP rule (Bi & Poo, 1998). In addition, all the neurons adjust their excitability using an IP rule(M. M. A. Weerasinghe et al., 2022). This combination of unsupervised STDP and IP prevents the network from getting caught up in a potentiation loop (Chen et al., 2013) ensuring homeostasis (Diehl & Cook, 2015) and helping neurons extract independent spiking features from the input (Savin et al., 2010). Moreover, the second layer of neurons undergoes a self-pruning process induced by error monitoring to avoid misclassifications caused by low-spiking neurons (M. M. A. Weerasinghe et al., 2022). The synapses from the second layer to the third layer are excitatory and follow a similar weight updating strategy discussed in dynamically evolving SNN (deSNN) (N. Kasabov et al., 2013), evolving a new neuron in the presence of new knowledge. However, unlike in deSNN, output neurons are not merged based on weight vector similarities (i.e., calculated using Euclidean distance of the input weight vector of a given neuron). In the presence of data drift, neurons of similar Euclidean distances may represent different classes. Therefore, we do not merge neurons, rather, we eliminate or preserve neurons created based on the classification errors made during the data processing (Please refer to the Methods section for an in-depth explanation). This combined process of neuron addition in the third layer and neuron pruning in the second layer are unique implementations that have not been discussed together in the published literature, to the best of our knowledge.

9.5.6 Acute stress and data collection

The dataset used in this study consists of EEG recordings from 22 healthy participants (twelve males - average age = 27.92 years, standard deviation (σ) = 3.09 and ten females - the average age of 25.9 years, σ = 8.20) across three different conditions. On each condition, the participants were asked to listen to one type of comment, either critical, neutral, or positive. Such critical comments stimulate the part of the human auditory system of which the primary objective is to alert and warn (Westman & Walters, 1981). Moreover, audio criticism has also been shown to induce mental stress levels in previous studies (Daly et al., 2019; Khosrowabadi, 2018a; Markova & Ganchev, 2018) and music to induce positive and negative emotions (Wegge et al., 2007). Based on these previous studies, we presumed that the critical comments would induce acute stress in the participant. The details of these comments used for this study have been validated and published previously (Premkumar et al., 2019; G. Y. Wang et al., 2021). In addition to EEG data, the perceived stress of each participant was recorded using the PSS-14 scale (Cohen et al., 1983). Each EEG recording lasted for two minutes, and the recordings were segmented into five-second splits to feed the O-NSNN. Consequently, a single sample of EEG data consisted of 1280 time points and four channels. From each participant, 72 such samples, with 24 samples for each class of stressed, neutral, and positive, were processed. Complete details of the dataset are given in the methods section.

EEG channels and evaluation

For the experiments of this study, we extracted signals from the FP1, FP2, T7, and T8 channels. In a previous study, researchers showed the sufficiency of two frontal channels for stress vs non-stress classification (Attallah, 2020). Furthermore, since the stimuli were auditory, T7 and T8 channels were used to capture the dynamics of the auditory cortex. Classification accuracy and sensitivity (true positive rate for stress EEG) was used to measure the performance. These measures using O-NSNN, was compared against 70/30 split batch learning and online learning without structural plasticity (SP). For this, we used individualised O-NSNN models since the effects of stress are found to be depending on an individual's neurobiological predisposition (Epel et al., 2018). Moreover, we used the prequential accuracy metric to evaluate the performance of online learning (Dawid & Vovk, 1999). Secondly, these individualised models were subjected to an exploratory analysis that was undertaken to test the interpretability of the model and see if relationships could be discovered between acute and participant-perceived stress.

This exploratory analysis involved comparing the personalised network activations to individually reported perceived mental stress levels. We categorised participants into one of three classes based on their PSS-14 scores (see Table 9.1). The connection weights of personalised models and Euclidean Distances (ED) of third-layer neurons were analysed to find patterns within and between the perceived mental stress groups.

In this work, we present a spatiotemporal data processing method for mental stress recognition and elucidate the possibility of investigating brain activity at an individual level. Therefore, the contribution of this study benefits both computer science and psychology/neuroscience research communities. The contributions of the study are as follows:

1. O-NSNN algorithm equipped with a biologically plausible repertoire of plasticity techniques for online mental stress recognition.
2. Insights into how perceived stress relates to incidences of acute stress.

Table 9.1

Participant categorisation based on Perceived Mental Stress scores calculated from PSS-14 questionnaire.

Label	PSS-14 score	Number of participants
High Stress (HS)	PSS > 30	6
Medium Stress (MS)	20 < PSS ≤ 29	11
Low Stress (LS)	0 < PSS ≤ 19	5

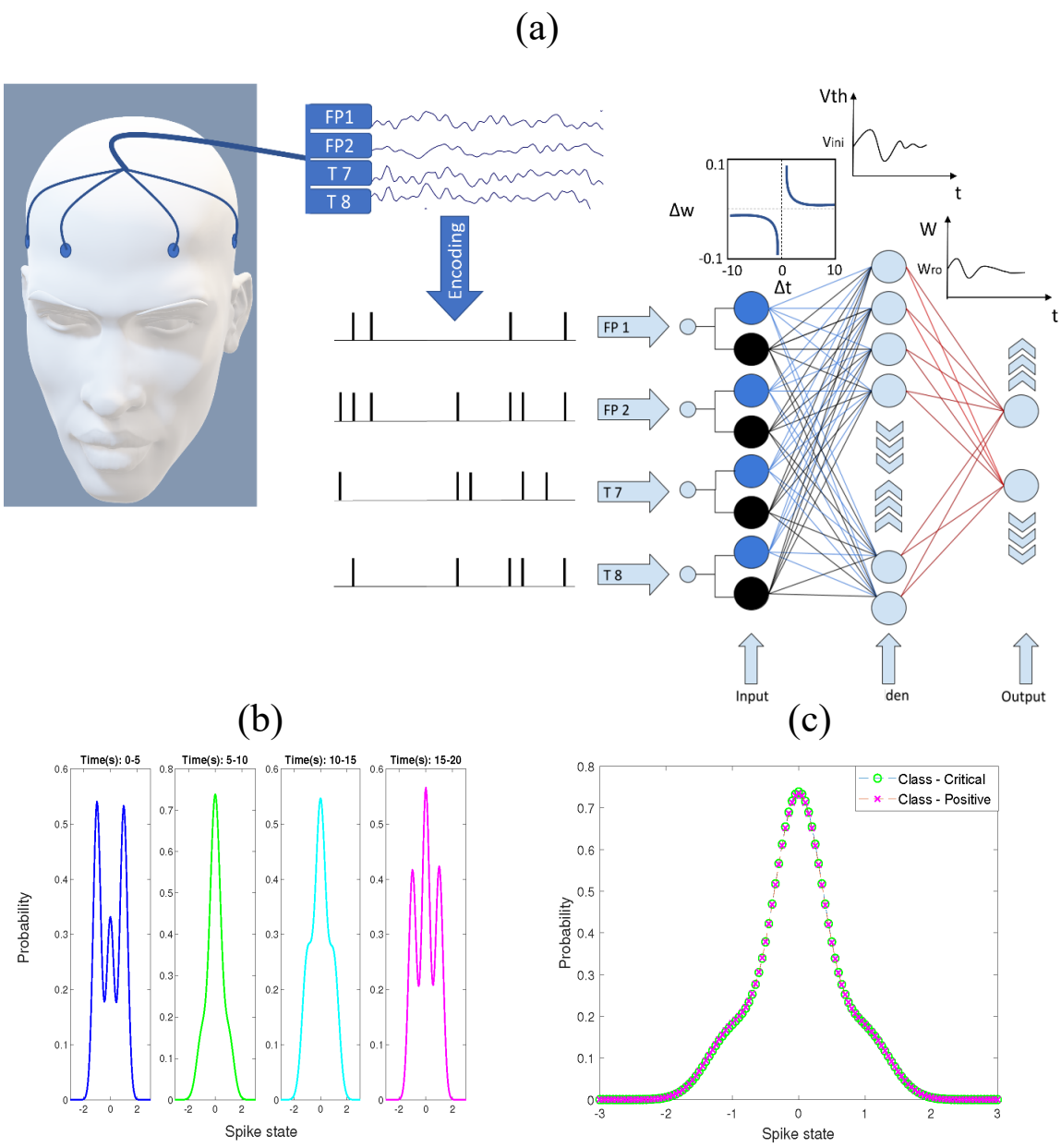


Figure 9.1: (a) The proposed O-NSNN architecture for stress recognition. EEG originating from FP1, FP2, T7 and T8 channels are encoded into spikes (using the AER algorithm) and propogated through a three-layered SNN architecture. An STDP rule is used for temporal learning between the input layer and the hidden layer. Hidden layer neurons use IP to adapt excitability based on the incoming data. The output layer learns using RO and SDSP rules. Each hidden layer neuron prunes itself according to soft-pruning rule and, the output layer evolves. (b) Stress class input samples of P1 with different spike rate distribution (Input drift) (c) Two separate classes of P1 (Critical and Positive) with the same input spiking distributions (Concept drift).

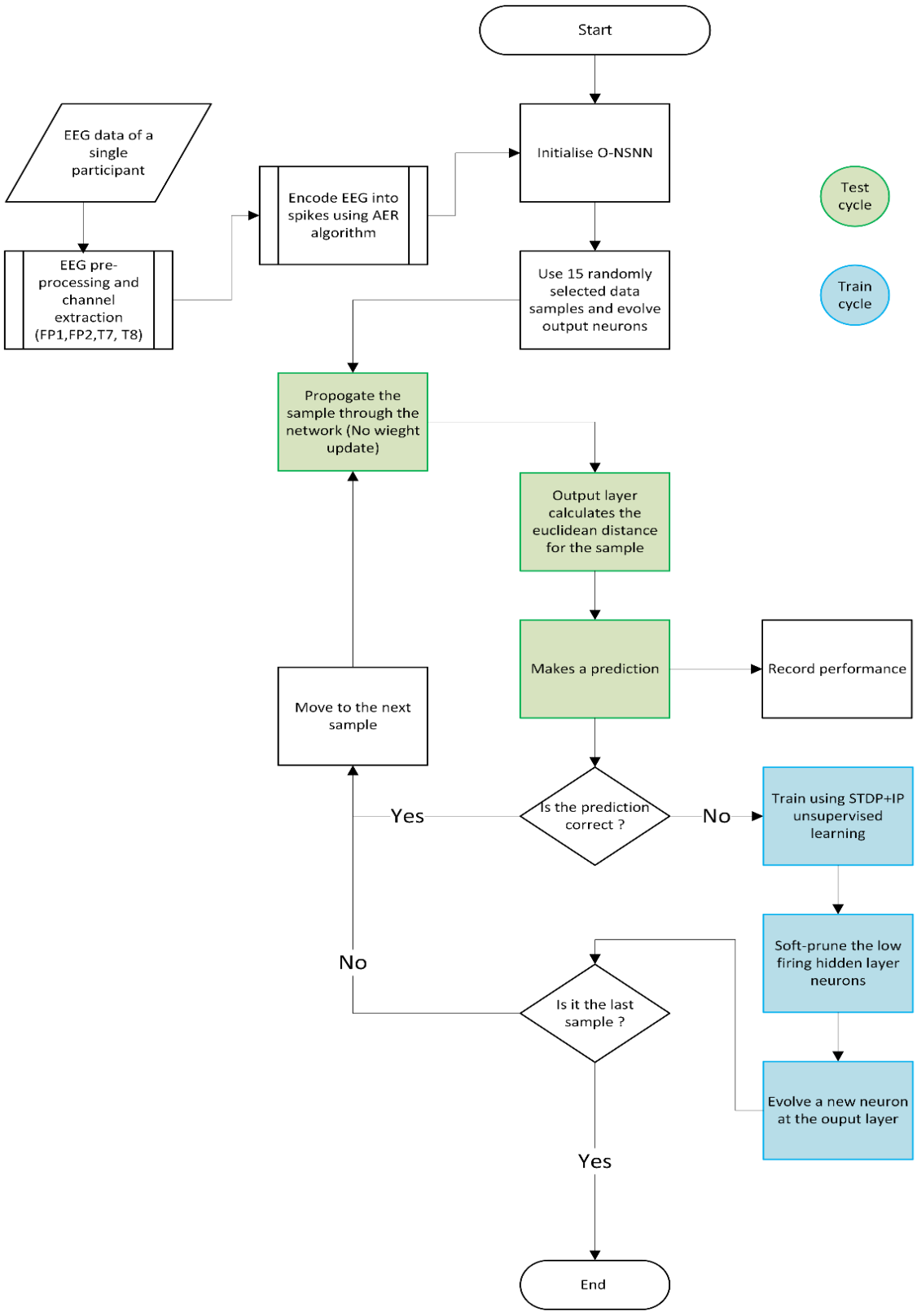


Figure 9.2: Flow diagram of the experiment. The experiment is conducted according to test-then-train regime (Lobo et al., 2018). Under this regime, the network is only trained when a prediction is incorrect.

9.6 Methods

9.6.1 Neuroplasticity spiking neural network

Here we present a description of the O-NSSN model and the experimental framework designed to test the model. The NSNN is a fully connected, feed-forward spiking neural network consisting of LIF neurons(Gerstner & Kistler, 2002). The input nodes can process both excitatory and inhibitory spikes. These nodes are connected to the hidden layer via excitatory and inhibitory synapses in which the weights are updated using an unsupervised STDP learning algorithm(Bi & Poo, 1998). The hidden layer neurons operate in an adaptive threshold scheme in an unsupervised manner using an IP learning rule(M. M. A. Weerasinghe et al., 2022). The hidden layer is connected to the output layer via excitatory synapses updated according to Spike Driven Synaptic Plasticity (Fusi et al., 2000) and initiated using the Rank Order (RO) rule(S. Thorpe & Gautrais, 1998). The hidden layer neurons undergo a self-pruning mechanism. The third layer acts as the classifier and can evolve new neurons. All the hyperparameter values of the NSNN introduced are given in Table 9.2.

Spike encoding using address event representation

AER is a biologically inspired spike encoding mechanism used in artificial retina applications (Delbruck & Lichtsteiner, 2007). Its simplicity, efficiency, and adaptiveness make it an attractive option for online applications. The temporal difference $tempdiff_{(t)}$ (refer Equation (38)), between two temporarily contiguous data points (denoted x_t and $x_{(t-1)}$) and, a user-defined threshold factor f is used to calculate an adaptive spike threshold at each time step (refer to Equation (39)). If the EEG voltage value of the current time step is more than the threshold, an excitatory spike is emitted otherwise, an inhibitory spike is emitted.

$$tempdiff_{(t)} = x_t - x_{(t-1)} \quad \textbf{(38)}$$

$$threshold = mean\left(tempdiff_{(t)}\right) + \left(f * std(tempdiff_{(t)})\right) \quad \textbf{(39)}$$

Leaky Integrate and Fire Neuron

The LIF neuron is commonly used in machine learning applications due to its computational tractability and the ability to produce basic spike behaviours (E. M. Izhikevich, 2004). Since this study involves an IP (adaptive voltage threshold) method, a wider variety of spiking behaviours can be produced than can

be produced by a normal LIF (E. M. Izhikevich, 2004). The membrane potential change $\frac{dv_t}{dt}$ of a LIF neuron can be modelled using a resistor-capacitor circuit and mathematically expressed using equation (40). In the equation, the time constant τ_m is equal to the product of resistance R and capacitance C. The membrane potential is given by v_t and, the input current at time t is given by I_t. The resting voltage of the neuron is given by v_{rest}.

$$\tau_m \frac{dv_t}{dt} = v_{rest} - v_t + RI_t\,, \qquad \tau_m = RC \tag{40}$$

Unsupervised learning

In the O-NSSN, the unsupervised weight update strategy STDP (Bi & Poo, 1998) is accompanied by an IP rule (M. M. A. Weerasinghe et al., 2022) that adapts the threshold of hidden layer neurons individually. This combination of plasticity is a key factor in maintaining firing homeostasis and enhancing SNN performance in terms of classification accuracy and efficiency (Diehl & Cook, 2015; Hao et al., 2020; M. M. A. Weerasinghe et al., 2022).

$$F(\Delta t) = A_+ \exp^{(-\Delta t/\tau_{pos})} \ \Delta t > 0 \tag{41}$$

$$F(\Delta t) = -A_- \exp^{(\Delta t/\tau_{neg})} \ \Delta t < 0 \tag{42}$$

$$\Delta w_{ij} = \sum_{a}^{b} \sum_{p}^{q} F(t_j^m - t_i^n) \tag{43}$$

Equations (41) and (42) represent STDP according to Long-Term Potentiation (LTP) and Long-Term Depreciation (LTD), respectively (Bi & Poo, 1998). Both equations are functions of the time difference Δt between spikes. In equation (43) the pre-synaptic neuron is denoted by i and the post-synaptic by j. If j fires before i, Δt is positive, leading to LTP. A reversed firing sequence leads to LTD. In equations (41) and (42), the positive and negative time constants are given by τ_{pos} and τ_{neg} respectively. These time constants are predetermined windows of time used for synaptic modifications.

A_+ and A_- terms determine the maximum synaptic modification. The cumulative weight change ΔW_{ij} is calculated using the spike timing of each pre-synaptic neuron from p to q and each post-synaptic neuron spiking from a to b. The instantaneous spike time of each post-synaptic neuron is given by t_j^m and each pre-synaptic neuron by t_i^n.

The IP rule operates simultaneously with STDP according to the two equations defined in (44). Here, the first expression of equation (44) is used to upregulate the neuron voltage thresholds and the second to down-regulate.

$$v_{thr}(t) = \begin{cases} v_{thr}(t-1) + N\theta_{pos}v_{init}, & s(t-1) = 1 \\ v_{thr}(t-1) - N\theta_{neg}v_{init}, & otherwise \end{cases} \quad \textbf{(44)}$$

The threshold voltage of a neuron at time t is given by $v_{thr}(t)$. If the neuron fired in the previous time step and satisfied the condition $s(t-1) = 1$, then a fraction of the initial voltage v_{init} is added to the threshold voltage of the previous time step $v_{thr}(t-1)$. This fraction is calculated using the product of the positive learning rate θ_{pos} and the number of neurons in the hidden layer N. If a spike did not occur in the previous time step, then the threshold voltage is lowered using the negative learning rate θ_{neg}. The two learning rates are determined based on the highest neuron activation and lowest information entropy (M. M. A. Weerasinghe et al., 2022) after each sample propagation.

Structural plasticity

The addition of new neurons in the output layer and self-pruning of the hidden layer are the two key SP techniques incorporated in the NSNN algorithm. There are no neurons in the output layer at first. During the initiation process, a predefined number of neurons are evolved. The number of samples used to evolve these initial neurons was 15 for the NSNN in this study. This set of neurons remains in the network and gets their weights updated at each sample pass. Since the NSNN operates under the test-then-train regime, if an error is made during the test phase, a new neuron is evolved in the following training phase. Here, an error symbolises the emergence of a new class or a representational change in an already-known class caused by concept drift (Minku & Yao, 2012). Moreover, self-pruning also takes place in the hidden layer if an error is identified in the previous time step. This self-pruning is executed on neurons with low spiking probability since they can cause poor generalisation (M. M. A. Weerasinghe et al., 2022).

$$W_{jk(init)} = \alpha \, . mod^{order(j,k)} \quad (45)$$

$$W_{jk}(t) = W_{jk(init)} + \sum_{t=1}^{n} d \quad (46)$$

The synaptic weights from the hidden to the output layer are initiated according to the RO rule given in equation (45). The initial weight between j pre-synaptic neuron and k post-synaptic neuron $W_{jk(init)}$, is determined using a learning parameter α and an exponent of mod. The modulation factor mod is predetermined and takes a value between 0 and 1. For the first spike to arrive at the synapse, $order(j,k)$ starts at 0 thereby allocating the highest possible weight and decreases in decimals as the other spikes arrive at the neuron (i.e., decreases $W_{jk(init)}$). Thereafter, a drift parameter d is used to increase or decrease the initial weight to form a weight value at time t, $W_{jk}(t)$.

9.6.2 Performance evaluation

To evaluate the performance in online learning, we used the prequential accuracy metric (Minku & Yao, 2012) with the test-then-train approach (Lobo et al., 2020a). In test-then-train, a sample is tested first before training. This method minimises the memory cost since samples need not be held in memory. By applying prequential memory with this approach, accuracy can be updated incrementally. The accuracies for online learning stated in the study are the final accuracy performance after 360 seconds or 72 samples.

$$ACC_{pre}(t) = \begin{cases} Acc_{pre}(t), & if\ t = t_{init} \\ Acc_{pre}(t-1) + \dfrac{Acc_{pre}(t) - Acc_{pre}(t-1)}{t - t_{init} + 1}, & else \end{cases} \quad (47)$$

In equation (47), the classification accuracy of the NSNN at time t is given by $ACC_{pre}(t)$. Here, t_{init} represents the initial time point which is taken as the reference time point. For the batch learning experiments (i.e., B-RSNN), we used the standard accuracy metric, which is defined as the ratio of the number of correct predictions over the total number of predictions (Urbanowicz & Moore, 2015).

