# Self-Structuring Artificial Intelligence for Digital Equilibrium in Hyper-Connected Environments

by

Dinithi Sachintha Nallaperuma Jayaratne
M.Sc. 2015. University of Westminister
B.Sc. (Hons) 2010. University of Westminister

A thesis submission in total fulfilment

of the requirements for the degree of

**Doctor of Philosophy**

Research Centre for Data Analytics and Cognition
La Trobe Business School
College of Arts, Social Sciences and Commerce
La Trobe University
Victoria, Australia

February 2021

*To Amma, who always encourage me on every adventure*

*To my husband, who always be by my side with patience, love and friendship*

*To my children, Thinuk and Akenya, who made me stronger, better and more fulfilled than I could have ever imagined.*

# List of Publications

**Dinithi Nallaperuma,** Rashmika Nawaratne, Tharindu Bandaragoda, Achini Adikari, Su Nguyen, Thimal Kempitiya, Daswin De Silva, Damminda Alahakoon, & Dakshan Pothuhera. (2019). Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 1–12. https://doi.org/10.1109/TITS.2019.2924883

**Dinithi Nallaperuma,** Daswin De Silva, Damminda Alahakoon, & Xinghuo Yu. (2018). Intelligent Detection of Driver Behavior Changes for Effective Coordination Between Autonomous and Human Driven Vehicles. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3120–3125. https://doi.org/10.1109/IECON.2018.8591357

**Dinithi Nallaperuma**, Daswin De Silva, Damminda Alahakoon, & Xinghuo Yu. (2017). A cognitive data stream mining technique for context-aware IoT systems. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 4777–4782. https://doi.org/10.1109/IECON.2017.8216824

Tharindu Bandaragoda, Achini Adikari, Rashmika Nawaratne,, **Dinithi Nallaperuma**, Ashish Luhach, Thimal Kempitiya, Su Nguyen, Damminda Alahakoon, Daswin De Silva, Naveen Chilamkurti. (2020). Artificial Intelligence based commuter behavior profiling framework using Internet of Things for real time decision making. *Neural Computing and Applications*, 16057–1607. https://doi.org/10.1007/s00521-020-04736-7

Madhura Jayaratne, **Dinithi Nallaperuma**, Daswin De Silva, Damminda Alahakoon, Brian Devitt, Kate Webster, & Naveen Chilamkurti. (2019). A data integration platform for patient-centered e-healthcare and clinical decision support. *Future Generation Computer Systems*, 996-1008. https://doi.org/10.1016/j.future.2018.07.061

**Dinithi Nallaperuma**, & Daswin De Silva (2017). A Participatory Model for Multi-Document Health Information Summarisation. *Australasian Journal of Information Systems*, 1-15. https://doi.org/10.3127/ajis.v21i0.1393

# Table of Contents

List of Publications ........ 3

List of Tables ........ 8

List of Figures ........ 9

Abstract ........ 12

Declaration ........ 13

Acknowledgements ........ 14

**Chapter 1 ........ 15**

Introduction ........ 15

1.1 Hyper-Connected Digital Environment ........ 15

1.2 Research Motivation ........ 17

1.3 Research Questions ........ 18

1.4 Research Objectives ........ 19

1.5 Research Contributions ........ 20

1.6 Chapter Outline of the Thesis ........ 21

**Chapter 2 ........ 23**

Literature Review ........ 23

2.1 Hyper-Connected Digital Environments (HCDE) ........ 24

2.1.1 Understanding Complex Environments ........ 24

2.2 System Dynamics ........ 25

2.2.1 What is a Natural System? ........ 27

2.2.2 How do systems keep stability/equilibrium? ................................................ 31

2.2.3 Challenges of Systems .................................................................................. 32

2.2.4 Events and System Behaviour for Understanding the System Structure ......... 35

2.3 Related Work in Conventional Digital Environments ........................................ 37

2.3.1 Knowledge Discovery from Conventional Digital Environment ..................... 37

2.3.2 Research Problems and Challenges in Data Stream Analytics ........................ 38

2.3.3 Concept Change ............................................................................................. 40

2.3.4 Characterisation of Change Detection Methods ............................................. 44

2.4 Chapter Summary .................................................................................................. 53

**Chapter 3 ................................................................................................................ 55**

A Conceptual Model of Self-Structuring AI ................................................................ 55

3.1 Equilibrium in Nature ............................................................................................ 56

Tropical Cascade – "Wolfs Change Flow of Rivers"/Eco-System Equilibrium ............. 57

Homeostasis ................................................................................................................ 59

3.2 Equilibrium in Human Society .............................................................................. 60

Economic Equilibrium ................................................................................................ 60

3.3 Equilibrium in an HCDE ....................................................................................... 62

3.4 A Conceptual Model for Understanding Digital Equilibrium using for Self-Structuring AI (SS-AI) ................................................................................................ 63

3.5 Chapter Summary .................................................................................................. 70

**Chapter 4 ................................................................................................................ 71**

A Self-Structuring AI Algorithm for Change Detection ............................................. 71

4.1 Learning Paradigms of the Proposed Algorithm ................................................... 72

4.1.1 Incremental Learning ..... 72

4.1.2 Decremental Learning ..... 73

4.1.3 Online Learning ..... 73

4.2 Incremental Knowledge Acquisition and Self-Learning (IKASL) Algorithm ..... 74

4.3 The HCDE Context for the Proposed Algorithm ..... 78

4.4 Proposed Self-Structuring AI Algorithm for Concept Change Detection ..... 79

4.5 Demonstration ..... 84

4.5.1 Demonstration with Modified SEA dataset ..... 86

4.6 Chapter Summary ..... 88

**Chapter 5 ..... 89**

Empirical Evaluation of the Proposed Algorithm ..... 89

5.1 Air Traffic ..... 89

5.2 Smart Energy ..... 92

5.3 Physical Activity Monitoring Dataset ..... 93

5.4 Smart City Traffic Dataset ..... 97

5.5 Case Study: Detecting Changes in Motor Traffic in the Arterial Road Network of Victoria, Australia ..... 103

5.5.1 Traffic Flow Modelling ..... 105

5.6 Case Study: Detecting Change in Driving Behaviours of Autonomous and Human Driven Vehicles ..... 108

5.7 Chapter Summary ..... 116

**Chapter 6 ..... 117**

A Self-Structuring AI Algorithm for Concept Change Causality ..... 117

6.1 Identifying Causal Relationships between Multiple Data Streams ........................ 118

6.2 Related Work in Change Detection Causality ...................................................... 119

6.3 A Self-Structuring AI Algorithm for Concept Change Causality .......................... 120

6.3.1 Concept change detection algorithm for detection of evolving concepts ...... 121

6.3.2 Generalised Suffix Trie for Exploration of Sequences of Behaviour ............ 124

6.3.3 Behaviour tree method for identification of causal relationships in multiple data streams ........................................................................................................ 128

6.4 Experiments and Results ........................................................................................ 129

6.4.1 Video Analytics for Sports ............................................................................. 129

6.4.2 Feature extraction from the video .................................................................. 130

6.4.3 Evaluation of Proposed Approach .................................................................. 131

6.4.4 Detection of Concept Changes ....................................................................... 131

6.4.5 PAMAP2 Physical Activity Monitoring Dataset ........................................... 136

6.5 Chapter Summary ................................................................................................... 140

**Chapter 7 .......................................................................................................................... 142**

Conclusion ........................................................................................................................... 142

7.1 Summary of Research Contributions ..................................................................... 143

7.2 Addressing the Research Questions ...................................................................... 145

7.3 Future Directions ..................................................................................................... 149

References ........................................................................................................................... 151