9.6.3 Ethics approval and consent to participate

All experiments were performed in accordance with the relevant guidelines and regulations. The Auckland University of Technology Ethics Committee (AUTEC) provided approval for the study on 2nd October 2019 (Approval identity number: 19/231). All participants were provided with a detailed written consent form explained verbally, detailing the objectives, activities and consequences related to the study. Thereafter, written consent was obtained from all participants individually.

9.6.4 EEG Data

The participant group consisted of 12 males with an average age of 27.92 (σ = 3.09) and 10 females with an average age of 25.9 (σ = 8.20). The EEG data were recorded over three sessions in a sound-attenuated room with a gap of at least one day between each session to prevent carry-over effects. Each participant attended three sessions. At each session, the participant followed a sequence of steps: starting with completing the PSS-14 survey, recording two minutes of resting EEG, recording EEG while listening to an audio of either a critical, neutral or negative comment, followed by recording two minutes of resting EEG. The selection of comments (i.e., Critical, Neutral, or Negative) for a given session was selected pseudo-randomly using a computer algorithm. Each comment lasted from 10 to 15 seconds and 40 such comments were made to listen through earphones during each session. It was presumed that critical comments would induce stress based on the result of previous studies51–53. However, it is noted that all participants may not be stressed to the same level by critical audio comments in an experimental setup. Therefore, the sensitivity to each comment was assessed using measurements of arousal and relevance on an 11-point Likert scale.

The 120 auditory comments used for the study were recordings of male and female native English speakers specifically trained to emphasize critical, neutral and positive comments through pitch and tone (Premkumar et al., 2019; G. Y. Wang et al., 2021). The critical and positive comments were typical remarks that one would hear from a close family member, and the neutral comments were factual statements that had no relevance to the participant. Samples of such comments include, “you are lazy and never finish anything you start! you’ve had chances but didn’t go through with it” (Critical comment); “you are good at organising things and paying attention to detail.” (Positive comment); “the Emu is the largest native bird in Australia, with long neck and legs” (Neutral comment). Details of these comments have been published previously (Premkumar et al., 2019; G. Y. Wang et al., 2021).

EEG recording was performed with a SynAmps amplifier and a 62-channel QuickCap with electrodes configured in the international 10-20 system. Electrodes channels were: FP1, FPZ, FP2, AF3, AF4, F7, F5, F3, F1, FZ, F2, F4, F6, F8, FT7, FC5, FC3, FC1, FCZ, FC2, FC4, FC6, FT8, T7, C5, C3, C1, CZ, C2, C4, C6, T8, TP7, CP5, CP3, CP1, CPZ, CP2, CP4, CP6, TP8, P7, P5, P3, P1, PZ, P2, P4, P6, P8,

PO7, PO5, PO3, POZ, PO4, PO6, PO8, CB1, O1, OZ, O2, CB2. Data was recorded at 1000Hz. Using multiple electrodes is a better approach than using a single electrode when assessing multiple levels of stress (Attallah, 2020). However, processing all the channels will require greater processing power. Therefore, FP1, FP2, T7 and T8 specific electrodes were selected. The selection of frontal electrodes were based on a previous EEG feature selection study conducted on stress classification which reported higher accuracy levels with FP1 and FP2 (Attallah, 2020). Moreover, since the stress stimulations were auditory, T7 and T8 were used in an attempt to capture the dynamics of the auditory cortex. Previously, emotional auditory stimuli had been found to evoke different levels of valence in individuals that co-varied significantly with EEG signals generated by the auditory region (Daly et al., 2019) and, negative valence is found to be strongly connected with stress (Seo & Lee, 2010).

EEG data pre-processing was performed in MATLAB 2019a (The Mathworks, Inc) (MATLAB, 2019) using custom scripts that involved functions from the EEGLAB plugin (Delorme & Makeig, 2004). Data were down-sampled offline to 256 Hz. A high-pass finite impulse response (FIR) filter at 0.01 Hz and a low-pass FIR filter at 50 Hz was applied. A baseline correction was not applied separately since the high-pass filter with low cutoff frequencies are found to rectify the baseline drift (Groppe et al., 2009). Using the CleanLine function (Delorme & Makeig, 2004), line noise was removed before data were manually inspected for the removal of bad channels (flat or extremely noisy). The removed channels were interpolated before an independent component analysis was performed to decompose the sample, using the runica function (Delorme & Makeig, 2004) from the MATLAB ICA Toolbox for Psychophysiological Data Analysis (Makeig & Al., 2000). The independent components derived from ICA were inspected, and muscular and ocular artefacts were removed from the data based on their activity spectra and scalp topographies. After the preprocessing steps, the last five seconds of the voltage signal was selected (Each original EEG signal consisted of 10 to 15 seconds. i.e., the stimulus presentation time). This extracted portion of the voltage signal was then converted into temporal spikes using AER protocol (Delbruck & Lichtsteiner, 2007) before feeding the SNNs. No other feature engineering or extractions were carried out.

Table 9.2

O-NSNN hyperparameters

Algorithm hyperparameter		Online learning with SP
AER encoder	f	0.7
LIF	v_{thresh}	0.05
	v_{rest}	0
	R	1
	C	10
STDP	A_+	0.001
	A_-	0.001
	τ_{pos}	10
	τ_{neg}	10
	w_{max}	0.5
	w_{min}	-0.5
IP	θ_{pos}	1×10^{-3}
	θ_{neg}	1×10^{-6}
Pruner	sp_{thresh}	1
Classifier	α	1
	mod	0.8
	drift	0.001

9.7 Results

We compared the classification accuracy and sensitivity of the O-NSNN model with the same learning framework without structural plasticity (SP) techniques (denoted as O-RSNN) and batch mode learning without SP (B-RSNN) (i.e., 70% of the samples for training and 30% for testing). The task involved measuring the accuracy of classifying EEG data into one of three possible classes: stress, neutral or positive conditions and the sensitivity (true positive rate) to recognize correctly classified stress instances. Since the synaptic weights of the first layer to the second are initiated randomly following Gaussian distribution, each experiment was conducted 30 times, allowing the accuracy and sensitivity to be reported statistically. The performance is discussed in terms of average accuracy and sensitivity (see Table 9.3) and efficiency. Furthermore, we explored patterns in network dynamics for knowledge extraction.

Table 9.3

Accuracy and sensitivity comparison between online (with (O-NSNN) and without SP (O-RSNN)) and batch mode (B-RSNN) learning

Participant ID	O-NSNN		O-RSNN		B-RSNN	
	Accuracy	Sensitivity	Accuracy	Sensitivity	Accuracy	Sensitivity
P1	0.94 ± 0.02	0.93 ± 0.05	0.66 ± 0.09	0.50 ± 005	0.84 ± 0.07	0.76 ± 0.12
P2	0.93 ± 0.03	0.93± 0.20	0.57 ± 0.07	0.49 ± 0.09	0.72 ± 0.08	0.75 ± 0.10
P3	0.91 ± 0.06	0.96± 0.70	0.60 ± 0.06	0.59 ± 0.14	0.69 ± 0.09	0.61 ± 0.15
P4	0.90 ± 0.08	0.88± 0.04	0.77 ± 0.11	0.67 ± 0.22	0.93 ± 0.05	0.96 ± 0.07
P5	0.91 ± 0.06	0.80± 0.06	0.66 ± 0.11	0.61 ± 0.27	0.83 ± 0.09	0.81 ± 0.14
P6	0.92 ± 0.03	0.90± 0.45	0.79 ± 0.08	0.72 ± 0.18	0.88 ± 0.08	0.83 ± 0.12
P7	0.90 ± 0.06	0.63± 0.04	0.59 ± 0.08	0.62 ± 0.05	0.74 ± 0.09	0.82 ± 0.11
P8	0.94 ± 0.02	0.95 ± 0.09	0.68 ± 0.11	0.66 ± 0.25	0.88 ± 0.09	0.90 ± 0.09
P9	0.94 ± 0.02	0.94 ± 0.07	0.76 ± 0.08	0.79 ± 0.06	0.75 ± 0.10	0.96 ±0.70
P10	0.91 ± 0.06	0.95 ± 0.34	0.42 ± 0.07	0.48 ± 0.13	0.49 ± 0.11	0.53 ± 0.18
P11	0.91 ± 0.07	0.96 ± 0.17	0.75 ± 0.08	0.64 ± 0.11	0.86 ± 0.08	0.84 ±0.14
P12	0.91 ± 0.04	0.93 ± 0.23	0.68 ± 0.08	0.76 ± 0.05	0.86 ± 0.08	0.81 ± 0.12
P13	0.92 ± 0.08	0.90 ± 0.06	0.82 ± 0.09	0.89 ± 0.21	0.92 ± 0.06	0.95 ± 0.07
P14	0.89 ± 0.07	0.86 ± 0.47	0.53 ± 0.10	0.54 ± 0.11	0.62 ± 0.10	0.60 ± 0.13
P15	0.92 ± 0.04	0.93 ± 0.09	0.66 ± 0.09	0.75 ± 0.14	0.77 ± 0.10	0.79 ± 0.16
P16	0.91 ± 0.04	0.93 ± 0.12	0.46 ± 0.10	0.36 ± 0.25	0.60 ± 0.12	0.71 ± 0.13
P17	0.86 ± 0.13	0.95 ± 0.05	0.49 ± 0.10	0.45 ± 0.21	0.71 ± 0.10	0.79 ± 0.16
P18	0.85 ± 0.09	0.87 ± 0.55	0.55 ± 0.09	0.55 ± 0.09	0.65 ± 0.09	0.62 ± 0.13
P19	0.90 ± 0.07	0.91 ± 0.06	0.41 ± 0.08	0.51 ± 0.15	0.61 ± 0.08	0.60 ± 0.16
P20	0.91 ± 0.07	0.95 ± 0.05	0.74 ± 0.10	0.64 ± 0.18	0.81 ± 0.08	0.87 ± 0.19
P21	0.90 ± 0.12	0.93 ± 0.04	0.64 ± 0.09	0.64 ± 0.05	0.75 ± 0.09	0.62 ±0.11
P22	0.92 ± 0.03	0.87 ± 0.13	0.67 ± 0.09	0.53 ± 0.11	0.82 ± 0.09	0.89 ± 0.12

9.7.1 Increased accuracy and robustness in O-NSNN

The highest average accuracy for O-NSNN was 93.63% for P1 and, the lowest was 85.29% for P18. The average accuracy across all participants was recorded at 90.91%, 63.18% and 76.04% for O-NSNN, O-RSNN and B-RSNN, respectively, whereas the average sensitivity was recorded at 90.27%, 60.86% and 77.36%. The O-NSNN outperformed O-RSNN across all 22 participants. In comparison, B-RSNN was outperformed in terms of accuracy by O-NSNN except for one participant (P4). Regarding sensitivity, the B-RSNN outperformed the O-NSNN with the data of P4, P5, P7, P9 and P22.

Table 9.4

Performance comparison with other studies that used the same data for stressed vs relaxed brain signal classification

Study	Method	Accuracy	Sensitivity
Bastos-Fiho et al.(Bastos-Filho et al., 2012)	K-NN (Batch mode)	0.70	-
Shon et al.(Shon et al., 2018)	K-NN (Batch mode)	0.72	
García-Martínez et al.(García-Martínez et al., 2017)	SVM (Batch mode)	0.81	-
Weerasinghe et al.(M. M. A. Weerasinghe et al., 2021b)	SNN (Batch mode)	0.92 ± 0.02	-
This study	NSNN (Online mode)	0.80 ± 0.05	0.79 ± 0.04

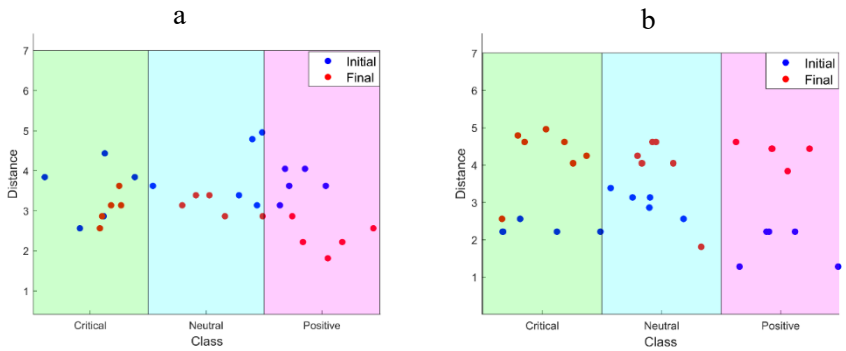


Figure 9.3: Euclidean Distance between initial(Blue) and final(Red) output neurons. The initiation process use the first 15 samples to evelove 15 output neurons. (a) without pruning or evolving new neurons (O-RSNN) (b) with pruining and evolving new neurons (O-NSNN)

The performance of the O-NSNN was also compared with the most relevant studies that used a common data source, the DEAP dataset (Koelstra et al., 2012), to classify stress vs relaxed brain signals (two classes). Here the O-NSNN recorded lower accuracy performance compared to batch mode experiments of SVM (García-Martínez et al., 2017) and SNN (Weerasinghe et al., 2021).

Figure 9.4 shows the variation of performance for personalized models for each participant obtained from 30 pseudo-random network initiations. Accordingly, for all 22 participants, the O-NSNN model had the lowest degree of performance variation.

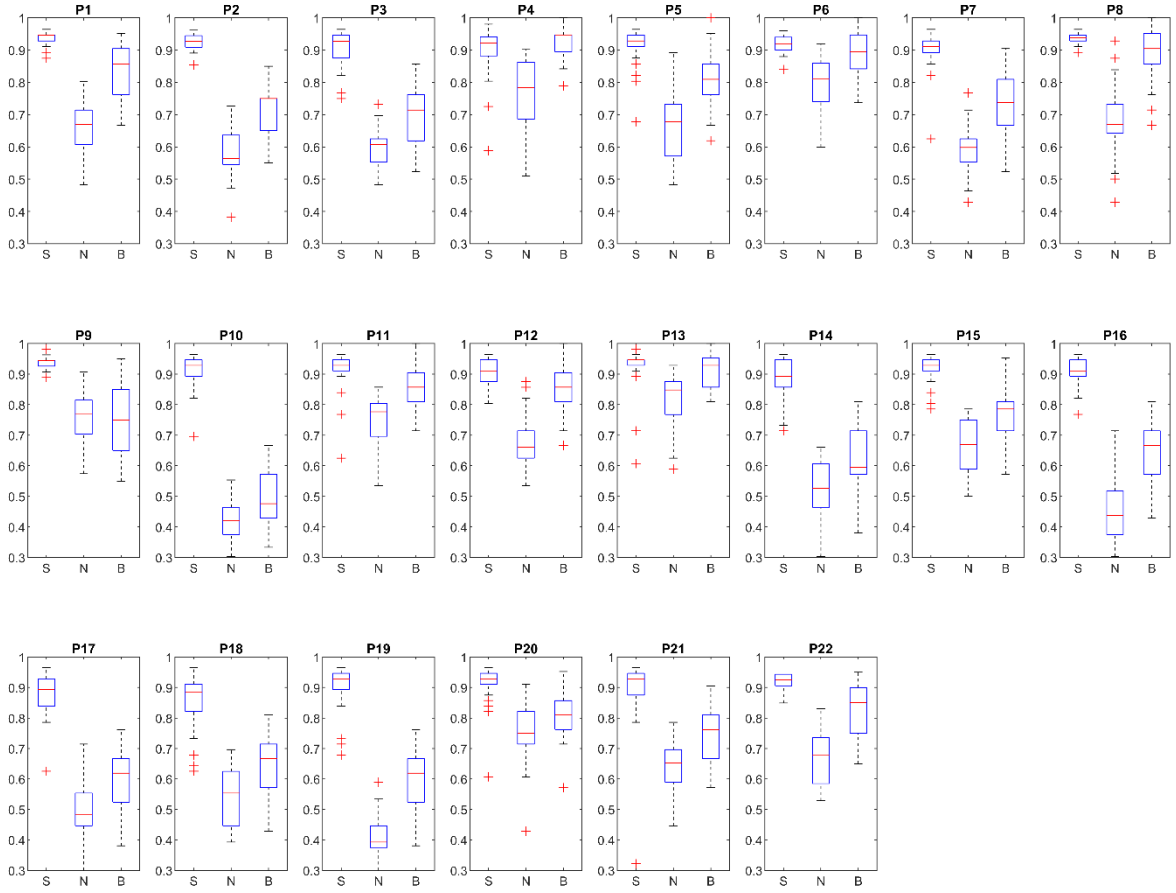


Figure 9.4: Performance variation of individual models. Performance distribution obtained from 30 testing cycles. At each cycle the initial weights between the input to hidden layers are selected pseudo randomly according to gaussian ditribution. S – Online learning with SP, N – Online learning without SP, B – Batch mode learning without SP

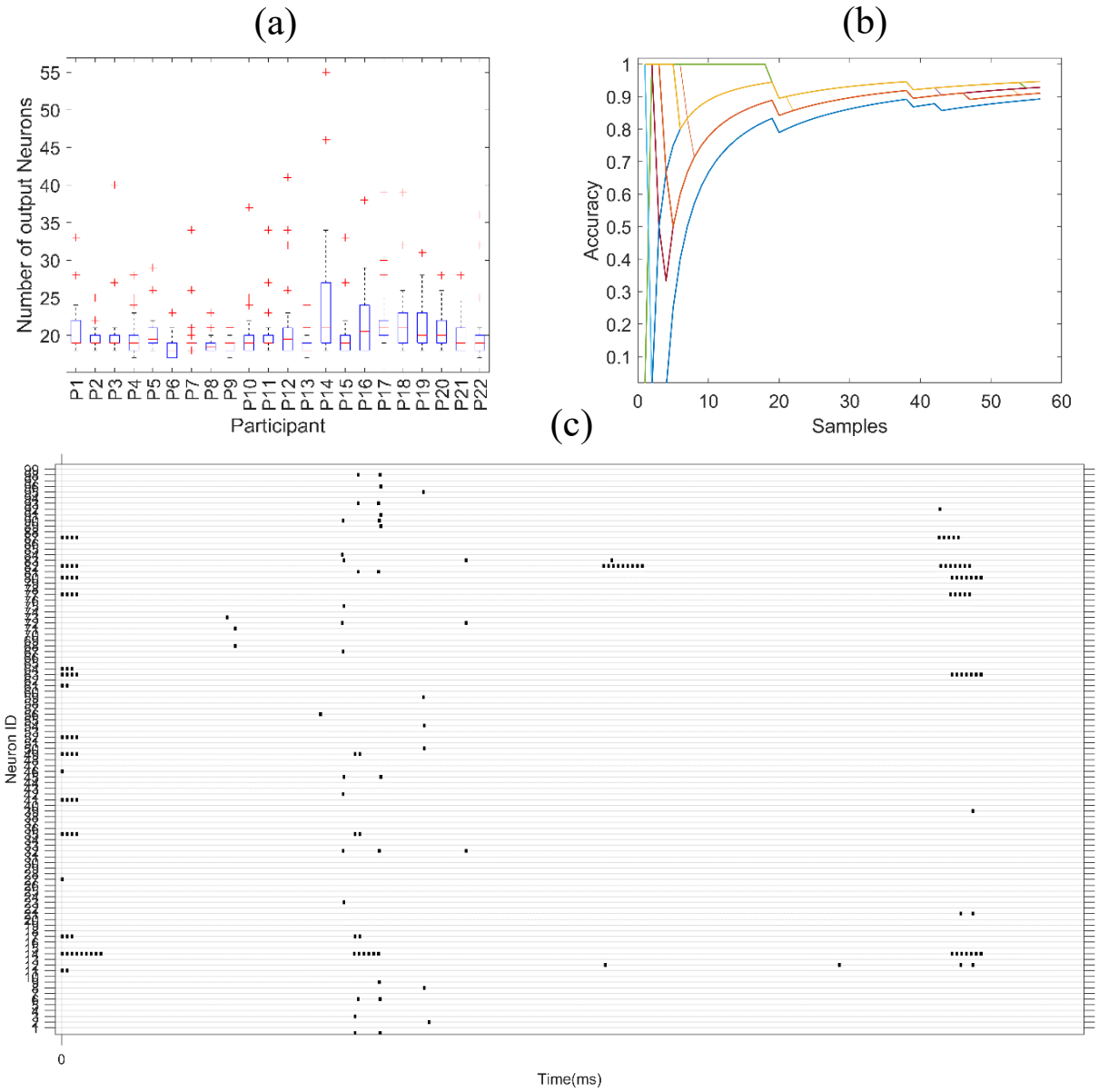


Figure 9.5:(a) Number of output neurons evolved by O-NSNN during 30 testing cycles for each participant model (b) Prequential accuracy progression with the number of samples increasing. Different colours for different participants (c) Sample spiking raster plot of the hidden layer for P1

9.7.1 The efficiency of O-NSNN

The efficiency factor of the O-NSNN can be presented in terms of the number of output neurons used and spikes generated in the hidden layer. When the number of output neurons used was investigated, the O-NSNN method used, on average 20.39 (σ = 3.84) neurons (see Figure 9.5a), whereas O-RSNN used 72 (i.e., absence of structural plasticity created a neuron for each input sample) and, B-RSNN used 50 output neurons respectively (i.e., 70/30 split training used 50 input samples for training where a neuron was created for each input). The spike generation of O-NSSN was measured as a ratio between the number of spikes received at the hidden layer to the number of spikes generated by the hidden layer,

where the mean was recorded at 0.063 (σ = 0.009). This spike encoding is epitomised in Figure 9.5c where the raster plot indicates the temporal sparseness of the spikes. When considering the trend of model accuracy over time, O-NSNN typically reached a prequential accuracy of 80% within 150 to 200 seconds of data processing commencement. An exception to this trend was noted in the case of P17 and P21 O-NSNN models (see Figure 9.5c).