# List of Tables


Table 4-1 Automated Time Window Analysis ........ 87

Table 5-1 Electricity dataset: event mapping on concept change ........ 93

Table 5-2 Average heart rate increase........ 94

Table 5-3 Activity mapping for Concept Change Detection........ 96

Table 5-4 Activity mapping for Concept Change Detection........ 101

Table 5-5 Summary of recurrent concept drift detection ........ 108

Table 5-6 Intelligent Detection of Driver Behaviour Change........ 113

Table 5- 7 Precision, Recall and F1 of behaviour change detection ........ 115

Table 6-1 Explanation of the concept in each learning iteration........ 132

Table 6-2 Explanation of captured concept changes........ 138

# List of Figures


Figure 1.1 A hyper-connected digital environment. .......... 16

Figure 2.1 Traffic volume dynamics of traffic congestion in Accra .......... 26

Figure 2.2 Changes in Natural System.......... 28

Figure 2.3 A system structure .......... 30

Figure 2.4 Properties of a System .......... 32

Figure 2.5 Challenges of a system .......... 35

Figure 2.6 Overall view of a system .......... 36

Figure 2.7 Real Concept Change .......... 41

Figure 2.8 Virtual Concept Change .......... 42

Figure 2.9 A demonstration of concept change types .......... 43

Figure 2.10 Data management methods.......... 47

Figure 2.11 Forgetting mechanisms.......... 48

Figure 2.12 Concept change detection methods .......... 49

Figure 2.13 Adaptation methods .......... 51

Figure 2.14 Learning methods for concept change detection and adaptation .......... 53

Figure 3.1 Ecosystem feedback .......... 56

Figure 3.2 Trophic Cascade .......... 58

Figure 3.3 Conceptualization of a Digital System .......... 64

Figure 3.4 Conceptual model for understanding digital equilibrium using SS-AI .......... 66

Figure 3.5 Self-Structuring AI (SS-AI) for Change Detection in HCDE .......... 67

Figure 3.6 Self-Structuring AI (SS-AI) for Concept Change Causality .......... 69

Figure 4.1 The HCDE context for the proposed Self-Structuring AI algorithm for concept change detection.......... 78

Figure 4.2 Proposed Self-Structuring AI algorithm for concept change detection .......... 80

Figure 4.3 SEA dataset – Concept Change Detection.......... 84

Figure 4.4 Support Vector Machine Accuracy with and without Concept Drift Detection ..... 85

Figure 4.5 MOA (Bifet et al., 2010) output for modified SEA dataset ..... 86

Figure 4. 6 Concept Change Detection for Modified SEA dataset ..... 87

Figure 5.1 Airline dataset: Concept Change Detection ..... 90

Figure 5.2 Airline dataset: Input Variable Analysis ..... 91

Figure 5.3 Airline dataset: Target Variable Analysis ..... 91

Figure 5.4 Demonstration of concept change detection in Electricity dataset. Top: Concept change detection, Bottom: Hourly electricity usage (segment marked in concept change detection diagram): Sub_meter_1 – kitchen appliances (dishwasher, oven, and microwave), Sub_meter_2 – laundry room (washing machine, tumble dryer, refrigerator and, a light), Sub_meter_3 – electric water heater and air-conditioner ..... 92

Figure 5.5 Activity Dataset: Unsupervised Concept Change Detection ..... 95

Figure 5.6 Sammon's Mapping of Generalization Nodes ..... 97

Figure 5.7 Reoccurring concept change detection (CD). ..... 98

Figure 5.8 Abrupt concept change detection (CD) ..... 102

Figure 5.9 Data distribution of detected abrupt concept change ..... 102

Figure 5.10 Bluetooth traffic monitoring system (BTMS) ..... 104

Figure 5.11 Schematic of the road network in the area of interest selected for traffic analysis. ..... 106

Figure 5.12 Recurrent and non-recurrent concept change detection ..... 107

Figure 5.14 Fuller's Driver Demands and Capabilities Model (Fuller, 2005) ..... 111

Figure 5.15 Extended Driver Capacity and Driving Demands Model for behavior recognition and change detection ..... 112

Figure 5.16 Intelligent Detection of Driver Behaviour Change ..... 112

Figure 5.17 Explanation of the change of behaviour detected by the algorithm: Top left – truck entering the highway, Top right – the truck is close to entering the lane of the car, Bottom left – a sudden change of lane by the human driver, Bottom right – car returns to original lane to avoid likely collision ..... 115

Figure 6.1 Overview of the proposed approach for capturing the causal relationship between related data streams as a sequence of evolving concepts ...................................................... 120

Figure 6. 2 Proposed concept change detection algorithm for detection of evolving concepts using online, incremental and decremental learning ............................................................ 122

Figure 6.3 Exploration of dynamic behaviours using generalised suffix trie ........................ 125

Figure 6.4 Selection of a player for feature extraction.......................................................... 130

Figure 6.5 Overview of the proposed algorithm applied in the Volleyball dataset................ 131

Figure 6.6 Concept change detection in the Volleyball dataset ............................................ 132

Figure 6.7 Sequence of behaviour generated by Brazilian Player – Libero........................... 134

Figure 6.8 Sequence of behaviour generated by Brazilian Player – Setter ............................ 134

Figure 6.9 Causal relationship between the Brazil team players ........................................... 136

Figure 6.10 Concept change detection in the PAMAP2 Physical Activity Monitoring dataset ............................................................................................................................ 137

Figure 6.11 Sequence of behaviour in PAMAP2 Physical Activity Monitoring Dataset ...... 140

# Abstract

The soaring maturity of technological innovation and its rapid adoption by human societies has led to the orchestration of a novel Hyper-Connected Digital Environment (HCDE) that transcends the boundaries of natural, human-made, social, virtual and artificial environments. Human behaviours and activities are no longer confined to a single environment as our presence is perpetually intertwined within the expanse of an HCDE. The digital data representations generated by this HCDE can provide unique insights into the structures and functions of the individual environments, constituent features, atomic elements and the interplay across multiple layers. These unique insights translate into decisions and actions that drive value in organisations, economies, societies and communities. However, the inherent sophistication of data streams and data repositories of an HCDE pose several complex challenges that need to be addressed. Although conventional Artificial Intelligence has been effective in singular digital environments, the plurality of an HCDE requires a novel approach. This thesis presents such an approach, that is motivated and modelled on the theories of system dynamics and general equilibrium. The notion of 'digital equilibrium' is introduced and formalised as an extension of general equilibrium and an abstraction of the complexity of an HCDE. This novel approach is materialised in Self-Structuring Artificial Intelligence, with a specific focus on change detection and causality of change that drives the digital equilibrium of an HCDE. The design and development of two new machine learning algorithms for change detection and causality of change are further technical contributions of this thesis. Hierarchical depiction of influence, similarity and gradual causality, as well as an online, incremental and decremental learning functionality are unique features of these two algorithms. Both algorithms are empirically evaluated across diverse HCDE settings of air traffic, smart energy, smart city traffic, physical activity monitoring, and sport analytics, to demonstrate the effectiveness and practicality of Self-Structuring Artificial Intelligence in addressing the complexities of HCDE.

# Declaration

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgement in the main text of the thesis. This thesis has not been submitted for award of any degree or diploma in any other tertiary institute.

This work was supported in part by an Australian Government Research Training Program Scholarship and in part by the Data to Decisions Cooperative Research Centre (D2D CRC).

Dinithi Sachintha Nallaperuma Jayaratne

23-02-2021

# Acknowledgements

First and foremost, I would like to thank my supervisor, Associate Prof. Daswin De Silva for the continuous guidance and patience. You have given much time, effort and knowledge to aid in the completion of the thesis. I would like to thank my second supervisor, Prof. Damminda Alahakoon, for not only the invaluable inspiration, encouragement, and support but for the words of wisdom that enriched my life. I am also thankful for Dr. Su Nguyen for the insightful discussions and feedback that enhanced my work immensely.