9.7.2 O-NSNN knowledge extraction

We also analysed the Euclidean distance (ED) of the output neuron weight vectors and input to the hidden layer synaptic weights (i.e., STDP weights) of each individualised NSNN model. The evolved output neurons of an individualised NSNN model represented a certain class (i.e., stress, neutral or positive). The NSNN used this weight vector of the output neurons to predict the class of the incoming signals. Therefore, each ED of a sample is a numerical representation of the individual's brain signal under a given stimulus. Similarly, the weights of input to the hidden layer in NSNN are updated in an unsupervised method using STDP and IP. Once all the data samples are passed through the network, the NSNN weights (i.e., input to the hidden layer) capture the spatiotemporal correlations of the input signals.

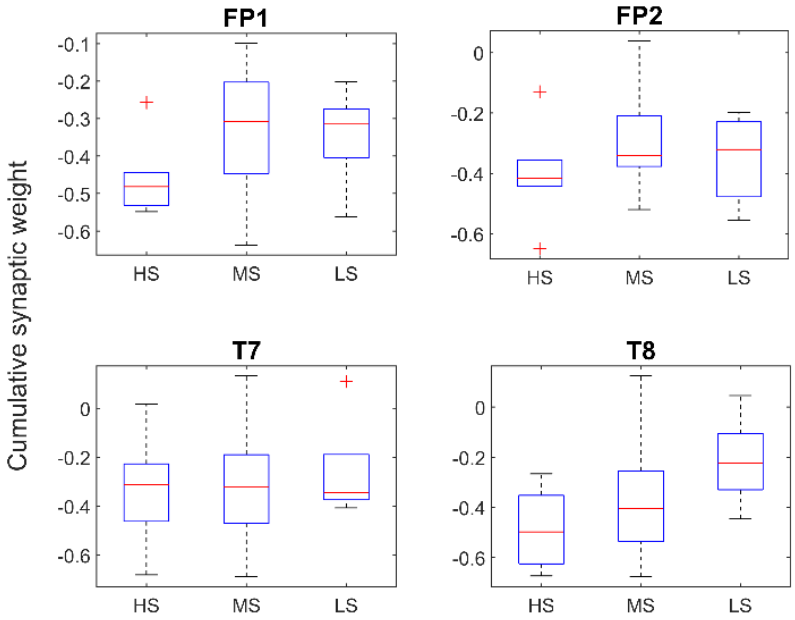


Figure 9.6: Cumulative weights of the synapses fanning out from respective inputs calculated according to perceived stress groups

We compared the EDs between the HS, MS, and LS groups and found that the mean distance between neutral and critical stimuli of the HS group was 0.95 ($\sigma = 0.41$). In contrast, the LS group's average distance between neutral and critical stimuli was much shorter at 0.25 ($\sigma = 0.22$). The average distance between neutral and positive stimuli of the HS group was 0.87 ($\sigma = 0.86$) and lower than that of the LS group's distance of 1.86 ($\sigma = 0.84$). According to these results, the HS group's EEG for positive stimuli did not differ to any notable extent from the EEG generated during neutral stimuli; this was the same for negative stimuli (i.e., under stress). However, the LS group recorded a much larger difference in both cases (see Figure 9.7a).

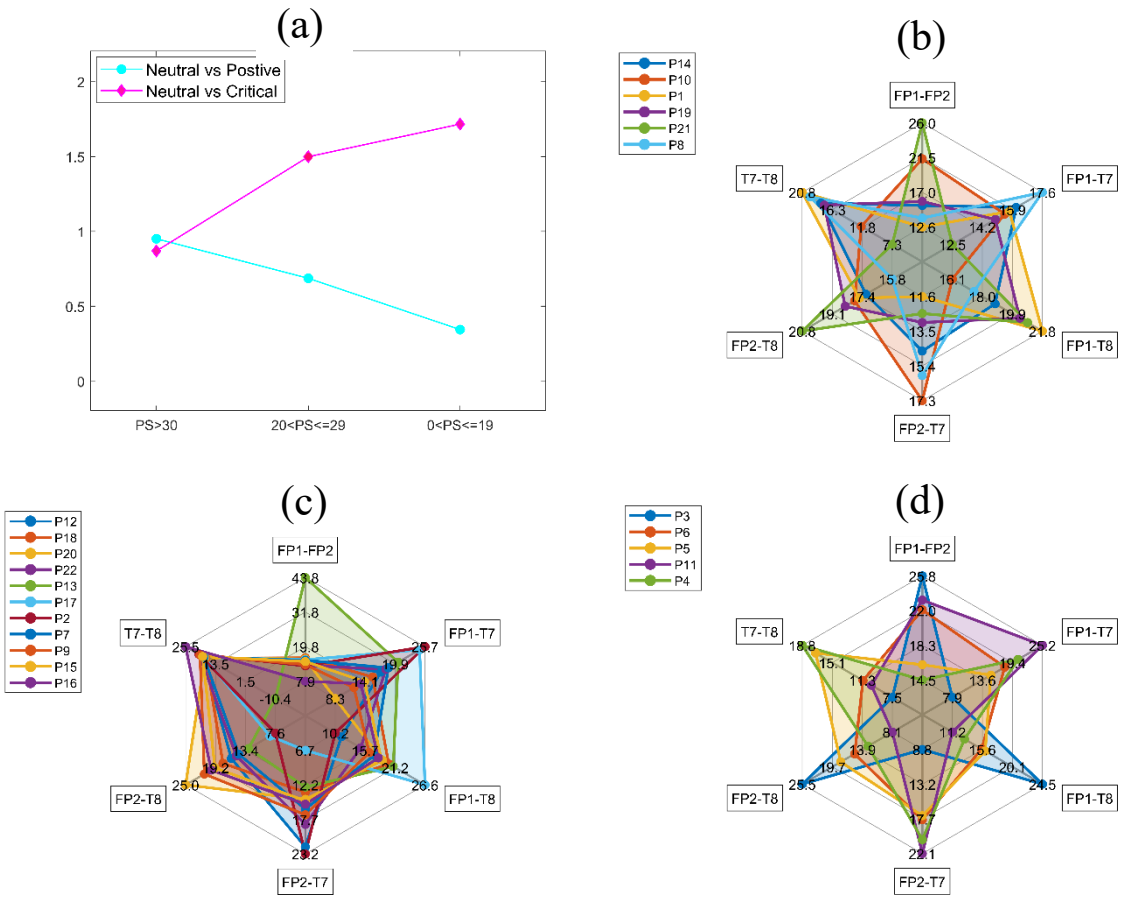


Figure 9.7: (a) Average differences between EEG samples represented by Euclidean distances. The signals during Neutral stimuli is selected as the baseline. (b) Spiking interaction pattern between channels for the High stress group (c) Spiking interaction pattern between channels for the Medium stress group (d) Spiking interaction pattern between channels for the Low stress group

Input channel correlation

When considering the activations between input channels (i.e., using the input to hidden layer synaptic weights), the majority of MS participants exhibited similar activation patterns (see Figure 9.7c), whereas the LS and HS groups exhibited irregular patterns of activation from one individual to another (see Figure 9.7b and Figure 9.7d). While investigating this further by examining the input synaptic weights of the hidden layer, we found that the HS group had higher inhibition than the LS group in the FP1 and FP2 channels (see Figure 9.6). The same inhibitory patterns were observed for T8 but not T7. When examining the right and left-brain activations, we discovered that the HS group showed higher inhibition in the right hemisphere (FP2 and T8) than in the left hemisphere (FP1 and T7). However, in the LS group, the average difference between right and left hemisphere activations was significantly smaller. Moreover, higher interactions were observed between FP1 and T8 than FP2 and T7 in five out of six participants in the HS group. The opposite activation pattern was observed in four out of five of the participants in the LS group.

9.8 Discussion

This study presents a neuroplasticity spiking neural network in an online learning setup for classifying the neural activity of healthy participants when exposed to comments that were intended to trigger different levels of mental states (i.e., stress, neutral, positive) and explores the link between these classifications and self-reported stress levels (i.e., perceived mental stress scores). This O-NSNN method produced higher pattern recognition capability on the fly, with increased efficiency, interpretability, and biological plausibility.

9.8.1 The performance of the O-NSNN.

The O-NSNN outperformed the other SNNs (O-RSNN and B-RSNN) in terms of average accuracy, as shown in Table 9.3. When comparing the two online learning methods, O-NSNN (90.76%, $\sigma = 2.09$) was found to perform significantly better than O-RSNN (63.08%, $\sigma = 11.09$) (Student's t-test, $\alpha=0.05$, $p=0.005$). As per Figure 9.4, the O-NSNN model produced the least performance variation indicating higher robustness (Navlakha et al., 2018). When considering the DEAP dataset, the O-NSNN could not outperform SNN and SVM techniques built for stress recognition (

Table 9.4). The methods that outperformed the O-NSNN used feature engineering (García-Martínez et al., 2017) or hyperparameter optimization (M. Weerasinghe et al., 2020) methods for the modelling tasks. Exploring the modelling mechanisms of O-NSNN, we found that the EDs of output neurons (i.e., numerical representations of input samples) to have better discriminative capability between the initial and final states of O-NSNN than in O-RSNN. This enhanced discriminative capability is presented in

Figure 9.5 for P1. With neurons evolving and self-pruning being the only difference between O-NSNN and O-RSNN, we propose this SP technique as a successful method for handling new classes and new representations of already-known classes. In other words, the O-NSSN approach is effective at handling concept drift.

STDP and IP learning

In a previous study, it was reported how hidden layer neuron pruning with STDP+IP leads to increased robustness and efficiency of SNNs in a batch learning setup for EEG classification(M. M. A. Weerasinghe et al., 2022). In the same study, hidden layer neurons with low firing probability causing classification errors were noted. In this study, instead of completely pruning these low-firing probability neurons, we have adopted a self-pruning method that stops a neuron activation for a limited period. This is achieved by increasing the neuron threshold voltage to the highest value found in the population. The advantage of this method is three-fold. Firstly, the inactivity of the neuron caused by threshold alteration help in reducing the number of dimensions used to represent an input sample at the output layer. Since classifications of the proposed O-NSNN are based on EDs calculated from output layer synaptic weights, part of the increase in performance can be attributed to the mitigation of the curse of dimensionality (Marimont & Shapiro, 1979). Secondly, the self-pruned neurons remain in the network to respond to salient features that may occur due to drifts or new data. This repurposing of neurons may account for the improvement of the performance of the network with time(Shi et al., 2019). Thirdly, the efficiency of this pruning is superior to regular synaptic pruning, which requires scanning of the entire weight matrix against a threshold(Rathi et al., 2019; Shi et al., 2019).

The efficiency of O-NSNN

The efficiency of the O-NSNN in terms of the number of neurons used and spikes generated reduced drastically with the use of STDP+IP learning and self-pruning. Unlike continuous streams of spiking, these techniques enabled sparser spiking activity resulting inactive states most of the time (see Figure 9.3c). When compared to STDP-only learning, STDP+IP was shown to have reduced the average spiking by 35 times (Student's t-test, α=0.05, p=0.008). This reduction of spikes minimises the calculations involved from the hidden to the output layer. Moreover, the O-NSNN output layer utilised 3.52- and 2.45-times fewer neurons on average when compared to the O-RSNN and B-RSNN models, respectively. In comparison to the early methods of evolving neurons where the spiking is not regulated(Dora et al., 2016; J. Wang et al., 2017) and the output repository grows indefinitely (Schliebs & Kasabov, 2013), this method is much more suitable for memory-restricted applications.

From trained NSNN models, HS participants showed lower activation levels in prefrontal channels FP1 and FP2 compared to the LS group. This was observed during the synaptic weight analysis of individual models, where the HS group had more inhibitory weights connected to FP1 and FP2 channels (see Figure 9.6). Moreover, the T8-connected synapses showed higher activations for the HS group (compared to the LS group), but this was not the case with T7-connected synapses (see Figure 9.6). In terms of the channel activation patterns, a similarity was observed among the individuals of the MS group but not in HS and LS groups (Figure 9.7b-d). In addition, the HS group had the smallest difference between EDs (numerical representations of spike patterns) produced during stress and positive stimuli compared to neutral states, whereas in the LS group, the observation was the opposite (Figure 9.7a). This suggests a lack of change in functional patterns of the brain to external stimuli in the HS group and a greater change in functional patterns in the LS group. This observation leads to an interesting hypothesis about the relationship between acute and perceived stress. Namely, the individuals with high perceived stress (HS group) have less discrimination between positive and negative stimuli. In a previous study, long-term stress has been found to alter the perception of emotional stimuli (Khosrowabadi, 2018).

9.8.2 Biological Plausibility in Learning

The biological plausibility of O-NSNN can be discussed in the aspects of data processing techniques employed and the spiking behaviour observed. Firstly, the data processing techniques inspired by neuroscientific concepts include STDP for temporal synaptic learning (Bi & Poo, 1998), IP for neuron spike regulation (Desai et al., 1999), self-pruning (apoptosis) to selectively restrict activation of neurons (Roth & D'Sa, 2001), and addition of new neurons (neurogenesis) for retention of new knowledge (Eriksson et al., 1998). Secondly, the model introduced demonstrates avalanche-like spiking, which is also found in neocortical circuits (Beggs & Plenz, 2003). Arguably, this makes O-NSNN much more biologically plausible than other online learning methods introduced, which do not utilise the same repertoire of plasticity techniques or show spiking behaviour close to what is found in biology (Dora et al., 2016; Lobo et al., 2018; J. Wang et al., 2014).

9.9 Conclusion

This work presents a novel neural network algorithm for mental stress classification using EEG data and online learning. The algorithm adapts to individuals and uses functional concepts of the biological brain to learn, on the fly, in a resource-efficient manner. The O-NSNN algorithm introduced displayed superior performance in terms of accuracy, robustness, and resource efficiency over models that did not use structural plasticity.

Our method introduced goes beyond traditional black box ANN models to reveal insights into individual brain dynamics for better interpretation. Improving the capability of this algorithm to recognise a higher number of classes under resource restrictions could potentially contribute to the applications of wearable technology for the detection and monitoring of mental stress.

Chapter 10 Conclusion & Future Work

This thesis aimed to develop efficient SNN learning algorithms for pattern classification. The investigations were carried out within the broader theme of biologically plausible learning, which has framed the development of ANNs throughout their history. Under these investigations, offline and online learning methods are proposed. Unlike much of the previously published work in the area, the algorithm development was intended to enhance efficiency[19] without compromising ML accuracy and robustness. The application areas included EEG classification of human motor movements, stress, and emotions.

The knowledge extraction capability found in SNNs separates them from traditional ANNs, which are often considered black-box approaches. Therefore, the models developed were also used for stress knowledge extraction by analysing the network behaviour. Some of the findings of these knowledge extraction experiments conformed with those reported in published literature, while others needed to be treated as hypotheses and investigated further.

This chapter first presents a summary of the key findings reported in this thesis and addresses each research question before highlighting the key contributions of this body of work to knowledge in the domains of SNNs and mental stress detection from EEG data. Finally, the chapter concludes with limitations of the work and some suggestions for future avenues of investigation.

10.1 Summary

The ability to change the network structure in a biological neural network is an integral part of learning (Deng et al., 2010; Eriksson et al., 1998; Kempermann et al., 2004; Shors et al., 2012) and maintaining energy efficiency (Butz et al., 2006; Roth & D'Sa, 2001; Shors et al., 2012). Some studies have attempted to integrate this concept in ASNNs to improve learning or efficiency. Some of these studies introduce techniques that prune networks to increase inference efficiency (Rathi et al., 2019; Shi et al., 2019). Others grow the classifier layers to accommodate new knowledge acquisition, enhancing learning (Dora et al., 2016; N. Kasabov et al., 2013; Lobo et al., 2018; J. Wang et al., 2014, 2017; Wysoski et al., 2008). None consider both efficiency and learning enhancement as goals for innovation. The relationship between ML performance and network efficiency means that performance is traded

[19] Efficiency refers to the resource utilisation vs the output. Resource utilisation is measured in terms of number of neurons and spikes used in an SNN for data processing. Output is measured using common ML performance metrics.

for advances in efficiency and vice versa. Thus, there is a need to investigate methods that can cohesively support learning capability and energy efficiency. From a practical point of view, these cohesive or win-win methods should also aim to replace exhaustive trial-and-error structural refinement methods and arbitrary allocation of resources (neurons and synapses) without proper scientific reasoning.

The literature review presented in Chapter 2 showed that most current SNNs developed using biologically plausible techniques dealt with ML problems involving static data. In contrast, the work of this thesis attempted to model EEG. EEG is a dynamic and complex data type that carries spatial and temporal information, often varying with time. Unlike a static data problem such as image processing, where pixel values are presented to the network at once, an EEG sample is presented to the network over a longer period. Therefore, learning from multiple spikes, over both long and short spike intervals, in a biologically plausible manner was a necessity. The basic SNN framework presented in this study (introduced in Chapter 4) catered to this need by incorporating an STDP learning layer which was unique from other comparative SNN algorithms that used an evolving classifier (Dora et al., 2016; N. Kasabov et al., 2013; Lobo et al., 2018). One study by Wang et al. (2014) presented an SNN algorithm equipped with STDP learning and a structurally adaptive classifier that used winner-takes-all neurons (selected based on the first neuron to spike) for classification. In contrast, the basic SNN architecture presented in this study used weights connected to the output neurons to calculate Euclidean distances for classification. These weights were updated according to Rank Order spiking and Spike Dependent Synaptic Plasticity (Fusi, 2003). According to the criteria of biologically plausible SNNs of Hao et al. (2020), the use of weights instead of spikes themselves may reduce the biological plausibility of the learning. However, it is important to note that the concept categorisation in the mammalian brain is most likely based on an entire circuit function (synapses and neurons) rather than the spiking pattern of an individual neuron (Seger & Miller, 2010). Therefore, it is reasonable to assume that using weights better represents a circuit function.

Before the investigations on methods to incorporate structural adaptations in the SNNs, the ML performance (split test accuracy) was tested against changing hidden layer sizes using a grid search. In these experiments (refer Section 4.4), certain numbers of neurons in the hidden layer performed better than others[20] in terms of average accuracy and robustness[21]. However, this finding is valid only for the search space used. Therefore, it is important not to interpret this finding as if there is an ideal number

[20] 111 neurons for Wrist dataset and 154 for DEAP dataset performed the best
[21] The robustness of the network was measured by 30 cycles of training and testing. At each cycle the initial weights, input mapping, training and testing sample selections were pseudo-randomly changed.

of neurons for a given ML task. However, the SNNs achieved significant performance levels within the search space. Nevertheless, it was clear that the number of neurons in the hidden layer operating with STDP could significantly affect the ML performance.

Following the preliminary experiments that demonstrated the impact of the number of neurons of the hidden layer, the process of incorporating structural adaptation was automated using a Differential Evolution (DE) algorithm (Storn & Price, 1997). However, previous studies have emphasised the importance of other hyperparameters on ML performance, such as the LIF threshold (Diehl & Cook, 2015; Hao et al., 2020). Therefore, in addition to structural adaptation, hyperparameter setting was also incorporated into the same adaptation mechanism. The introduction of DE for structural adaptation and hyperparameter tuning resulted in state-of-the-art accuracy levels for the datasets used. This approach also performed better compared to networks with 50% more and fewer neurons (refer Chapter 6). Interestingly, the best-performing SNNs also exhibited reduced neuron firing rates (see Figure 6.13) and fewer inactive neurons (refer Figure 6.10). In other words, the neurons lowered thresholds which increased sensitivity and resulted in more neurons participating in the training process. In previous simulation studies, researchers introduced the idea of *Runaway Synaptic Potentiation,* which is a condition that causes weights to increase independently in STDP learning and breaks the firing homeostasis of the SNN (Chen et al., 2013). Therefore, increasing weights (both negative and positive) could potentially result in hyper-activated and completely deactivated neurons. This supports the observation made in Chapter 6 related to overgrown networks, which showed high firing rates and redundant neurons. However, firing homeostasis can be effectively managed by having an appropriate number of neurons with neuron excitability modulation. This finding aligns with earlier work in which the importance of having the appropriate number of neurons to achieve firing homeostasis has been highlighted (Butz et al., 2006).