I was fortunate to share the PhD journey with my best friend and dearest husband, Madhura, taking the same ride, from day one. It was fun to prepare for presentations, attend conferences and achieve milestones together whilst facing all the parenting challenges. Thank you for helping and encouraging me to in every challenge and cherishing my achievements. I cannot begin to express my gratitude to my kids, Thinuk and Akenya, for all the love and happiness you brought to my life. You are my inspiration to achieve greatness.

I would not have continued my PhD studies if it were not for my mom. Thank you *amma* for adapting to a foreign country to support me in taking care of our kids and for the continuous encouragement and endless love that kept me strong and motivated throughout. My heartfelt gratitude also goes to *ammi* and *thaththi*, my sisters *loku akka and podi akka*, sister-in-law *nangi*, brothers-in-laws *loku aiya, Viraj aiya and Kalana,* nephews *Kushan and Deshan* for the support, encouragement and love.

Finally, I am thankful for my colleagues, Achini, Rashmika, Tharindu, Nilu, Yujie and Qing for the friendships and making the PhD journey enjoyable. I have had many unforgettable experiences with them including attending conferences, working on research and tutoring as well as casual chats over lunch. Thank you very much for the friendships and the good times we shared.

# Chapter 1
# Introduction

This chapter introduces the thesis. It begins with an explication of Hyper-Connected Digital Environments (HCDE), followed by the research questions, research objectives and the research contributions. The chapter concludes with an outline of the subsequent chapters of this thesis.

## 1.1 Hyper-Connected Digital Environment

Human society is rapidly transitioning from the Information Age (Cortada, 2020) to an Age of Connections, where intelligent and automation technologies are interconnecting the animate and inanimate across physical, geographical, socio-demographic, behavioural, psychological and emotional boundaries. Social media platforms with billions of participants, gig economies of lifestyle convenience, smart mobility with driverless vehicles, robotic surgery using augmented reality and decentralised digital currencies are some of the prime technological manifestations of this Age of Connections.

The Information Age is depicted in the bottom half of Figure 1.1, where the environments that surround us, natural, human-made, virtual, social and artificial, are digitally represented. A digital representation of (a) natural environment captures and represents behaviours of living and non-living things occurring naturally, (b) human-made environment represents buildings, bridges, and roads, (c) virtual environment represents interaction via networks (such as email, chat, web-based document sharing platforms), (d) the social environment represents the interaction between people on various social media channels and (e) the artificial environment represents interactions between robots, chatbots and agents. These digital representations of different environments capture the corresponding entities' behaviours and interactions at high granularity, scale, and frequency.