The structural adaptation method presented in Chapter 6 provided new insights into the correlations between spiking activity and performance. However, the DE method of structural adaptation was not biologically plausible or as computationally efficient as hoped. In an attempt to improve both biological plausibility and computational efficiency, an ensemble learning approach inspired by biological plasticity methods was investigated in Chapter 7. *Intrinsic plasticity* (IP) (Desai et al., 1999) and *pruning based on use it or lose it* (Shors et al., 2012) were incorporated into the SNN instead of DE. The pruned networks performed better in classifying data in one of the two EEG datasets (Wrist dataset) than unpruned networks. The IP training method presented here was unique. It used the active-to-

inactive neuron ratio maximisation[22] and entropy minimisation techniques to maintain firing homeostasis thereby increasing learning capability. This method differed from SPIKL-IP (W. Zhang & Li, 2019), an IP method introduced for maintaining firing homeostasis. SPIKL-IP does not use biological learning methods such as STDP and only learns by adjusting neuron excitability to produce a desired exponential firing distribution. SPIKL-IP is based on the information maximisation theory introduced by Bell et al.(Bell & Sejnowski, 1995). Information maximisation of each neuron could be a better approach for IP when STDP is not used. However, since the runaway synaptic weights could perpetually increase synaptic weights in the presence of STDP, information maximisation strategies may hypothetically lead to the overpopulation of spikes.

IP with STDP learning has been found to make neurons respond to independent input signal components (Savin et al., 2010). Therefore, this work hypothesised that the neurons fit into a hierarchy of firing rates, representing their contribution to the learning task at hand. The gradual decay of performance (observed in Chapter 8) when neurons were pruned according to the hierarchy of spikes generated supports this hypothesis. Moreover, the spiking behaviour in raster plots (see Figure 8.6b and Figure 8.9) exhibits avalanche-like spiking suggesting the neurons are operating in criticality. This biologically plausible spiking behaviour enhances pattern separation (X. Li, Chen, et al., 2017).

The IP and pruning techniques introduced in Chapter 8 were further developed to operate in an online learning environment for mental stress classification in Chapter 9 This chapter introduced an Online-Neuroplasticity Spiking Neural Network (O-NSNN) algorithm that operates with STDP, IP and Structural Plasticity techniques (SP). The hidden layer adds and removes neurons temporarily based on spiking activity, and the classifier layer adds neurons to represent new knowledge. The hidden layer resembles *Apoptosis* found in the mammalian brain, where neurons are eliminated based on usage after an initial proliferation[23] (Butz et al., 2006; Roth & D'Sa, 2001), and the output layer creates new neurons in the presence of new knowledge, resembling *Neurogenesis* (Deng et al., 2010; Eriksson et al., 1998). To date, there have been several online learning SNNs that are structurally adaptive, namely, eSNN (Wysoski et al., 2008), deSNN (N. Kasabov et al., 2013), SREN (Dora et al., 2016), structurally adaptive SNN (SASNN) (J. Wang et al., 2014) and O-SNN (Lobo et al., 2018). Structural adaptation is vital in handling concept drift, where input distributions of the same class and the distributions of classes themselves change over time (Lobo et al., 2018; Minku & Yao, 2012). Compared to other methods,

[22] Based on Chapter 6 findings that suggested increased number of active neurons in a SNN to have better accuracy and robustness

[23] The O-NSNN initiation sets the network with predefined number on neurons and based on usage neurons are temporarily deactivated.

except for SASNN, O-NSNN is the only algorithm that employs SP with STDP. Furthermore, it is the only algorithm where SP is adopted in both output and hidden layers. However, O-NSNN can be developed further by adopting effective pruning methods found in the output layer of O-SNN to forget obsolete knowledge representations and better manage memory requirements (Lobo et al., 2018).

The SNN algorithms presented in this study are evaluated using EEG data classification tasks of motor movements and brain states related to stress and emotions. When considering the performances of models presented in Chapter 5 and Chapter 6 state-of-the-art accuracies have been recorded in classifying acute stress[24]. Furthermore, Chapter 8 presents experiments that modelled valance levels, with the accuracy recorded at ~ 62%. The base study that produced the data set for valence classification recorded ~58% accuracy using a SVM classifier (Koelstra et al., 2012). The state-of-the-art accuracy levels for the same dataset was recorded at ~81% using a Convolutional Neural Network (Tripathi et al., 2017). However, it is important to note that Koelstra et al. and Tripathi et al. used individualised models and feature extractions. In contrast, modelling studies of Chapter 5 , Chapter 6 Chapter 7 and Chapter 8 have used global models for classification where common EEG patterns related to stress and emotions across participants are learned by the SNN. In contrast, Chapter 9 builds individual models for each participant. Using individualised models for the classification of EEG could be more advisable over global models since the brain functions related to intricate and complex subjects such as emotions and stress can vary from one participant to another.

The O-NSSN presented in Chapter 9 is one of the first attempts made to predict mental state related to acute stress on-the-fly. Its significance lies in the ability of the SNN to adapt to a particular person and operate efficiently with a high level of accuracy and robustness. Considering the importance of mental wellbeing in the modern day, the findings of the chapter can be further developed for personal health monitoring methods amalgamated with wearable technology. The study proceeds to extract knowledge of individual models to make inferences regarding acute and perceived stress based on network behaviour. Unlike other methods that use feature extractions prior to stress modelling (Bastos-Filho et al., 2012; Chao et al., 2018; García-Martínez et al., 2017; Koelstra et al., 2012; Shon et al., 2018; Tripathi et al., 2017) the O-NSNN models processes data without manual feature extraction techniques. Therefore, it is expected that the SNN models developed in this thesis to be more representative of the original data.

[24]Stress induced by audio and visual stimuli

10.2 Revisiting the research questions

This section revisits the research questions stated in Section 1.2. The central hypothesis that drives this body of work is that the increase in biological plausibility in learning strategies of SNNs would lead to better ML performance. The learning had to be fast and efficient to uphold the original premise of efficient data processing in SNNs.

Research Question 1: How does the number of LIF spiking neurons in an SNN with Spike Time Dependent Plasticity (STDP) learning affect data classification performance?

When the number of neurons is increased, the classification performance increases, producing better average accuracy and robustness. A significant saturation effect at a higher performance level occurs after a certain number of neurons (the number of neurons differs depending on the complexity of the function approximation). However, a further increase in neurones creates performance instabilities. The initial performance deficiency (before saturation) is caused by under-activation and the latter by over-activation. The inactive to active neuron ratio is an indication that can inform these network states in an SNN with STDP learning.

Another major factor that affects the number of neurons and data classification is the threshold settings of the LIF neurons. The number of neurons needed for a function approximation can be reduced if the thresholds are adapted. Importantly, this adaptation should enable sparser firing activity among neurons avoiding under-activation and over-activation.

The fundamental components of a neural network are neurons and synapses. These components directly affect the ML performance and the efficiency of the data processing in any neural network. ML research in non-spiking neurons has already introduced heuristics-based neural architecture search methods instead of the hand-designing of networks. However, in SNNs, there are no heuristics to determine the number of neurons required for an ML task. Therefore, the relationship between spiking behaviour, number of neurons and ML performance is not understood. This leads to the common approach of arbitrary or default setting of neurons for ML tasks. This thesis investigated answers to this research question through experiments presented in Chapter 5 and Chapter 6 . As a result, a novel SNN algorithm with STDP learning is introduced with an automated network structure optimisation method using Differential Evolution.

Research Question 2: How can brain-inspired plasticity techniques enable more efficient and accurate pattern separation in an SNN comprised of LIF neurons?

STDP and Intrinsic Plasticity (IP) are brain-inspired learning techniques operating at synaptic and neuron levels. The combined application (STDP+IP) of these techniques helps maintain spiking homeostasis levels in SNNs. Here, IP controls over and under activation, states are that found with standalone STDP. As a result, STDP+IP increases data processing efficiency and pattern separation accuracy compared to standalone STDP. More specifically, the data processing efficiency increases since IP limits the spiking rate of each neuron, whilst pattern separability is increased by driving all neurons towards activation, which maps the input to a higher dimensional space.

However, STDP and IP combined learning can lead to overfitting as well. This can be rectified by applying neuron pruning. This pruning should be applied based on the firing rates of individual neurones. The low-firing neurons contribute less to function approximation (pruning neurons in descending order of firing rates gradually decreases the performance). Since STDP+IP is known to tune neurones to represent independent components of the input, pruning low-firing neurones can be conjectured as the removal of non-salient feature representations which cause overfitting. However, if overfitting does not occur, pruning will lead to a performance decline. In terms of efficiency, neuron pruning after a training cycle improves data processing during the inferential cycle. Neuron elimination (Apoptosis) in the brain draws a close resemblance to this technique, where the initial proliferation of neurons is pruned based on the usage history.

The combined learning strategy of STDP and IP has rarely been tested on real-world applications. The studies that apply these strategies to real-world data lack clarity in terms of the STDP+IP governing rules. Moreover, the role of neuron pruning in an ensemble learning setup of STDP+IP was not investigated previously. To address this gap, an SNN with STDP+IP is presented in Chapter 8 . Here, STDP+IP is governed to reduce neuron redundancy and minimise information entropy of the entire network. This method outperformed standalone STDP and produced avalanche-like spiking behaviour, which is also found in biological systems. In biology, this spiking behaviour is characterised by the Power law. To fully answer this research question, further investigations should be conducted to understand the relationship between governing STDP+IP, spiking behaviours characterised by Power law and ML performance.

Research Question 3: How can brain-inspired plasticity techniques be incorporated into SNNs for mental stress recognition on the fly?

On-the-fly learning (online learning) demands the SNN to learn from a single pass and cope with concept drifts in EEG data. STDP+IP learning introduced in this thesis adapts weights and neuronal thresholds within a single pass in an unsupervised manner. However, when an already learned concept changes or a new class emerges, the final representation needs to be identified separately. Adding a new neuron at the output layer provides a solution to this. Besides adding new neurons at the output layer, pruning the hidden layer, depending on the usage, helps minimise possible overfitting caused by STDP+IP learning. However, it is important not to prune them fully and enable them to be reactivated if a substantial amount of input spikes arrive that represent salient information.

Neurogenesis (addition of new neurons) has been found in the hippocampal area of the mammalian brain as a mechanism that enables new memory formation for learning. This idea has inspired some SNN online learning studies to incorporate new neuron addition for new class representation or representational change of an already learned class. However, they are not equipped with techniques such as STDP+IP to fully exploit the temporal dynamics. In order to investigate the online learning capabilities of brain-inspired plasticity techniques, an SNN with a collection of plasticity techniques is introduced in Chapter 9 . Here, STDP+IP, pruning (self-pruning with an ability to reactivate), and new neuron addition is used in an online learning SNN namely, Neuroplasticity SNN (NSNN).

SNNs can be used to develop individual models (per participant) as well as global models. The SNN presented in Chapter 5 classifying stressed and relaxed states used a global modelling approach. Whereas the NSNN used to classify three classes (stressed, neutral and positive) in online mode used an individualised modelling approach. Comparatively, individualised modelling makes the function approximation simpler and requires fewer resources for data processing. Therefore, the method is suitable for low-power personal mental health monitoring solutions. However, the NSNN algorithm should be improved to restrict the output layer neuron growth. Not making this adjustment may cause memory issues in real-world implementation as the number of processing samples increases.

Research sub-question 3a: What knowledge can be extracted on stress using SNNs?

In the experiments presented in Chapter 9 , each NSNN is considered a function approximation representing each participant's brain activity. These functions are then compared in terms of synaptic weights to uncover spatiotemporal connectivity patterns for stress-related knowledge extraction. The perceived stress levels reported during EEG data collection by individuals were also utilised for this purpose.

Based on the NSNN behaviour, the brain signals of participants with high perceived stress showed less discrimination between positive and negative stimuli and vice versa for individuals with low perceived stress (Section 9.8.1). This observation highlighted the possibility of people with high perceived stress having less sensitivity to acute stressors, whereas individuals with low perceived stress having more sensitivity. Moreover, reduced prefrontal lobe activity was observed in participants with high perceived stress. In terms of the spiking correlation between EEG channels, the individuals with mid-level perceived stress showed similar patterns, whereas the high and low perceived stress cohorts did not show similarities between individuals. The consolidation of the knowledge extractions of this sub-question required further discussion with relevant psychology and neuroscience literature on stress.

10.3 Contribution

The work conducted during this doctoral study contributes to the body of SNN knowledge through three key methodological developments:

1. The introduction of foundational feedforward SNN with Differential Evolution-based architecture search setup.

A **foundational SNN algorithm** was proposed in Chapter 4 that has multi-spike processing capability. Most SNN algorithms use a single-spike approach which requires multiple passes of the same sample to learn. This foundational SNN uses STDP learning, which is considered more biologically plausible than the single-spike approach. This method can learn spike correlations on a longer and shorter time scales, supporting one-pass learning. Furthermore, a method based on Differential Evolution (DE) was incorporated into the SNN, enabling structural adaptation and hyperparameter optimisation. This systematic method can be used for setting up SNNs instead of using arbitrary structures. The structures developed using this method demonstrated more robust performance in pattern classification than arbitrary overgrown and undergrown structures. To the author’s knowledge, this is the first time

heuristics were generated for the structural optimisation of SNNs. The findings revealed that the inactive-to-active neuron ratio in the hidden layer is a predictor of ML performance.

2. A novel brain-inspired learning technique with rules for governing spiking behaviour.

The **ensemble learning algorithm** presented in Chapter 8 used a combined STDP+IP learning technique with a proliferate and prune method. The spike-controlling mechanism introduced for STDP+IP is novel and uses a mechanism that involves neuron activation and information entropy. Some studies have previously used STDP+IP for real-world data classification. However, their method of spike control was unclear. The new ensemble algorithm from this thesis classified EEG data with better efficiency and robustness than standalone STDP. Additionally, the better performing SNNs with STDP+IP produced avalanche-like spiking behaviour, also observed in biological systems. This leads to spike-based learning optimisation techniques and better understanding on how the brain computes information.

3. Neuroplasticity Spiking Neural Network algorithm for online learning and knowledge extraction.

The **Neuroplasticity Spiking Neural Network** developed for this thesis is an online learning algorithm that uses a collection of brain-inspired plasticity methods. Similar systems reported in the literature comprise of two-layer SNN architectures operating in online data classifications with structural adaptation in the classifier layer. However, they lack sufficient plasticity techniques such as STDP and IP to fully exploit temporal dynamics. Compared to those methods, the NSNN uses STDP+IP method with neuron pruning and addition capability in a data driven manner. The NSNN performed better than online learning methods without structural plasticity (neuron pruning and addition), elucidating aptness in using such techniques to tackle concept drift occurrences in streaming data.

Complexity and noise interferences are the primary challenges of EEG data processing, and this work proposes both online and offline methods to model such data using SNNs. Without restricting only to the classification of data, this thesis shows the use of SNNs in analysing brain data patterns between test subjects, including spatial activation patterns of individuals. Therefore, communities in psychology and neuroscience can benefit from SNNs as a knowledge extraction method in the time domain.

10.4 Limitations

Biologically plausible learning is an underlying theme of this study. Despite the best efforts to adhere to methods that may be closer to the biological system of the human brain, it is important to note this is not always possible. There are two underlying reasons for this:

1. There is a limited understanding of how the brain learns at the spike level, so the algorithms presented here are based on current assumptions and theories. For example, the method rules introduced to guide the STDP+IP process may not be biologically plausible, although it helps bring spiking activity that exhibits similar behaviours to those of the biological brain.
2. Computational tractability had to be prioritised for the experiments to be conducted within a feasible timeframe. One of the examples is the selection of the Leaky Integrate and Fire (LIF) neuron model instead of the Hodgkin-Huxley (HH) model. Although HH neuron is more biologically plausible computationally is much more expensive than LIF neuron.

In terms of hardware requirements, the proposed algorithms ideally should be simulated on Neuromorphic hardware. As per SNN literature, the true potential of SNN algorithms can only be revealed in such hardware. However, due to the limitation of Neuromorphic hardware availability, the simulations of this thesis are executed on standard computers operating with Von Neumann architecture.

The other limitation is related to the data selected for experiments. Firstly, using balanced datasets in batch learning experiments limits commenting on the system performance with unbalanced data. However, this does not apply to the online learning NSNN experiments presented, where the data used was unbalanced.

10.5 Future work

Future work should focus on three areas of improvement: algorithmic development, application expansion, and hardware implementation.

A**lgorithmic development:** One of the interesting findings of this research is the emergence of avalanche-like spiking behaviour. Further investigations in the area may reveal potential spiking behaviours that could enable better pattern separation and increase biological plausibility. This would not only enhance ML performance but also enable the generation of hypothesises on how the biological

brain encodes information for pattern classification. Moreover, the operation of the output layer (classifier layer) currently uses pre-synaptic weights for classification. Converting this to a spike-based method would decrease the time to make an inference.

Application expansion: The proposed systems need to be tested with data gathered from other brain imaging techniques, such as functional Magnetic Resonance Imaging (fMRI) data, ideally in the application area of stress classification. This is expected to validate the generalisability ML performance and the knowledge extraction capability of the proposed system. Imbalanced data should also be used to test the batch model learning methods presented since the datasets used in this thesis were balanced.

Hardware implementation: The applicability of the online stress recognition method proposed should be further tested on mobile hardware platforms to test the feasibility of its use in personal mental health monitoring.

References

Abbott, L. F., & Nelson, S. B. (2000). Synaptic plasticity: Taming the beast. *Nature Neuroscience*. https://doi.org/10.1038/81453

Abeles, M. (1982). Role of the cortical neuron: Integrator or coincidence detector? *Israel Journal of Medical Sciences*, *18*(1).

Abraham, W. C., Jones, O. D., & Glanzman, D. L. (2019). Is plasticity of synapses the mechanism of long-term memory storage? *Npj Science of Learning*, *4*(1), 9. https://doi.org/10.1038/s41539-019-0048-y

Aimone, J. B. (2016). Computational modeling of adult neurogenesis. *Cold Spring Harbor Perspectives in Biology*, *8*(4), a018960. https://doi.org/10.1101/cshperspect.a018960

Aimone, J. B., Li, Y., Lee, S. W., Clemenson, G. D., Deng, W., & Gage, F. H. (2014). Regulation and Function of Adult Neurogenesis: From Genes to Cognition. *Physiological Reviews*, *94*(4), 991–1026. https://doi.org/10.1152/physrev.00004.2014

Al-Fahoum, A. S., & Al-Fraihat, A. A. (2014). Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains. *ISRN Neuroscience*, *2014*, 1–7. https://doi.org/10.1155/2014/730218

Alarcao, S. M., & Fonseca, M. J. (2019). Emotions Recognition Using EEG Signals: A Survey. *IEEE Transactions on Affective Computing*, *10*(3), 374–393. https://doi.org/10.1109/TAFFC.2017.2714671

Alhagry, S., Aly, A., & A., R. (2017). Emotion Recognition based on EEG using LSTM Recurrent Neural Network. *International Journal of Advanced Computer Science and Applications*, *8*(10), 8–11. https://doi.org/10.14569/IJACSA.2017.081046

Apicella, A., Donnarumma, F., Isgrò, F., & Prevete, R. (2021). A survey on modern trainable activation functions. *Neural Networks*, *138*, 14–32. https://doi.org/10.1016/j.neunet.2021.01.026

Arnsten, A. F. T. (2009). Stress signalling pathways that impair prefrontal cortex structure and function. *Nature Reviews Neuroscience*, *10*(6), 410–422. https://doi.org/10.1038/nrn2648

Atluri, G., Karpatne, A., & Kumar, V. (2018). Spatio-temporal data mining: A survey of problems and methods. In *ACM Computing Surveys* (Vol. 51, Issue 4, pp. 1–41). https://doi.org/10.1145/3161602

Attallah, O. (2020). An effective mental stress state detection and evaluation system using minimum number of frontal brain electrodes. *Diagnostics*, *10*(5), 292. https://doi.org/10.3390/diagnostics10050292

Audiovisual, A., & Recognition, P. (2008). *Evolving Spiking Neural Networks for Adaptive Audiovisual Pattern Recognition Simei Gomes Wysoski Doctor of Philosophy (PhD)*.

Baernstein, H. D., & Hull, C. L. (1931). A mechanical model of the conditioned reflex. *Journal of General Psychology*, *5*(1), 99–106. https://doi.org/10.1080/00221309.1931.9918381

Bastos-Filho, T. F., Ferreira, A., Atencio, A. C., Arjunan, S., & Kumar, D. (2012). Evaluation of feature extraction techniques in emotional state recognition. *2012 4th International Conference on*

Intelligent Human Computer Interaction (IHCI), 1–6. https://doi.org/10.1109/IHCI.2012.6481860

Beggs, J. M., & Plenz, D. (2003). Neuronal Avalanches in Neocortical Circuits. *Journal of Neuroscience*, *23*(35), 11167–11177. https://doi.org/10.1523/jneurosci.23-35-11167.2003

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*(6), 1129–1159. https://doi.org/10.1162/neco.1995.7.6.1129

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*(7), 1413–1436. https://doi.org/10.1162/089976604323057443

Betti, S., Lova, R. M., Rovini, E., Acerbi, G., Santarelli, L., Cabiati, M., Ry, S. Del, & Cavallo, F. (2018). Evaluation of an integrated system of wearable physiological sensors for stress monitoring in working environments by using biological markers. *IEEE Transactions on Biomedical Engineering*, *65*(8). https://doi.org/10.1109/TBME.2017.2764507

Bi, G., & Poo, M. (1998). Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *The Journal of Neuroscience*, *18*(24), 10464–10472. https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998

Bialek, W., & Rieke, F. (1992). Reliability and information transmission in spiking neurons. *Trends in Neurosciences*, *15*(11), 428–434. https://doi.org/10.1016/0166-2236(92)90005-S

Bohte, S. M., Kok, J. N., & La Poutré, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. In *Neurocomputing* (Vol. 48, Issues 1–4). https://doi.org/10.1016/S0925-2312(01)00658-0

Borst, A., & Theunissen, F. E. (1999). Information theory and neural coding. In *Nature Neuroscience* (Vol. 2, Issue 11). https://doi.org/10.1038/14731

Bradley, M. M., & Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, *25*(1), 49–59. https://doi.org/10.1016/0005-7916(94)90063-9

Brons, J. F., & Woody, C. D. (1980). Long-term changes in excitability of cortical neurons after Pavlovian conditioning and extinction. *Journal of Neurophysiology*, *44*(3), 605–615. https://doi.org/10.1152/jn.1980.44.3.605

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *2020-Decem*. http://arxiv.org/abs/2005.14165

Butz, M., Lehmann, K., Dammasch, I. E., & Teuchert-Noodt, G. (2006). A theoretical network model to analyse neurogenesis and synaptogenesis in the dentate gyrus. *Neural Networks*, *19*(10), 1490–1505. https://doi.org/10.1016/j.neunet.2006.07.007

Calvo, R. A., & D'Mello, S. (2010). Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, *1*(1), 18–37. https://doi.org/10.1109/T-AFFC.2010.1

Capecci, E., Morabito, F. C., Campolo, M., Mammone, N., Labate, D., & Kasabov, N. (2015). A feasibility study of using the neucube spiking neural network architecture for modelling Alzheimer's disease EEG data. *Smart Innovation, Systems and Technologies*, *37*, 159–172. https://doi.org/10.1007/978-3-319-18164-6_16

Chao, H., Zhi, H., Dong, L., & Liu, Y. (2018). Recognition of Emotions Using Multichannel EEG Data and DBN-GC-Based Ensemble Deep Learning Framework. *Computational Intelligence and Neuroscience*, *2018*, 1–11. https://doi.org/10.1155/2018/9750904

Chaturvedi, S., & Khurshid, A. A. (2011). Review of spiking neural network architecture for feature extraction and dimensionality reduction. *International Conference on Emerging Trends in Engineering and Technology, ICETET*, 317–322. https://doi.org/10.1109/ICETET.2011.57

Chechik, G., Meilijson, I., & Ruppin, E. (1999). Neuronal regulation: A mechanism for synaptic pruning during brain maturation. *Neural Computation*, *11*(8), 2061–2080. https://doi.org/10.1162/089976699300016089

Chen, J. Y., Lonjers, P., Lee, C., Chistiakova, M., Volgushev, M., & Bazhenov, M. (2013). Heterosynaptic plasticity prevents runaway synaptic dynamics. *Journal of Neuroscience*, *33*(40), 15915–15929. https://doi.org/10.1523/JNEUROSCI.5088-12.2013

Chikara, R. K., & Ko, L.-W. (2019). Neural Activities Classification of Human Inhibitory Control Using Hierarchical Model. *Sensors*, *19*(17), 3791. https://doi.org/10.3390/s19173791

Clauset, A. (2011). *Inference, Models and Simulation for Complex Systems 1 A brief primer on probability distributions 1.1 Probability distribution functions*. http://tuvalu.santafe.edu/~aaronc/courses/7000/csci7000-001_2011_L0.pdf

Cline, H. T. (2001). Dendritic arbor development and synaptogenesis. In *Current Opinion in Neurobiology* (Vol. 11, Issue 1, pp. 118–126). Elsevier Ltd. https://doi.org/10.1016/S0959-4388(00)00182-3

Coan, J. A., & Allen, J. J. B. (2004). Frontal EEG asymmetry as a moderator and mediator of emotion. In *Biological Psychology* (Vol. 67, Issues 1–2, pp. 7–50). https://doi.org/10.1016/j.biopsycho.2004.03.002

Cohen, S., Kamarck, T., & Mermelstein, R. (1983). A Global Measure of Perceived Stress. *Journal of Health and Social Behavior*, *24*(4), 385. https://doi.org/10.2307/2136404

Copeland, B. J. (2020). *Artificial intelligence - Alan Turing and the beginning of AI*. Encyclopædia Britannica. https://www.britannica.com/technology/artificial-intelligence/Alan-Turing-and-the-beginning-of-AI

Cox, D. D., & Dean, T. (2014). Neural Networks and Neuroscience-Inspired Computer Vision. *Current Biology*, *24*(18), R921–R929. https://doi.org/10.1016/j.cub.2014.08.026

Crowley, O. V., McKinley, P. S., Burg, M. M., Schwartz, J. E., Ryff, C. D., Weinstein, M., Seeman, T. E., & Sloan, R. P. (2011). The interactive effect of change in perceived stress and trait anxiety on vagal recovery from cognitive challenge. *International Journal of Psychophysiology*, *82*(3), 225–232. https://doi.org/10.1016/j.ijpsycho.2011.09.002

Daly, I., Williams, D., Hwang, F., Kirke, A., Miranda, E. R., & Nasuto, S. J. (2019). Electroencephalography reflects the activity of sub-cortical brain regions during approach-withdrawal behaviour while listening to music. *Scientific Reports*, *9*(1), 9415.

https://doi.org/10.1038/s41598-019-45105-2

Damasio, A. R., Grabowski, T. J., Bechara, A., Damasio, H., Ponto, L. L. B., Parvizi, J., & Hichwa, R. D. (2000). Subcortical and cortical brain activity during the feeling of self-generated emotions. *Nature Neuroscience*, *3*(10), 1049–1056. https://doi.org/10.1038/79871

Dawid, A. P., & Vovk, V. G. (1999). Prequential Probability: Principles and Properties. *Bernoulli*, *5*(1), 125. https://doi.org/10.2307/3318616

Decker, D., Schöndorf, M., Bidlingmaier, F., Hirner, A., & von Ruecker, A. A. (1996). Surgical stress induces a shift in the type-1/type-2 T-helper cell balance, suggesting down-regulation of cell-mediated and up-regulation of antibody-mediated immunity commensurate to the trauma. *Surgery*, *119*(3), 316–325. https://doi.org/10.1016/S0039-6060(96)80118-8

Delbruck, T., & Lichtsteiner, P. (2007). Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. *2007 IEEE International Symposium on Circuits and Systems*, *80 cm*, 845–848. https://doi.org/10.1109/ISCAS.2007.378038

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, *134*, 9–21. http://www.sccn.ucsd.edu/eeglab/

Deng, W., Aimone, J. B., & Gage, F. H. (2010). New neurons and new memories: how does adult hippocampal neurogenesis affect learning and memory? *Nature Reviews Neuroscience*, *11*(5), 339–350. https://doi.org/10.1038/nrn2822

Dennis, M. (2021). *Marvin Minsky | American scientist | Britannica*. https://www.britannica.com/biography/Marvin-Lee-Minsky

Desai, N. S., Rutherford, L. C., & Turrigiano, G. G. (1999). Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nature Neuroscience*, *2*(6), 515–520. https://doi.org/10.1038/9165

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *9*. https://doi.org/10.3389/fncom.2015.00099

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., & Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. *Proceedings of the International Joint Conference on Neural Networks*, *2015-Septe*, 1–8. https://doi.org/10.1109/IJCNN.2015.7280696

Doborjeh, Z., Doborjeh, M., Taylor, T., Kasabov, N., Wang, G. Y., Siegert, R., & Sumich, A. (2019). Spiking Neural Network Modelling Approach Reveals How Mindfulness Training Rewires the Brain. *Scientific Reports*, *9*(1), 6367. https://doi.org/10.1038/s41598-019-42863-x

Domingos, P., & Hulten, G. (2003). A General Framework for Mining Massive Data Streams. *Journal of Computational and Graphical Statistics*, *12*(4). https://doi.org/10.1198/1061860032544

Dora, S., Subramanian, K., Suresh, S., & Sundararajan, N. (2016). Development of a Self-Regulating Evolving Spiking Neural Network for classification problem. *Neurocomputing*, *171*, 1216–1229. https://doi.org/10.1016/j.neucom.2015.07.086

Dora, S., Sundaram, S., & Sundararajan, N. (2015). A two stage learning algorithm for a Growing-Pruning Spiking Neural Network for pattern classification problems. *2015 International Joint*

Conference on Neural Networks (IJCNN), *2015-Septe*(978), 1–7. https://doi.org/10.1109/IJCNN.2015.7280592

Draelos, T. J., Miner, N. E., Lamb, C. C., Cox, J. A., Vineyard, C. M., Carlson, K. D., Severa, W. M., James, C. D., & Aimone, J. B. (2017). Neurogenesis deep learning: Extending deep networks to accommodate new classes. *2017 International Joint Conference on Neural Networks (IJCNN)*, *2017-May*, 526–533. https://doi.org/10.1109/IJCNN.2017.7965898

Dutoit, X., Schrauwen, B., Van Campenhout, J., Stroobandt, D., Van Brussel, H., & Nuttin, M. (2009). Pruning and regularization in reservoir computing. *Neurocomputing*, *72*(7–9), 1534–1546. https://doi.org/10.1016/j.neucom.2008.12.020

Eklund, A., Nichols, T. E., & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, *113*(28), 7900–7905. https://doi.org/10.1073/pnas.1602413113

Emotiv. (2021). *EMOTIV | Brain Data Measuring Hardware and Software Solutions*. Emotiv Inc. https://www.emotiv.com/

Epel, E. S., Crosswell, A. D., Mayer, S. E., Prather, A. A., Slavich, G. M., Puterman, E., & Mendes, W. B. (2018). More than a feeling: A unified view of stress measurement for population science. *Frontiers in Neuroendocrinology*, *49*(March), 146–169. https://doi.org/10.1016/j.yfrne.2018.03.001

Eriksson, P. S., Perfilieva, E., Björk-Eriksson, T., Alborn, A.-M., Nordborg, C., Peterson, D. A., & Gage, F. H. (1998). Neurogenesis in the adult human hippocampus. *Nature Medicine*, *4*(11), 1313–1317. https://doi.org/10.1038/3305

FitzHugh, R. (1961). Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal*. https://doi.org/10.1016/S0006-3495(61)86902-6

Florian, R. V. (2012). The Chronotron: A Neuron That Learns to Fire Temporally Precise Spike Patterns. *PLoS ONE*, *7*(8), e40233. https://doi.org/10.1371/journal.pone.0040233

Frémaux, N., & Gerstner, W. (2015). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in Neural Circuits*, *9*(JAN2016). https://doi.org/10.3389/fncir.2015.00085

Frick, A., & Johnston, D. (2005). Plasticity of dendritic excitability. In *Journal of Neurobiology* (Vol. 64, Issue 1, pp. 100–115). https://doi.org/10.1002/neu.20148

Froemke, R. C., Tsay, I. A., Raad, M., Long, J. D., & Dan, Y. (2006). Contribution of individual spikes in burst-induced long-term synaptic modification. *Journal of Neurophysiology*, *95*(3). https://doi.org/10.1152/jn.00910.2005

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics 1975 20:3*, *20*(3), 121–136. https://doi.org/10.1007/BF00342633

Fusi, S. (2003). Spike-driven synaptic plasticity for learning correlated patterns of mean firing rates. *Reviews in the Neurosciences*, *14*(1–2), 73–84. https://doi.org/10.1515/REVNEURO.2003.14.1-2.73

Fusi, S., Annunziato, M., Badoni, D., Salamon, A., & Amit, D. J. (2000). Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation. *Neural Computation*, *12*(10), 2227–2258.

https://doi.org/10.1162/089976600300014917

García-Martínez, B., Martínez-Rodrigo, A., Zangróniz, R., Pastor, J. M., & Alcaraz, R. (2017). Symbolic analysis of brain dynamics detects negative stress. *Entropy*, *19*(5), 196. https://doi.org/10.3390/e19050196

Gerstner, W., Kempter, R., van Hemmen, J. L., & Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, *383*(6595), 76–78. https://doi.org/10.1038/383076a0

Gerstner, W., & Kistler, W. M. (2002). Spiking Neuron Models. In *Spiking Neuron Models*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815706

Ghosh-Dastidar, S., & Adeli, H. (2007). Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*, *14*(3). https://doi.org/10.3233/ica-2007-14301

Golmohammadi, M., Harati Nejad Torbati, A. H., Lopez de Diego, S., Obeid, I., & Picone, J. (2019). Automatic analysis of EEGs using big data and hybrid deep learning architectures. *Frontiers in Human Neuroscience*, *13*, 76. https://doi.org/10.3389/fnhum.2019.00076

Grech, R., Cassar, T., Muscat, J., Camilleri, K. P., Fabri, S. G., Zervakis, M., Xanthopoulos, P., Sakkalis, V., & Vanrumste, B. (2008). Review on solving the inverse problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, *5*(1), 1–33. https://doi.org/10.1186/1743-0003-5-25/COMMENTS

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Groppe, D. M., Makeig, S., & Kutas, M. (2009). Identifying reliable independent components via split-half comparisons. *NeuroImage*, *45*(4), 1199–1211. https://doi.org/10.1016/J.NEUROIMAGE.2008.12.038

Grüning, A., & Bohte, S. M. (2014). Spiking neural networks: Principles and challenges. *22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2014 - Proceedings*, 1–10. http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Gütig, R., & Sompolinsky, H. (2006). The tempotron: A neuron that learns spike timing-based decisions. *Nature Neuroscience*. https://doi.org/10.1038/nn1643

Guyonneau, R., VanRullen, R., & Thorpe, S. J. (2005). Neurons Tune to the Earliest Spikes Through STDP. *Neural Computation*, *17*(4), 859–879. https://doi.org/10.1162/0899766053429390

Hammoodi, M. S., Stahl, F., & Badii, A. (2018). Real-time feature selection technique with concept drift detection using adaptive micro-clusters for data stream mining. *Knowledge-Based Systems*, *161*, 205–239. https://doi.org/10.1016/J.KNOSYS.2018.08.007

Hao, Y., Huang, X., Dong, M., & Xu, B. (2020). A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule. *Neural Networks*, *121*, 387–395. https://doi.org/10.1016/j.neunet.2019.09.007

Hebb, D. O. (2005). The Organization of Behavior. In *Brain Research Bulletin* (Vol. 50, Issues 5–6). Psychology Press. https://doi.org/10.4324/9781410612403

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/SCIENCE.1127647/SUPPL_FILE/HINTON.SOM.PDF

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500–544. https://doi.org/10.1113/jphysiol.1952.sp004764

Hogg, R. V, Tanis, E. A., Zimmerman Boston, D. L., Indianapolis, C., York, N., Francisco, S., Madrid, L., Munich, M., Montreal, P., Delhi, T., Sao, M. C., Sydney, P., Kong, H., Singapore, S., & Tokyo, T. (1977). *PROBABILITY AND STATISTICAL INFERENCE Ninth Edition Upper Saddle River Amsterdam Cape Town Dubai*. www.pearsonhighered.com

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554

Hosseini, S. A., & Khalilzadeh, M. A. (2010). Emotional Stress Recognition System Using EEG and Psychophysiological Signals: Using New Labelling Process of EEG Signals in Emotional Stress State. *2010 International Conference on Biomedical Engineering and Computer Science*, *November 2014*, 1–6. https://doi.org/10.1109/ICBECS.2010.5462520

Hough, M., de Garis, H., Korkin, M., Gers, F., & Nawa, N. E. (1999). SPIKER Analog waveform to digital spiketrain conversion in ATR's artificial brain (CAM-Brain) project. *International Conference on Robotics and Artificial Life*. http://www.stanford.edu/mhoughhttp://www.hip.atr.co.jp/fdegaris,xnawaghttp://www.genobyte.comhttp://www.idsia.ch/felix

Hu, J., Hou, Z. G., Chen, Y. X., Kasabov, N., & Scott, N. (2014). EEG-based classification of upper-limb ADL using SNN for active robotic rehabilitation. *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, 409–414. https://doi.org/10.1109/biorob.2014.6913811

Humble, J., Denham, S., & Wennekers, T. (2012). Spatio-temporal pattern recognizers using spiking neurons and spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *6*(SEPTEMBER), 1–12. https://doi.org/10.3389/fncom.2012.00084

Iglesias, J., & Villa, A. E. P. (2006). Neuronal Cell Death and Synaptic Pruning Driven by Spike-Timing Dependent Plasticity. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 4131 LNCS* (pp. 953–962). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11840817_99

Iglesias, J., & Villa, A. E. P. (2007). Effect of stimulus-driven pruning on the detection of spatiotemporal patterns of activity in large neural networks. *BioSystems*, *89*(1–3), 287–293. https://doi.org/10.1016/j.biosystems.2006.05.020

Izhikevich, E. M. (2004). Which Model to Use for Cortical Spiking Neurons? *IEEE Transactions on Neural Networks*, *15*(5), 1063–1070. https://doi.org/10.1109/TNN.2004.832719

Izhikevich, E. M. (2006). Polychronization: Computation with Spikes. *Neural Computation*, *18*(2). https://doi.org/10.1162/089976606775093882

Izhikevich, E. M. M. (2003). Simple Model of Spiking Neurons, IEEE Transctions on Neural Networks. *IEEE Trans. Neural Networks*, *14*(6), 1569–1572. https://doi.org/10.1109/TNN.2003.820440

Jaeger, H. (2003). Adaptive nonlinear system identification with Echo State networks. *Advances in Neural Information Processing Systems*. http://www.ais.fraunhofer.de/INDY

James, C. D., Aimone, J. B., Miner, N. E., Vineyard, C. M., Rothganger, F. H., Carlson, K. D., Mulder, S. A., Draelos, T. J., Faust, A., Marinella, M. J., Naegle, J. H., & Plimpton, S. J. (2017). A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications. *Biologically Inspired Cognitive Architectures*, *19*, 49–64. https://doi.org/10.1016/j.bica.2016.11.002

Javed, F., He, Q., Davidson, L. E., Thornton, J. C., Albu, J., Boxt, L., Krasnow, N., Elia, M., Kang, P., Heshka, S., & Gallagher, D. (2010). Brain and high metabolic rate organ mass: contributions to resting energy expenditure beyond fat-free mass. *The American Journal of Clinical Nutrition*, *91*(4), 907–912. https://doi.org/10.3945/ajcn.2009.28512

Jin, P. (1992). Efficacy of Tai Chi, brisk walking, meditation, and reading in reducing mental and emotional stress. *Journal of Psychosomatic Research*, *36*(4), 361–370. https://doi.org/10.1016/0022-3999(92)90072-A

Kasabov, N., Dhoble, K., Nuntalid, N., & Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Networks*, *41*, 188–201. https://doi.org/10.1016/j.neunet.2012.11.014

Kasabov, N. K. (2014). NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, *52*, 62–76. https://doi.org/10.1016/j.neunet.2014.01.006

Kasabov, N., Scott, N. M., Tu, E., Marks, S., Sengupta, N., Capecci, E., Othman, M., Doborjeh, M. G., Murli, N., Hartono, R., Espinosa-Ramos, J. I., Zhou, L., Alvi, F. B., Wang, G., Taylor, D., Feigin, V., Gulyaev, S., Mahmoud, M., Hou, Z. G., & Yang, J. (2016). Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications. *Neural Networks*, *78*, 1–14. https://doi.org/10.1016/j.neunet.2015.09.011

Kasiński, A., & Ponulak, F. (2006). Comparison of supervised learning methods for spike time coding in spiking neural networks. In *International Journal of Applied Mathematics and Computer Science* (Vol. 16, Issue 1, pp. 101–113).