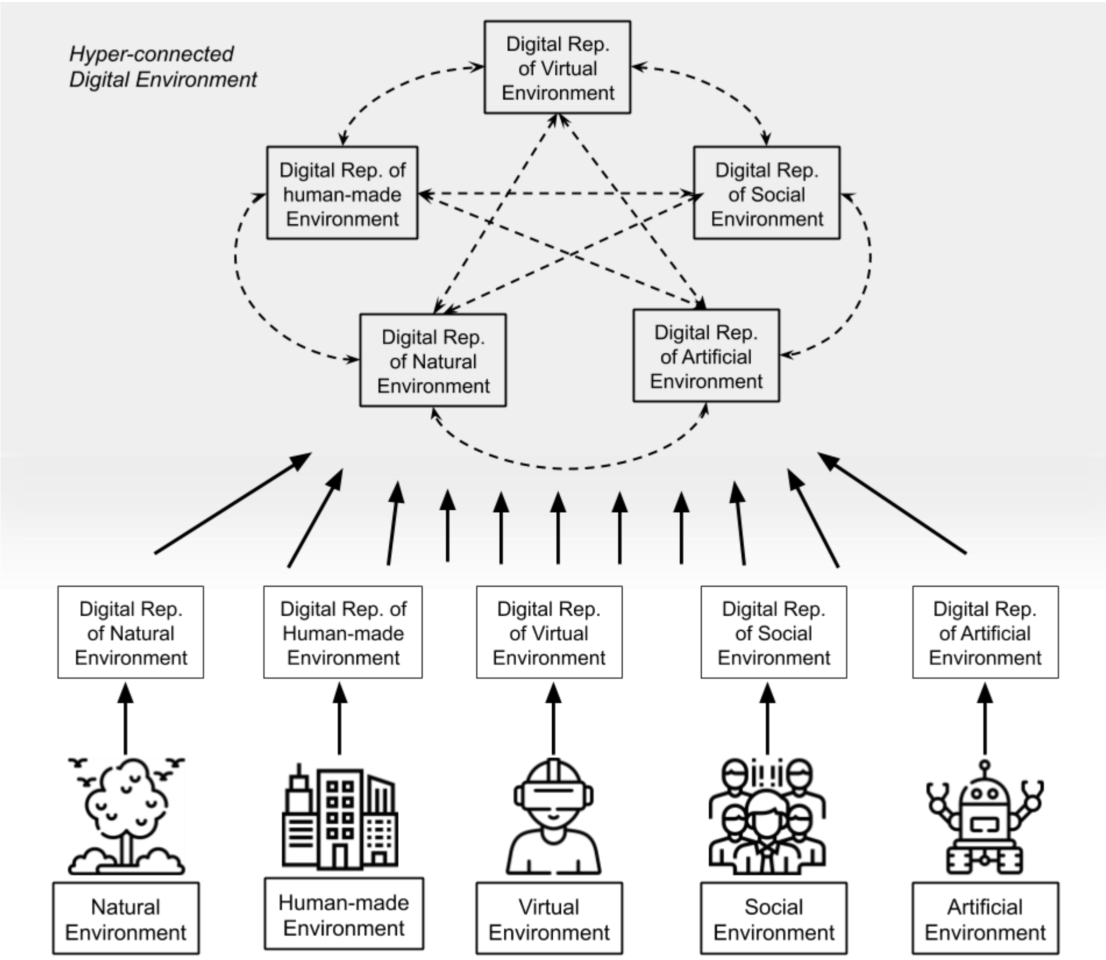



Figure 1.1 In the *Age of Information*, natural, human-made, virtual, social and artificial environments are represented in the digital environment. In the *Age of Connections*, those digital representations of different environments would be interacting with each other, creating a Hyper-Connected Digital Environment.

The Age of Connections has led to the conception of hyper-connectivity, where these digital environments (natural, human-made, virtual, social and artificial) that existed in siloes are interconnected through intelligent and automation technological innovations (top half of Figure 1.1). This thesis refers to this novel environment as Hyper-Connected Digital Environment (HCDE). In an HCDE, service provision, resource utilisation and personalisation will be hyper-optimised to suit each individual's needs and expectations. For instance, the movement of individuals and groups will be optimised in a social setting based on data from autonomous vehicles scheduled on entries in electronic diaries and real-time conversations on email or social media. In a smart city setting, pedestrian and vehicular movement will be optimised based on congestion, public events and the weather. Drawing on this context of an HCDE, let us now explore the research motivation of this thesis.

## 1.2 Research Motivation

As explicated above, an HCDE is a network of environments that surround us - natural, social, virtual, human-made and artificial. Although these environments are inherently complex and systems within these environments are quite different, they demonstrate similar properties at a high level (Mitchell, 2011). These properties are (1) complex collective behaviour that gives rise to the complex, hard to predict and changing patterns of behaviour, (2) signalling and information processing where the system in these environments use information and signals from both their internal and external environments and (3) adaptation to change their behaviour to improve their chance of survival or success through the learning of evolutionary processes (Johnson, 2009; Mitchell, 2011). These characterise unexpected events, both natural and human-made, and consequences that change the complex systems' structure and function. Even though these events and consequences are influenced by the interconnectivity between natural, social, virtual and artificial environments, every environment returns to stability, followed by equilibrium (Meadows, 2008). The digital representations of HCDE enable researchers to understand the complexities in this dynamic network of environments and analyse the events in the form of a coherent whole to provide actionable insights leading to informed decisions (S. K. Kim & Kim, 2020).