Katmah, R., Al-Shargie, F., Tariq, U., Babiloni, F., Al-Mughairbi, F., & Al-Nashash, H. (2021). A Review on Mental Stress Assessment Methods Using EEG Signals. *Sensors*, *21*(15), 5043. https://doi.org/10.3390/s21155043

Kempermann, G., Wiskott, L., & Gage, F. H. (2004). Functional significance of adult neurogenesis. In *Current Opinion in Neurobiology* (Vol. 14, Issue 2, pp. 186–191). https://doi.org/10.1016/j.conb.2004.03.001

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Stdp-, T. M. (2022). STDP-based spiking deep convolutional neural networks for object recognition To cite this version : HAL Id : hal-02341957. *Neural Networks*, *99*, 56–67. https://doi.org/10.1016/j.neunet.2017.12.005

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, *99*. https://doi.org/10.1016/j.neunet.2017.12.005

Khosrowabadi, R. (2018a). Stress and perception of emotional stimuli: Long-term stress rewiring the

brain. *Basic and Clinical Neuroscience*, *9*(2). https://doi.org/10.29252/nirp.bcn.9.2.107

Khosrowabadi, R. (2018b). Stress and Perception of Emotional Stimuli: Long-term Stress Rewiring the Brain. *Basic and Clinical Neuroscience*, *9*(2), 107. https://doi.org/10.29252/NIRP.BCN.9.2.107

Khosrowabadi, R., Quek, C., Ang, K. K., Tung, S. W., & Heijnen, M. (2011). A Brain-computer interface for classifying EEG correlates of chronic mental stress. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2011.6033297

Klem, G. H., Lüders, H. O., Jasper, H. H., & Elger, C. (1961). The Ten Twenty Electrode System: International Federation of Societies for Electroencephalography and Clinical Neurophysiology. *American Journal of EEG Technology*, *1*(1), 13–19. https://doi.org/10.1080/00029238.1961.11080571

Ko, L., Chikara, R. K., Lee, Y., & Lin, W. (2020). Exploration of User's Mental State Changes during Performing Brain–Computer Interface. *Sensors*, *20*(11), 3169. https://doi.org/10.3390/s20113169

Koelstra, S., Muhl, C., Soleymani, M., Jong-Seok Lee, Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., & Patras, I. (2012). DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, *3*(1), 18–31. https://doi.org/10.1109/T-AFFC.2011.15

König, P., Engel, A. K., & Singer, W. (1996). Integrator or coincidence detector? The role of the cortical neuron revisited. *Trends in Neurosciences*, *19*(4). https://doi.org/10.1016/S0166-2236(96)80019-1

Kumarasinghe, K., Kasabov, N., & Taylor, D. (2021). Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements. *Scientific Reports*, *11*(1), 2486. https://doi.org/10.1038/s41598-021-81805-4

LaRocco, J., Le, M. D., & Paeng, D. G. (2020). A Systemic Review of Available Low-Cost EEG Headsets Used for Drowsiness Detection. *Frontiers in Neuroinformatics*, *14*, 42. https://doi.org/10.3389/FNINF.2020.553352/BIBTEX

Laughlin, S. (1981). A Simple Coding Procedure Enhances a Neuron's Information Capacity. *Zeitschrift Für Naturforschung C*, *36*(9–10), 910–912. https://doi.org/10.1515/znc-1981-9-1040

Lawrence, D. (2000). Central/peripheral nervous system and immune responses. *Toxicology*, *142*(3), 189–201. https://doi.org/10.1016/S0300-483X(99)00144-4

Lazar, A., Pipa, G., & Triesch, J. (2007). Fading memory and time series prediction in recurrent networks with different forms of plasticity. *Neural Networks*, *20*(3), 312–322. https://doi.org/10.1016/j.neunet.2007.04.020

Lee, C. S., Wang, M. H., Yen, S. J., Wei, T. H., Wu, I. C., Chou, P. C., Chou, C. H., Wang, M. W., & Yan, T. H. (2016). Human vs. Computer Go: Review and Prospect [Discussion Forum]. *IEEE Computational Intelligence Magazine*, *11*(3), 67–72. https://doi.org/10.1109/MCI.2016.2572559

Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, *10*. https://doi.org/10.3389/fnins.2016.00508

Legenstein, R., Naeger, C., & Maass, W. (2005). What can a neuron learn with spike-timing-dependent plasticity? *Neural Computation*, *17*(11). https://doi.org/10.1162/0899766054796888

Lerner, J. S., Dahl, R. E., Hariri, A. R., & Taylor, S. E. (2007). Facial Expressions of Emotion Reveal Neuroendocrine and Cardiovascular Stress Responses. *Biological Psychiatry*, *61*(2). https://doi.org/10.1016/j.biopsych.2006.08.016

Li, C. (2011). A Model of Neuronal Intrinsic Plasticity. *IEEE Transactions on Autonomous Mental Development*, *3*(4), 277–284. https://doi.org/10.1109/TAMD.2011.2159379

Li, C., & Li, Y. (2013). A Spike-Based Model of Neuronal Intrinsic Plasticity. *IEEE Transactions on Autonomous Mental Development*, *5*(1), 62–73. https://doi.org/10.1109/TAMD.2012.2211101

Li, X., Chen, Q., & Xue, F. (2017). Biological modelling of a computational spiking neural network with neuronal avalanches. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *375*(2096), 20160286. https://doi.org/10.1098/rsta.2016.0286

Li, X., Song, D., Zhang, P., Yu, G., Hou, Y., & Hu, B. (2017). Emotion recognition from multi-channel EEG data through Convolutional Recurrent Neural Network. *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*. https://doi.org/10.1109/BIBM.2016.7822545

Li, X., Wang, W., Xue, F., & Song, Y. (2018). Computational modeling of spiking neural network with learning rules from STDP and intrinsic plasticity. *Physica A: Statistical Mechanics and Its Applications*, *491*, 716–728. https://doi.org/10.1016/j.physa.2017.08.053

Lobo, J. L., Del Ser, J., Bifet, A., & Kasabov, N. (2020a). Spiking Neural Networks and online learning: An overview and perspectives. *Neural Networks*, *121*, 88–100. https://doi.org/10.1016/j.neunet.2019.09.004

Lobo, J. L., Del Ser, J., Bifet, A., & Kasabov, N. (2020b). Spiking Neural Networks and online learning: An overview and perspectives. *Neural Networks*, *121*, 88–100. https://doi.org/10.1016/j.neunet.2019.09.004

Lobo, J. L., Laña, I., Del Ser, J., Bilbao, M. N., & Kasabov, N. (2018). Evolving Spiking Neural Networks for online learning over drifting data streams. *Neural Networks*, *108*, 1–19. https://doi.org/10.1016/j.neunet.2018.07.014

Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmén, P., Engström, M., & Elfsberg Dohns, I. (1994). Psychophysiological stress and emg activity of the trapezius muscle. *International Journal of Behavioral Medicine*, *1*(4). https://doi.org/10.1207/s15327558ijbm0104_5

Lystad, R. P., & Pollard, H. (2009). Functional neuroimaging: a brief overview and feasibility for use in chiropractic research. *The Journal of the Canadian Chiropractic Association*, *53*(1), 59–72. http://www.ncbi.nlm.nih.gov/pubmed/19421353%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2652631

M.Abeles. (1991). *Corticonics: Neural Circuits of the Cerebral Cortex*.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560. https://doi.org/10.1162/089976602760407955

Makeig, S., & Al., E. (2000). *ICA Toolbox for Psychophysiological Research (version 3.4)*.

https://sccn.ucsd.edu/~scott/ica-download-form.html

Marimont, R. B., & Shapiro, M. B. (1979). Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)*, *24*(1), 59–70. https://doi.org/10.1093/imamat/24.1.59

Markova, V., & Ganchev, T. (2018). Automated Recognition of Affect and Stress evoked by Audio-Visual stimuli. *2018 7th Balkan Conference on Lighting, BalkanLight 2018 - Proceedings*. https://doi.org/10.1109/BALKANLIGHT.2018.8546887

Markram, H., Gerstner, W., & Sjöström, P. J. (2011). A history of spike-timing-dependent plasticity. In *Frontiers in Synaptic Neuroscience*. https://doi.org/10.3389/fnsyn.2011.00004

Masquelier, T., & Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, *3*(2). https://doi.org/10.1371/journal.pcbi.0030031

Mateos-Aparicio, P., & Rodríguez-Moreno, A. (2019). The impact of studying brain plasticity. *Frontiers in Cellular Neuroscience*, *13*, 66. https://doi.org/10.3389/FNCEL.2019.00066/BIBTEX

MATLAB. (2019). *9.7.0.1190202 (R2019a)*. The MathWorks Inc.

McCarthy, J. (1963). Programs with common sense. *Proceedings of the Symposium on the Mechanization of Thought Processes*, 1–15. http://www-formal.stanford.edu/jmc/mcc59.ps%5Cnhttp://aitopics.org/sites/default/files/classic/Teddington Conference/Teddington-1.3-McCarthy.pdf

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

McKennoch, S., Dingding, L., & Bushnell, L. G. (2006). Fast modifications of the SpikeProp algorithm. *IEEE International Conference on Neural Networks - Conference Proceedings*. https://doi.org/10.1109/ijcnn.2006.246918

Mellor, J., Turner, J., Storkey, A., & Crowley, E. J. (2020). Neural Architecture Search without Training. *ArXiv*. http://arxiv.org/abs/2006.04647

Milakov, M., & Devtech, S. H. P. C. (2014). *Deep Learning Nvidia*. https://www.nvidia.co.uk/docs/IO/147844/Deep-Learning-With-GPUs-MaximMilakov-NVIDIA.pdf

Minku, L. L., & Yao, X. (2012). DDD: A new ensemble approach for dealing with concept drift. *IEEE Transactions on Knowledge and Data Engineering*, *24*(4). https://doi.org/10.1109/TKDE.2011.58

Mohemmed, A., Schliebs, S., Matsuda, S., & Kasabov, N. (2012). Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International Journal of Neural Systems*, *22*(4). https://doi.org/10.1142/S0129065712500128

Mohemmed, A., Schliebs, S., Matsuda, S., & Kasabov, N. (2013). Training spiking neural networks to associate spatio-temporal input-output spike patterns. *Neurocomputing*, *107*. https://doi.org/10.1016/j.neucom.2012.08.034

Moreno-Jiménez, E. P., Flor-García, M., Terreros-Roncal, J., Rábano, A., Cafini, F., Pallas-Bazarra, N., Ávila, J., & Llorens-Martín, M. (2019). Adult hippocampal neurogenesis is abundant in

neurologically healthy subjects and drops sharply in patients with Alzheimer's disease. In *Nature Medicine* (Vol. 25, Issue 4, pp. 554–560). Nature Publishing Group. https://doi.org/10.1038/s41591-019-0375-9

Morris, R. G. M. (1999). D.O. Hebb: The Organization of Behavior, Wiley: New York; 1949. In *Brain Research Bulletin* (Vol. 50, Issues 5–6, p. 437). https://doi.org/10.1016/S0361-9230(99)00182-3

Mozzachiodi, R., & Byrne, J. H. (2010). More than synaptic plasticity: role of nonsynaptic plasticity in learning and memory. *Trends in Neurosciences*, *33*(1), 17–26. https://doi.org/10.1016/j.tins.2009.10.001

Naud, R., & Gerstner, W. (2012). The performance (and limits) of simple neuron models: Generalizations of the leaky integrate-and-fire model. In *Computational Systems Neurobiology*. https://doi.org/10.1007/978-94-007-3858-4_6

Navlakha, S., Bar-Joseph, Z., & Barth, A. L. (2018). Network Design and the Brain. *Trends in Cognitive Sciences*, *22*(1), 64–78. https://doi.org/10.1016/j.tics.2017.09.012

O'Connor, D. B., Thayer, J. F., & Vedhara, K. (2021). Stress and Health: A Review of Psychobiological Processes. *Annual Review of Psychology*, *72*(1), 663–688. https://doi.org/10.1146/annurev-psych-062520-122331

Panda, P., & Roy, K. (2016). Unsupervised regenerative learning of hierarchical features in Spiking Deep Networks for object recognition. *Proceedings of the International Joint Conference on Neural Networks*, *2016-Octob*, 299–306. https://doi.org/10.1109/IJCNN.2016.7727212

Pardey, J., Roberts, S., & Tarassenko, L. (1996). A review of parametric modelling techniques for EEG analysis. *Medical Engineering & Physics*, *18*(1), 2–11. https://doi.org/10.1016/1350-4533(95)00024-0

Peter R., H. (1979). Synaptic density in human frontal cortex — Developmental changes and effects of aging. *Brain Research*, *163*(2), 195–205. https://doi.org/10.1016/0006-8993(79)90349-4

Petro, B., Kasabov, N., & Kiss, R. M. (2019). Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/tnnls.2019.2906158

Pfeiffer, M., & Pfeil, T. (2018). Deep Learning With Spiking Neurons: Opportunities and Challenges. *Frontiers in Neuroscience*, *12*(October). https://doi.org/10.3389/fnins.2018.00774

Pham, D. T., Packianather, M. S., & Charles, E. Y. A. (2008). Control chart pattern clustering using a new self-organizing spiking neural network. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *222*(10). https://doi.org/10.1243/09544054JEM1054

Pickering, T. G. (2001). Mental stress as a causal factor in the development of hypertension and cardiovascular disease. *Current Hypertension Reports*, *3*(3), 249–254. https://doi.org/10.1007/s11906-001-0047-1

Pomer-Escher, A. G., de Souza, M. D. P., & Filho, T. F. B. (2014). Methodology for analysis of stress level based on asymmetry patterns of alpha rhythms in EEG signals. *5th ISSNIP-IEEE Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC)*, 1–5. https://doi.org/10.1109/BRC.2014.6880963

Ponulak, F., & Kasiński, A. (2010). Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. In *Neural Computation*. https://doi.org/10.1162/neco.2009.11-08-901

Premkumar, P., Dunn, A. K., Onwumere, J., & Kuipers, E. (2019). Sensitivity to criticism and praise predicts schizotypy in the non-clinical population: The role of affect and perceived expressed emotion. *European Psychiatry*, *55*, 109–115. https://doi.org/10.1016/j.eurpsy.2018.10.009

Rakic, P., Bourgeois, J.-P., Eckenhoff, M. F., Zecevic, N., & Goldman-Rakic, P. S. (1986). Concurrent Overproduction of Synapses in Diverse Regions of the Primate Cerebral Cortex. *Science*, *232*(4747), 232–235. https://doi.org/10.1126/science.3952506

Rathi, N., Panda, P., & Roy, K. (2019). STDP-Based Pruning of Connections and Weight Quantization in Spiking Neural Networks for Energy-Efficient Recognition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *38*(4), 668–677. https://doi.org/10.1109/TCAD.2018.2819366

Reisman, S. (1997). Measurement of physiological stress. *Proceedings of the IEEE 23rd Northeast Bioengineering Conference*, 21–23. https://doi.org/10.1109/NEBC.1997.594939

Rekabdar, B., Nicolescu, M., Kelley, R., & Nicolescu, M. (2014). Unsupervised Learning of Spatio-temporal Patterns Using Spike Timing Dependent Plasticity. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 8598 LNAI* (pp. 254–257). https://doi.org/10.1007/978-3-319-09274-4_28

Roberts, P. D., & Leen, T. K. (2010). Anti-Hebbian Spike-Timing-Dependent Plasticity and Adaptive Sensory Processing. *Frontiers in Computational Neuroscience*, *4*(December), 1–11. https://doi.org/10.3389/fncom.2010.00156

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. https://doi.org/10.1037/h0042519

Roth, K. A., & D'Sa, C. (2001). Apoptosis and brain development. *Mental Retardation and Developmental Disabilities Research Reviews*, *7*(4), 261–266. https://doi.org/10.1002/mrdd.1036

Roy, K., Jaiswal, A., & Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature*, *575*(7784), 607–617. https://doi.org/10.1038/s41586-019-1677-2

Roy, S., & Basu, A. (2017). An Online Unsupervised Structural Plasticity Algorithm for Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(4), 900–910. https://doi.org/10.1109/TNNLS.2016.2582517

Ruf, B., & Schmitt, M. (1997). Learning Temporally Encoded Patterns in Networks of Spiking Neurons. *Neural Processing Letters*, *5*(1). https://doi.org/10.1023/A:1009697008681

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2013). Learning Internal Representations by Error Propagation. In *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence* (pp. 399–421). Elsevier. https://doi.org/10.1016/B978-1-4832-1446-7.50035-2

Saeed, S. M. U., Anwar, S. M., Khalid, H., Majid, M., & Bagci, U. (2020). EEG based classification of long-term stress using psychological labeling. *Sensors (Switzerland)*, *20*(7), 1–15. https://doi.org/10.3390/s20071886

Saidatul, A., Paulraj, M. P., Yaacob, S., & Yusnita, M. A. (2011). Analysis of EEG signals during

relaxation and mental stress condition using AR modeling techniques. *Proceedings - 2011 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2011*, 477–481. https://doi.org/10.1109/ICCSCE.2011.6190573

Sanes, J. N. (2003). Neocortical mechanisms in motor learning. In *Current Opinion in Neurobiology* (Vol. 13, Issue 2, pp. 225–231). Curr Opin Neurobiol. https://doi.org/10.1016/S0959-4388(03)00046-1

Savin, C., Joshi, P., & Triesch, J. (2010). Independent component analysis in spiking neurons. *PLoS Computational Biology*, *6*(4), e1000757. https://doi.org/10.1371/journal.pcbi.1000757

Savran, & Arman. (2006). Proceedings Chapter Reference Emotion Detection in the Loop from Brain Signals and Facial Images. *Proceedings of the ENTERFACE 2006 Workshop*, 205–218. https://archive-ouverte.unige.ch/unige:47926%0Ahttp://archive-ouverte.unige.ch/unige:47926

Sawangjai, P., Hompoonsup, S., Leelaarporn, P., Kongwudhikunakorn, S., & Wilaiprasitporn, T. (2020). Consumer Grade EEG Measuring Sensors as Research Tools: A Review. *IEEE Sensors Journal*, *20*(8), 3996–4024. https://doi.org/10.1109/JSEN.2019.2962874

Schliebs, S., & Kasabov, N. (2013). Evolving spiking neural network—a survey. *Evolving Systems*, *4*(2), 87–98. https://doi.org/10.1007/s12530-013-9074-9

Schrauwen, B., & Van Campenhout, J. (2003). BSA, a Fast and Accurate Spike Train Encoding Scheme. *Proceedings of the International Joint Conference on Neural Networks*, *4*, 2825–2830. https://doi.org/10.1109/IJCNN.2003.1224019

Seger, C. A., & Miller, E. K. (2010). Category learning in the brain. *Annual Review of Neuroscience*, *33*, 203–219. https://doi.org/10.1146/annurev.neuro.051508.135546

Sejnowski, T. J., & Poggio, T. A. (2007). Dynamical Systems in Neuroscience. In *Dynamical Systems* (Vol. 25).

Selye, H. (1965). The Stress Syndrome. *The American Journal of Nursing*, *65*(3), 97. https://doi.org/10.2307/3453119

Seo, S.-H., & Lee, J.-T. (2010). Stress and EEG. In *Convergence and Hybrid Information Technologies* (Issue March). InTech. https://doi.org/10.5772/9651

Shannon, C. (2009). Presentation of a MazeSolving MachineTransactions 8th Cybernetics Conference, Josiah Macy Jr. Foundation, 1952. In *Claude E. Shannon* (pp. 173–180). IEEE. https://doi.org/10.1109/9780470544242.ch44

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(4), 623–656. https://doi.org/10.1002/j.1538-7305.1948.tb00917.x

Shekhar, S., Jiang, Z., Ali, R., Eftelioglu, E., Tang, X., Gunturi, V., & Zhou, X. (2015). Spatiotemporal Data Mining: A Computational Perspective. *ISPRS International Journal of Geo-Information*, *4*(4), 2306–2338. https://doi.org/10.3390/ijgi4042306

Sherrington, D., & Kirkpatrick, S. (1975). Solvable Model of a Spin-Glass. *Physical Review Letters*, *35*(26), 1792–1796. https://doi.org/10.1103/PhysRevLett.35.1792

Shi, Y., Nguyen, L., Oh, S., Liu, X., & Kuzum, D. (2019). A Soft-Pruning Method Applied During Training of Spiking Neural Networks for In-memory Computing Applications. *Frontiers in*

Neuroscience, *13*(APR), 1–13. https://doi.org/10.3389/fnins.2019.00405

Shon, D., Im, K., Park, J.-H., Lim, D.-S., Jang, B., & Kim, J.-M. (2018). Emotional Stress State Detection Using Genetic Algorithm-Based Feature Selection on EEG Signals. *International Journal of Environmental Research and Public Health*, *15*(11), 2461. https://doi.org/10.3390/ijerph15112461

Shors, T. J., Anderson, M. L., Curlik, D. M., & Nokia, M. S. (2012). Use it or lose it: How neurogenesis keeps the brain fit for learning. In *Behavioural Brain Research* (Vol. 227, Issue 2, pp. 450–458). NIH Public Access. https://doi.org/10.1016/j.bbr.2011.04.023

Shrestha, S. B., & Song, Q. (2015). Adaptive learning rate of SpikeProp based on weight convergence analysis. *Neural Networks*, *63*. https://doi.org/10.1016/j.neunet.2014.12.001

Silva, S. M., & Ruano, A. E. (2006). Application of the Levenberg-Marquardt method to the training of spiking neural networks. *IEEE International Conference on Neural Networks - Conference Proceedings*. https://doi.org/10.1109/ijcnn.2006.246919

Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, *3*(9), 919–926. https://doi.org/10.1038/78829

Spiess, R., George, R., Cook, M., & Diehl, P. U. (2016). Structural Plasticity Denoises Responses and Improves Learning Speed. *Frontiers in Computational Neuroscience*, *10*(SEP), 1–13. https://doi.org/10.3389/fncom.2016.00093

Sporea, I., & Grüning, A. (2013). Supervised learning in multilayer spiking neural networks. In *Neural Computation* (Vol. 25, Issue 2). https://doi.org/10.1162/NECO_a_00396

Stemmler, M., & Koch, C. (1999). How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nature Neuroscience*, *2*(6), 521–527. https://doi.org/10.1038/9173

Storn, R., & Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11(4)*, 341–359. https://doi.org/10.1023/A:1008202821328

Subhani, A. R., Mumtaz, W., Saad, M. N. B. M., Kamel, N., & Malik, A. S. (2017). Machine learning framework for the detection of mental stress at multiple levels. *IEEE Access*, *5*, 13545–13556. https://doi.org/10.1109/ACCESS.2017.2723622

Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., & McGinnity, T. M. (2020). A review of learning in biologically plausible spiking neural networks. *Neural Networks*, *122*, 253–272. https://doi.org/10.1016/j.neunet.2019.09.036

Taherkhani, A., Belatreche, A., Li, Y., & Maguire, L. (2014). A new biologically plausible supervised learning method for spiking neurons. *22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2014 - Proceedings*.

Talairach, J., & Tournoux, P. (1988). Co-Planar Stereotaxis Atlas of the Human Brain: : an approach to cerebral imaging. In *Direct* (Vol. 270). Thieme Medical Publisher, Inc.

Tan, C., Ceballos, G., Kasabov, N., & Puthanmadam Subramaniyam, N. (2020). FusionSense: Emotion Classification Using Feature Fusion of Multimodal Data and Deep Learning in a Brain-Inspired Spiking Neural Network. *Sensors*, *20*(18), 5328. https://doi.org/10.3390/s20185328

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural Networks*, *111*, 47–63. https://doi.org/10.1016/j.neunet.2018.12.002

Taylor, D., Scott, N., Kasabov, N., Capecci, E., Tu, E., Saywell, N., Chen, Y., Hu, J., & Hou, Z.-G. (2014). Feasibility of NeuCube SNN architecture for detecting motor execution and motor intention for use in BCIapplications. *2014 International Joint Conference on Neural Networks (IJCNN)*, 3221–3225. https://doi.org/10.1109/IJCNN.2014.6889936

Thorpe, S., Delorme, A., & Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*. https://doi.org/10.1016/S0893-6080(01)00083-1

Thorpe, S., & Gautrais, J. (1998). Rank Order Coding. In *Computational Neuroscience* (pp. 113–118). Springer US. https://doi.org/10.1007/978-1-4615-4831-7_19

Thorpe, S. J., & Imbert, M. (1989). Biological constraints on connectionist modelling. In *Connectionism in perspective* (pp. 63–93). https://doi.org/10.1.1.96.6484

Triesch, J. (2007). Synergies between intrinsic and synaptic plasticity mechanisms. *Neural Computation*, *19*(4), 885–909. https://doi.org/10.1162/neco.2007.19.4.885

Tripathi, S., Acharya, S., Sharma, R., Mittal, S., & Bhattacharya, S. (2017). Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(2), 4746–4752. https://doi.org/10.1609/aaai.v31i2.19105

Tsodyks, M. (2002). Spike-timing-dependent synaptic plasticity - The long road towards understanding neuronal mechanisms of learning and memory. In *Trends in Neurosciences* (Vol. 25, Issue 12). https://doi.org/10.1016/S0166-2236(02)02294-4

Tu, E., Kasabov, N., & Yang, J. (2017). Mapping Temporal Variables Into the NeuCube for Improved Pattern Recognition, Predictive Modeling, and Understanding of Stream Data. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(6), 1305–1317. https://doi.org/10.1109/TNNLS.2016.2536742

Turney, S. G., & Lichtman, J. W. (2012). Reversing the outcome of synapse elimination at developing neuromuscular junctions in vivo: Evidence for synaptic competition and its mechanism. *PLoS Biology*, *10*(6), e1001352. https://doi.org/10.1371/journal.pbio.1001352

Turrigiano, G. G., & Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. In *Nature Reviews Neuroscience* (Vol. 5, Issue 2, pp. 97–107). https://doi.org/10.1038/nrn1327

Urbanowicz, R. J., & Moore, J. H. (2015). ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary Intelligence*, *8*(2–3), 89–116. https://doi.org/10.1007/s12065-015-0128-8

Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the Liquid State Machine: A case study. *Information Processing Letters*, *95*(6 SPEC. ISS.). https://doi.org/10.1016/j.ipl.2005.05.019

Wang, D., & Shang, Y. (2013). Modeling Physiological Data with Deep Belief Networks. *International Journal of Information and Education Technology (IJIET)*, *3*(5), 505–511. https://doi.org/10.7763/IJIET.2013.V3.326

Wang, G. Y., Premkumar, P., Lee, C. Q., & Griffiths, M. D. (2021). The Role of Criticism in Expressed Emotion Among Psychoactive Substance Users: an Experimental Vignette Study. *International Journal of Mental Health and Addiction*, 1–15. https://doi.org/10.1007/s11469-021-00591-2

Wang, J., Belatreche, A., Maguire, L., & McGinnity, T. M. (2014). An online supervised learning method for spiking neural networks with adaptive structure. *Neurocomputing*, *144*, 526–536. https://doi.org/10.1016/j.neucom.2014.04.017

Wang, J., Belatreche, A., Maguire, L. P., & McGinnity, T. M. (2017). SpikeTemp: An Enhanced Rank-Order-Based Learning Approach for Spiking Neural Networks with Adaptive Structure. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(1), 30–43. https://doi.org/10.1109/TNNLS.2015.2501322

Watt, A. J., & Desai, N. S. (2010). Homeostatic Plasticity and STDP: Keeping a Neuron's Cool in a Fluctuating World. *Frontiers in Synaptic Neuroscience*, *2*(JUN), 5. https://doi.org/10.3389/fnsyn.2010.00005

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. https://doi.org/10.1038/30918

Weerasinghe, M. M. A., Espinosa-Ramos, J. I., Wang, G. Y., & Parry, D. (2021a). Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling. *IEEE Access*, *9*, 117338–117348. https://doi.org/10.1109/ACCESS.2021.3099492

Weerasinghe, M. M. A., Espinosa-Ramos, J. I., Wang, G. Y., & Parry, D. (2021b). Incorporating Structural Plasticity Approaches in Spiking Neural Networks for EEG Modelling. *IEEE Access*, *9*, 117338–117348. https://doi.org/10.1109/ACCESS.2021.3099492

Weerasinghe, M. M. A., Parry, D., Wang, G., & Whalley, J. (2022). *Ensemble plasticity and network adaptability in SNNs*. http://arxiv.org/abs/2203.07039

Weerasinghe, M., Wang, G., & Parry, D. (2020). *Emotional Stress State Classification and Analysis using Spiking Neural Networks | Request PDF*. Affect, Personality and Embodied Brain 2020 Meeting. https://www.researchgate.net/publication/345128980_Emotional_Stress_State_Classification_and_Analysis_using_Spiking_Neural_Networks

Wegge, J., Vogt, J., & Wecking, C. (2007). Customer-induced stress in call centre work: A comparison of audio- and videoconference. *Journal of Occupational and Organizational Psychology*, *80*(4), 693–712. https://doi.org/10.1348/096317906X164927

Westman, J. C., & Walters, J. R. (1981). Noise and stress: a comprehensive approach. *Environmental Health Perspectives*, *41*, 291–309. https://doi.org/10.1289/ehp.8141291

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *ACM International Conference Proceeding Series*, 1–10. https://doi.org/10.1145/2601248.2601268

Wu, J., Feng, M., Liu, Y., Fang, H., & Duan, H. (2019). The relationship between chronic perceived stress and error processing: evidence from event-related potentials. *Scientific Reports*, *9*(1), 11605. https://doi.org/10.1038/s41598-019-48179-0

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2008). Fast and adaptive network of spiking neurons for multi-view visual pattern recognition. *Neurocomputing*, *71*(13–15), 2563–2575.

https://doi.org/10.1016/j.neucom.2007.12.038

Xu, B., Gong, Y., & Wang, B. (2013). Delay-induced firing behavior and transitions in adaptive neuronal networks with two types of synapses. *Science China Chemistry*, *56*(2). https://doi.org/10.1007/s11426-012-4710-y

Xu, T., Zhou, Y., Wang, Z., & Peng, Y. (2018). Learning Emotions EEG-based Recognition and Brain Activity: A Survey Study on BCI for Intelligent Tutoring System. *Procedia Computer Science*, *130*, 376–382. https://doi.org/10.1016/J.PROCS.2018.04.056

Yamaguchi, Y., & Miura, M. (2015). Programmed Cell Death in Neurodevelopment. *Developmental Cell*, *32*(4), 478–490. https://doi.org/10.1016/j.devcel.2015.01.019

Yuan, Y., Liu, J., Zhao, P., Xing, F., Huo, H., & Fang, T. (2019). Structural Insights Into the Dynamic Evolution of Neuronal Networks as Synaptic Density Decreases. *Frontiers in Neuroscience*, *13*. https://doi.org/10.3389/fnins.2019.00892

Zhang, A., Zhou, H., Li, X., & Zhu, W. (2019). Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism. *Neurocomputing*, *365*, 102–112. https://doi.org/10.1016/j.neucom.2019.07.009

Zhang, J. C., Lau, P. M., & Bi, G. Q. (2009). Gain in sensitivity and loss in temporal contrast of STDP by dopaminergic modulation at hippocampal synapses. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(31), 13028–13033. https://doi.org/10.1073/PNAS.0900546106/SUPPL_FILE/0900546106SI.PDF

Zhang, W., & Li, P. (2019). Information-Theoretic Intrinsic Plasticity for Online Unsupervised Learning in Spiking Neural Networks. *Frontiers in Neuroscience*, *13*(FEB), 1–14. https://doi.org/10.3389/fnins.2019.00031

Zhang, W., & Li, P. (2021). Skip-Connected Self-Recurrent Spiking Neural Networks With Joint Intrinsic Parameter and Synaptic Weight Training. *Neural Computation*, *33*(7), 1886–1913. https://doi.org/10.1162/neco_a_01393

Zhang, W., & Linden, D. J. (2003). The other side of the engram: Experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, *4*(11), 885–900. https://doi.org/10.1038/nrn1248

Zhang, Y., Wang, Q., Chin, Z. Y., & Keng Ang, K. (2020). Investigating different stress-relief methods using Electroencephalogram (EEG). *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2020-July*. https://doi.org/10.1109/EMBC44109.2020.9175900

Zuo, F., Panda, P., Kotiuga, M., Li, J., Kang, M., Mazzoli, C., Zhou, H., Barbour, A., Wilkins, S., Narayanan, B., Cherukara, M., Zhang, Z., Sankaranarayanan, S. K. R. S., Comin, R., Rabe, K. M., Roy, K., & Ramanathan, S. (2017). Habituation based synaptic plasticity and organismic learning in a quantum perovskite. *Nature Communications*, *8*(1). https://doi.org/10.1038/s41467-017-00248-6

Glossary

The key terms and their abbreviations used in this thesis is summarised in this glossary. The terms are in alphabetical order with a short description and the page number where the term was used first.

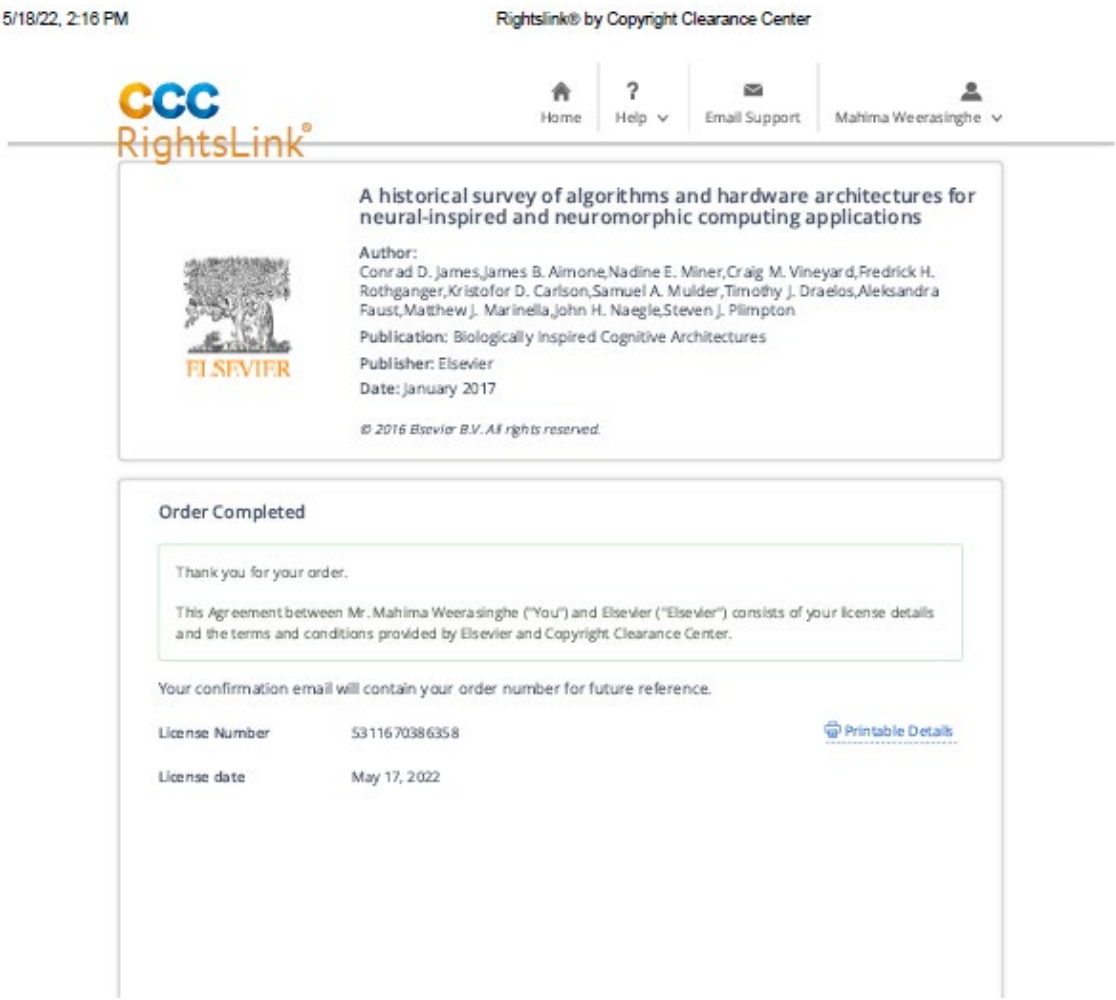
Term (Acronym)	Description	Page
AER	Address Event Representation. An information communication protocol found mostly in neuromorphic systems.	31
ANN	Artificial Neural Networks. A computational representation of the biological neuron used in groups for complex function approximation.	1
Apoptosis	Programmed cell death. A method used to remove cells in biological systems.	93
ASNN	Artificial Spiking Neural Networks. Another term used for Spiking Neural Networks when compared with biological neural networks.	1
ATB	Adaptive Threshold Based. An analogue to spike encoding method that adapts its threshold values that decide when to generate a spike.	35
AUTEC	Auckland University of Technology Ethics Committee.	138
Autocorrelation	The connectivity between the current time series data has towards its previously produced data, in discrete time case.	31
BCI	Brain-Computer Interface. The connectivity pathway for data between the computer and the brain.	13
BP	Backpropagation. Supervised learning methods used in ANNs to learn patterns by passing the error feedback.	12
BSA	Ben's Spike Algorithm. A temporal contrast, analogue to spike conversion technique.	35
CD	Coincidence Detectors. In terms of artificial neural networks, the neurons that detect coincidental spiking activity	24
CNN	Convolutional Neural Network. An Artificial Neural Network based on the convolution method. Generally used with image and video data.	50

Concept drift	The change of properties over time of a variable that the machine tries to predict.	23
CV	Cross-Validation. A validation method of machine learning models by resampling datasets into training and testing sets.	113
DBN	Deep Belief Network. A form of deep learning algorithm comprises stacked layers of Restricted Boltzmann Machines.	84
DE	Differential Evolution. An optimisation algorithm that iteratively improves a candidate solution	4
DEAP	A dataset for emotion analysis using electroencephalogram, physiological and video signals	42
deSNN	Dynamic evolving SNN. An algorithm that connects to a single output neuron and classifies inputs based on these connection weights.	29
DNN	Deep Neural Networks. An ANN with two or more hidden layers of neurons.	2
ED	Euclidean Distance. A mathematical distance measure between two points in an n-dimensional space.	39
EEG	Electroencephalogram. A neuroimaging technique that captures the electrical activity of the brain at the scalp level.	3
EEGLAB	A tool developed to process high-density EEG signals.	139
EOG	Electrooculogram. A variety of signals represent eye blinking.	42
FIR	Finite Impulse Response. A digital filter that produces an output for a finite duration in response to an input lasting for a finite duration	139
fMRI	Functional Magnetic Resonance Imaging. Neuroimaging method based on detecting the blood floor of the brain.	132
IAPS	International Affect Picture System. An image database for the analysis of emotions.	51
ICA	Independent Component Analysis. A signal processing technique used to separate signals into additive sub-components	102
Inion	The protuberance of the surface of the back of the skull, above the neck.	34

IoT	Internet of Things. Smart objects connected to the internet to deliver services.	2
IP	Intrinsic Plasticity. The change of properties inside the neuron in response to the passing signals.	92
ISI	Inter Spike Interval. The durational gap between two spikes.	102
Mastoids	Part of the temporal bone of the skull.	34
MEG	Magnetoencephalography. A neuroimaging technique based on the magnetic fields created by the electrical activity of the brain.	32
MNIST	Modified National Institute of Standards and Technology database. A base of hand-written digits.	94
Naison	The point of depression between the forehead and the nose.	34
NAS	Neural Architecture Search. A method developed for ANNs with non-spiking neurons to predict the most suitable structures for machine learning	122
Neurogenesis	The process of creating new neurons in the brain.	93
Occipital	A reference to the back of the skull.	34
On-the-fly learning	Learning and inferring at the same time.	154
PET	Positron Emission Tomography. A functional imaging technique used for measuring the metabolic activity of cells.	32
Snowballing	A method followed for studying research literature.	9
SP	Structural Plasticity. The change of structures in the brain in the process of learning.	65
SPECT	Single Photon Emission Computed Tomography. An imaging method to test the blood flow to tissues and organs.	32
ST/STD	Spatiotemporal Data. A type of data that carries both spatial and temporal information	3
Synaptogenesis	The elimination process of synapses in the brain.	93

Appendix – Copyright Letters for Figures

Figure 2.1 – Page 9



Licensed Content

Licensed Content Publisher	Elsevier
Licensed Content Publication	Biologically Inspired Cognitive Architectures
Licensed Content Title	A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications
Licensed Content Author	Conrad D. James,James B. Aimone,Nadine E. Miner,Craig M. Vineyard,Fredrick H. Rothganger,Kristofar D. Carlson,Samuel A. Mulder,Timothy J. Draelos,Aleksandra Faust,Matthew J. Marinella,John H. Naegle,Steven J. Plimpton
Licensed Content Date	Jan 1, 2017
Licensed Content Volume	19
Licensed Content Issue	n/a
Licensed Content Pages	16

Order Details

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No

About Your Work

Title	Neuromorphic Techniques for Machine Learning
Institution name	Auckland University of Technology
Expected presentation date	Aug 2022

Additional Data

Portions	Fig. 1. Historical timeline of neuroscience and psychology and the influence of the fields on neuromorphic and neural-inspired algorithms and hardware research. promise

Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Tax Details

Publisher Tax ID	GB 494 6272 12

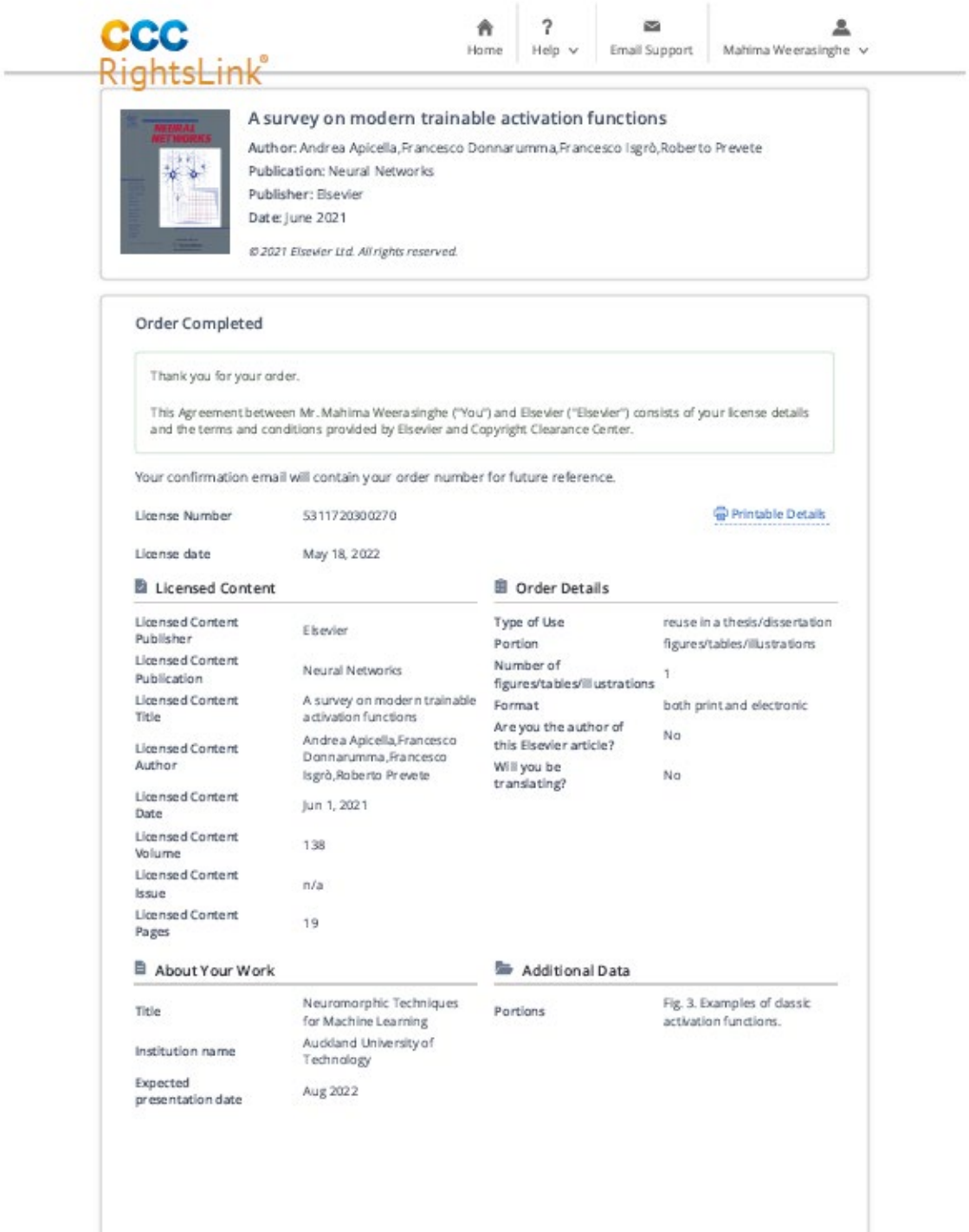
$ Price

Total	0.00 USD

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

Figure 2.3 – Page 13



CCC RightsLink

Home | Help | Email Support | Mahima Weerasinghe

A survey on modern trainable activation functions

Author: Andrea Apicella,Francesco Donnarumma,Francesco Isgrò,Roberto Prevete
Publication: Neural Networks
Publisher: Elsevier
Date: June 2021

Order Completed

Thank you for your order.

This Agreement between Mr. Mahima Weerasinghe ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

License Number	5311720300270
License date	May 18, 2022

Printable Details

Licensed Content

Licensed Content Publisher	Elsevier
Licensed Content Publication	Neural Networks
Licensed Content Title	A survey on modern trainable activation functions
Licensed Content Author	Andrea Apicella,Francesco Donnarumma,Francesco Isgrò,Roberto Prevete
Licensed Content Date	Jun 1, 2021
Licensed Content Volume	138
Licensed Content Issue	n/a
Licensed Content Pages	19

Order Details

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No

About Your Work

Title	Neuromorphic Techniques for Machine Learning
Institution name	Auckland University of Technology
Expected presentation date	Aug 2022

Additional Data

Portions	Fig. 3. Examples of classic activation functions.

Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Tax Details

Publisher Tax ID	GB 494 6272 12

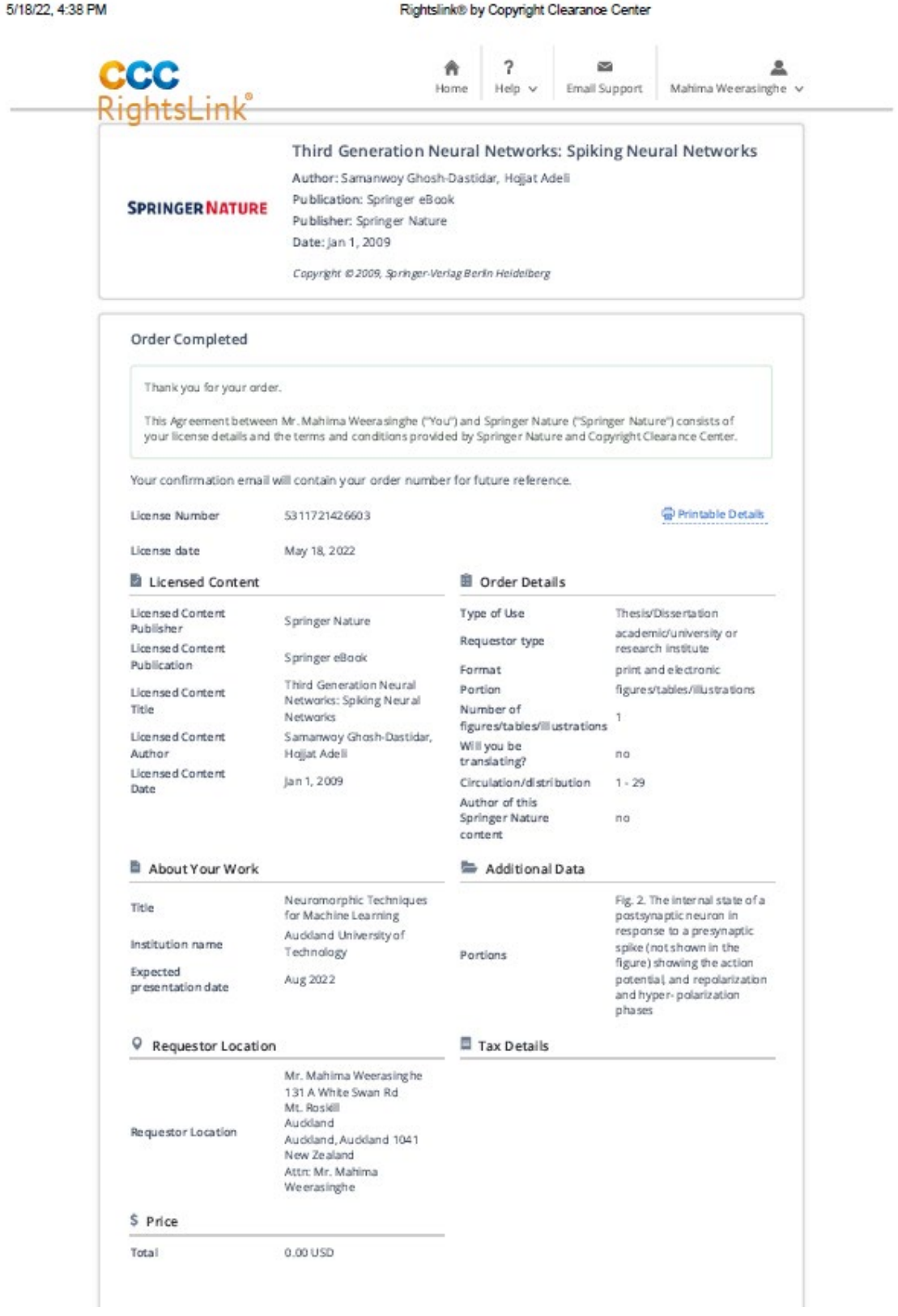
Price

Total	0.00 USD

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

Figure 2.4 – Page 14, Figure 2.5 – Page 15



5/18/22, 4:38 PM Rightslink® by Copyright Clearance Center

CCC RightsLink

Home | Help | Email Support | Mahima Weerasinghe

SPRINGER NATURE

Third Generation Neural Networks: Spiking Neural Networks

Author: Samanwoy Ghosh-Dastidar, Hojjat Adeli
Publication: Springer eBook
Publisher: Springer Nature
Date: Jan 1, 2009

Copyright © 2009, Springer-Verlag Berlin Heidelberg

Order Completed

Thank you for your order.

This Agreement between Mr. Mahima Weerasinghe ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

Printable Details

License Number	5311721426603
License date	May 18, 2022

Licensed Content

Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Third Generation Neural Networks: Spiking Neural Networks
Licensed Content Author	Samanwoy Ghosh-Dastidar, Hojjat Adeli
Licensed Content Date	Jan 1, 2009

Order Details

Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	no

About Your Work

Title	Neuromorphic Techniques for Machine Learning
Institution name	Auckland University of Technology
Expected presentation date	Aug 2022

Additional Data

Portions	Fig. 2. The internal state of a postsynaptic neuron in response to a presynaptic spike (not shown in the figure) showing the action potential and repolarization and hyper-polarization phases

Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Tax Details

$ Price

Total	0.00 USD

Figure 2.6 – Page 17, Figure 2.7 – Page 18

Figure 2.8 – Page 20

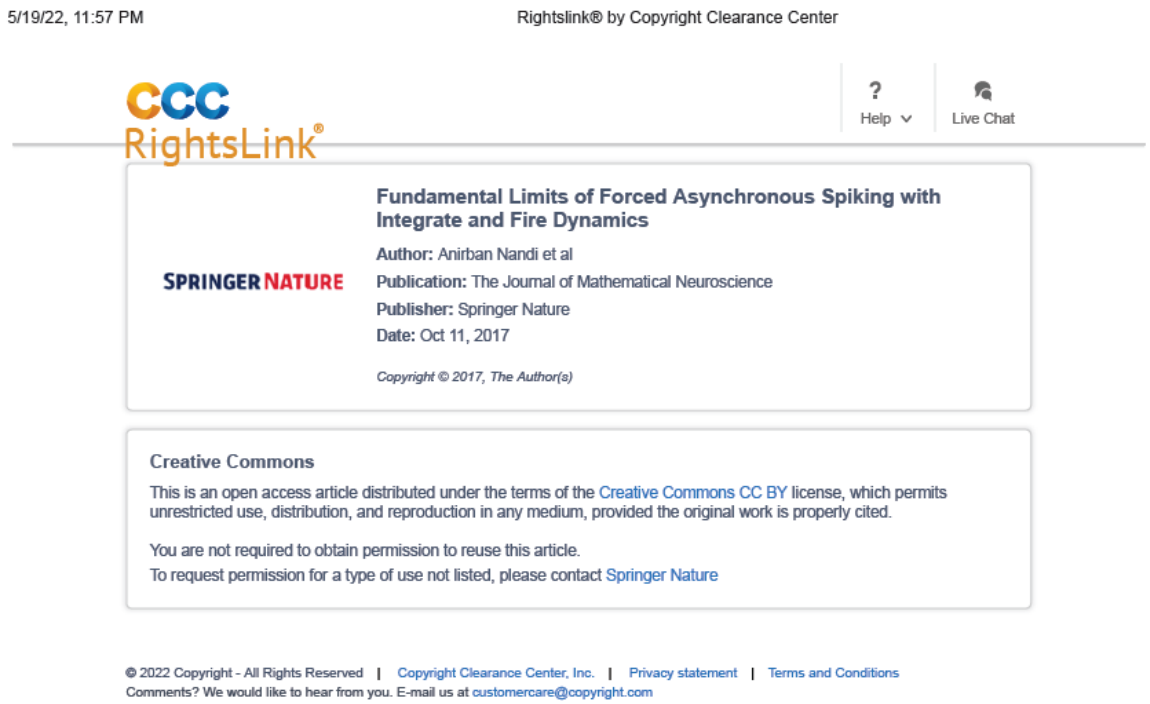
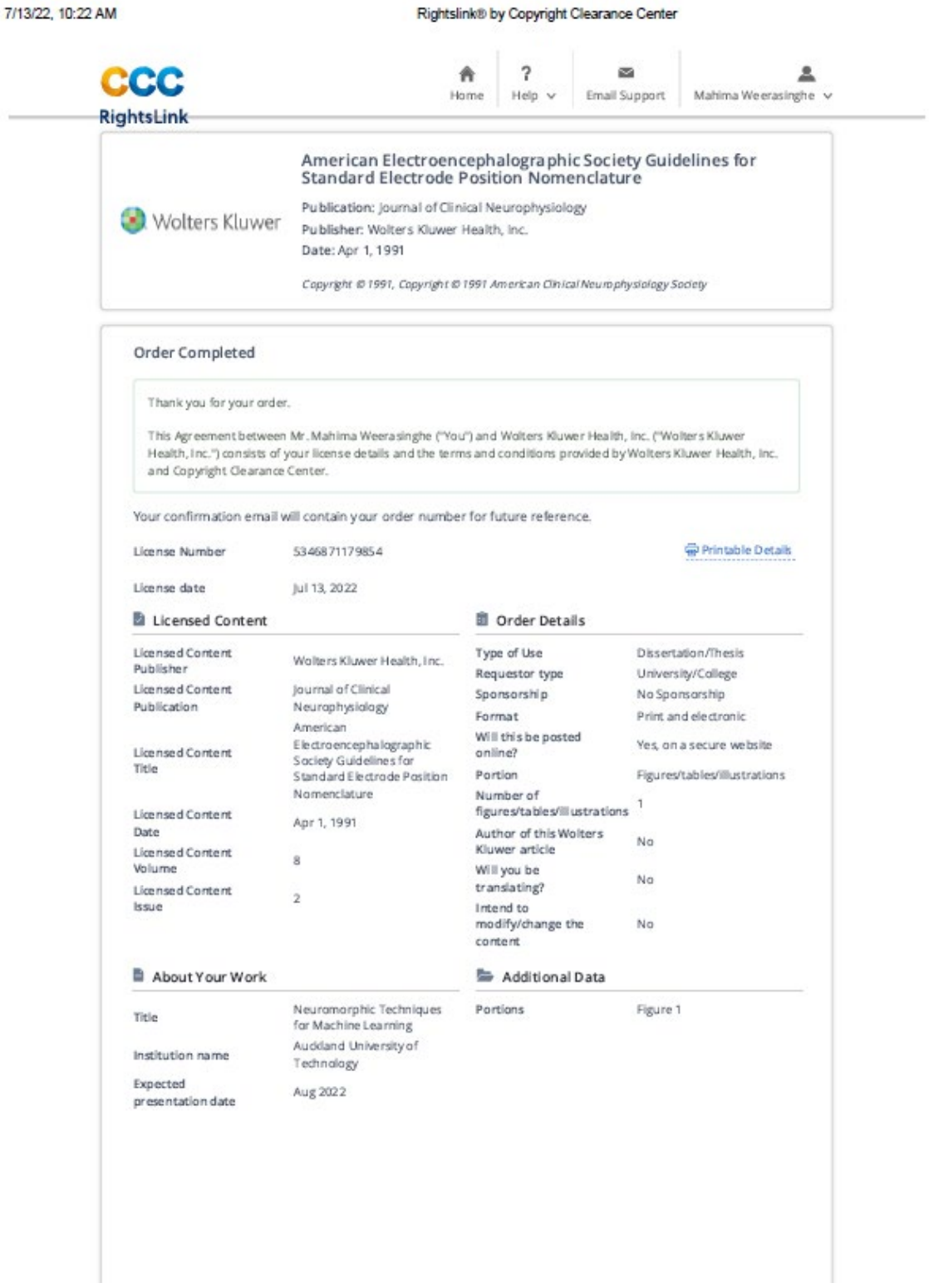


Figure 3.1 – Page 34



Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Price

Total	0.00 USD

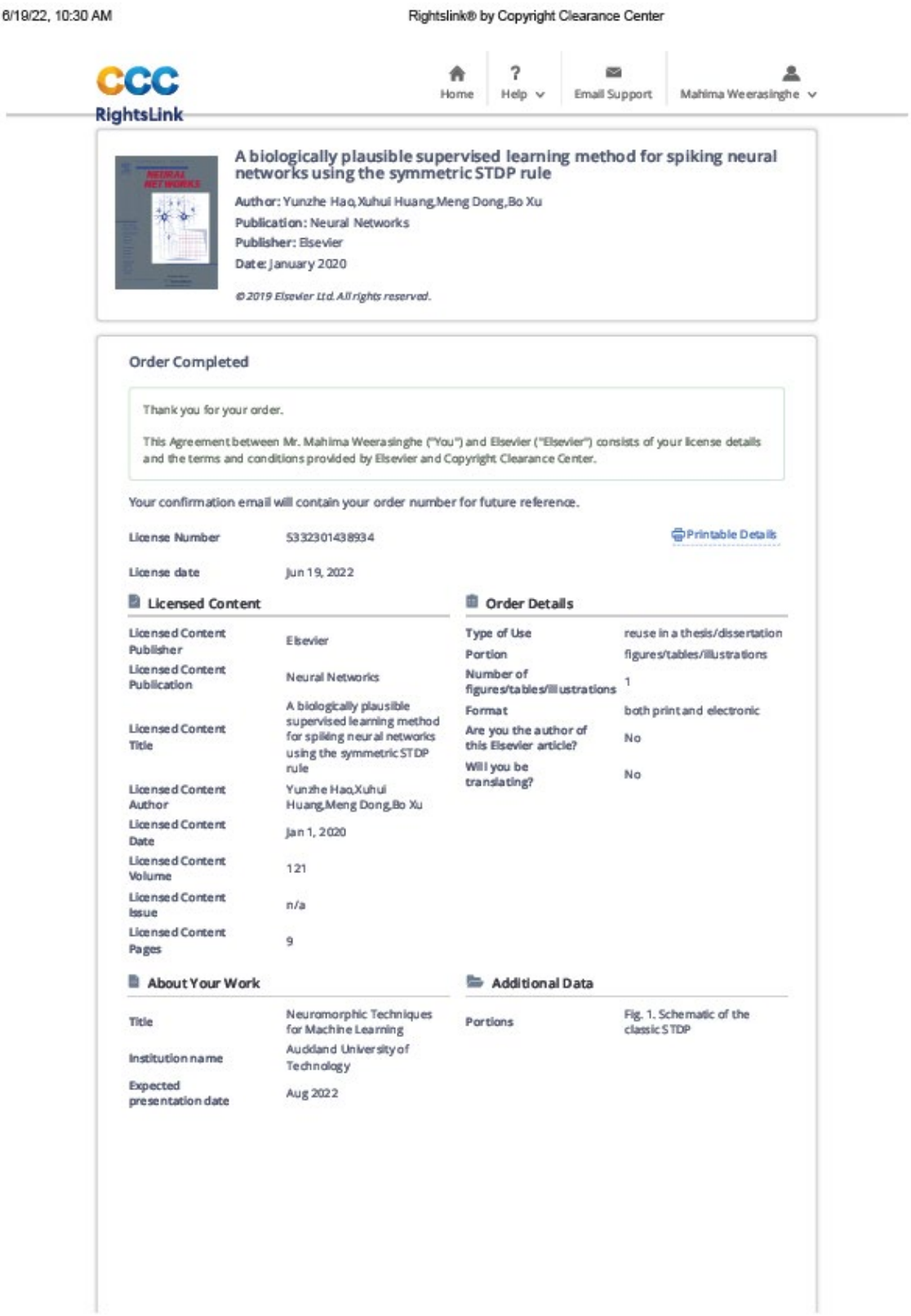
Tax Details

Publisher Tax ID	13-2932696

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

Figure 7.2 – Page 95



RightsLink

Home | Help | Email Support | Mahima Weerasinghe

A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule

Author: Yunzhe Hao,Xuhui Huang,Meng Dong,Bo Xu
Publication: Neural Networks
Publisher: Elsevier
Date: January 2020

Order Completed

Thank you for your order.

This Agreement between Mr. Mahima Weerasinghe ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

Printable Details

License Number	5332301438934
License date	Jun 19, 2022

Licensed Content

Licensed Content Publisher	Elsevier
Licensed Content Publication	Neural Networks
Licensed Content Title	A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule
Licensed Content Author	Yunzhe Hao,Xuhui Huang,Meng Dong,Bo Xu
Licensed Content Date	Jan 1, 2020
Licensed Content Volume	121
Licensed Content Issue	n/a
Licensed Content Pages	9

Order Details

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No

About Your Work

Title	Neuromorphic Techniques for Machine Learning
Institution name	Auckland University of Technology
Expected presentation date	Aug 2022

Additional Data

Portions	Fig. 1. Schematic of the classic STDP

Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Tax Details

Publisher Tax ID	GB 494 6272 12

Price

Total	0.00 USD

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

Figure 7.1 – Page 92

Licensed Content

Licensed Content Publisher	Elsevier
Licensed Content Publication	Biologically Inspired Cognitive Architectures
Licensed Content Title	A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications
Licensed Content Author	Conrad D. James,James B. Aimone,Nadine E. Miner,Craig M. Vineyard,Fredrick H. Rothganger,Kristofor D. Carlson,Samuel A. Mulder,Timothy J. Draelos,Aleksandra Faust,Matthew J. Marinella,John H. Naegle,Steven J. Plimpton
Licensed Content Date	Jan 1, 2017
Licensed Content Volume	19
Licensed Content Issue	n/a
Licensed Content Pages	16

Order Details

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No

About Your Work

Title	Neuromorphic Techniques for Machine Learning
Institution name	Auckland University of Technology
Expected presentation date	Aug 2022

Additional Data

Portions	Fig 2 on page 59

Requestor Location

Requestor Location	Mr. Mahima Weerasinghe 131 A White Swan Rd Mt. Roskill Auckland Auckland, Auckland 1041 New Zealand Attn: Mr. Mahima Weerasinghe

Tax Details

Publisher Tax ID	GB 494 6272 12

$ Price

Total	0.00 USD

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

The end.