This thesis draws inspiration from natural equilibrium to comprehend the digital representations of an HCDE, based on the properties of self-organisation (organising and grouping underlying concepts), hierarchy (arranging objects or behaviours linked directly or indirectly) and resilience (the ability of the environment to adapt to changes or disturbances). However, understanding the environments is faced with the complexities of these environments, such as voluminous high-velocity data streams, non-linear relationships, non-existent boundaries, ubiquitous delays, and granularity layers. This thesis is motivated by the potential role of artificial intelligence to understand and address these complexities of an HCDE. An HCDE generates dynamic, unlabelled, and continuous data in structured and unstructured digital big data streams which can be better understood, analysed and synthesised using novel artificial intelligent algorithms that detect these changes and infer causality for such changes.

## 1.3 Research Questions

Motivated by the opportunities presented by artificial intelligence to understand and address the complexities of digital representations of an HCDE based on the notions of digital equilibrium, the high-level research question is:

**How can Artificial Intelligence orchestrate equilibrium in a Hyper-Connected Digital Environment?**

The primary research question is dissected into specific research questions as follows.

1. What factors of natural equilibrium stabilise a natural environment, and how are these factors represented in an HCDE?

2. How can these factors of equilibrium be used to design an artificial intelligence model capable of detecting changes that lead to disequilibrium and detecting the causality of such change in an HCDE?

3. How can unsupervised machine learning be advanced to develop new algorithms based on the artificial intelligence model designed in Question 2?

4. How can the algorithms developed in Question 3 be applied to address the practical challenges and complexities of real-world HCDE, demonstrated in use cases of smart cities, smart homes, digital health and sport?

## 1.4 Research Objectives

The first objective is to explore existing methods for understanding complex systems and identify the factors that influence these complex systems to remain stable. These factors will be used to develop a conceptual model for detecting change and the causality of changes in a complex environment.

The second objective is to design and develop an artificial intelligence approach for change detection in an HCDE and causality analysis of that change. This approach is based on Self-Structuring Artificial Intelligence (Self-Structuring AI). Self-Structuring AI is defined as intelligence structures that autonomously evolve based on the unstructured and unlabelled nature of data, spatially, temporally, laterally, and semantically (De Silva et al., 2020). This objective will explore the existing methods for change detection, investigate the limitations of those methods, and propose a novel Self-Structuring AI algorithm for change detection and causality analysis of that change in an HCDE.

The third objective is to evaluate the proposed machine learning technique in a wide range of real-world applications representative of HCDE. This objective aims to demonstrate the effectiveness, accuracy and utility of the proposed techniques in detecting change and causality of change in an HCDE, leading to digital equilibrium.

## 1.5 Research Contributions

The contributions of this thesis are:

1. A comprehensive review of the state of the art of HCDE, system dynamics, and machine learning techniques for concept change detection.
2. The design of a conceptual model for the detection of concept change and causality of change in HCDE.
3. Based on the conceptual model, the design and development of a Self-Structuring AI algorithm for concept change detection using three machine learning paradigms, online, incremental and decremental.
4. The design and development of a Self-Structuring AI algorithm for understanding causality of concept change in HCDE using a generalised suffix trie and a behaviour tree algorithm that identifies the causal relationships between multiple data streams.
5. Demonstration of the Self-Structuring AI algorithm for concept change detection using a widely-cited benchmark SEA dataset (Street & Kim, 2001).
6. Application of the concept change detection algorithm in real-life HCDE scenarios of air traffic, smart energy, smart city traffic and physical activity monitoring.
7. Application of both algorithms on an intelligent transport case study based on real-time data from the arterial road network of Victoria, in collaboration with VicRoads, the Victorian Roads Authority.
8. Application of the causality of change detection algorithm on physical activity monitoring dataset and video analytics for sports.

## 1.6 Chapter Outline of the Thesis

Chapter 2 presents the literature review on HCDE, system dynamics, and AI techniques for concept change detection. The chapter elaborates how theories of system dynamics are used in this thesis to understand the natural, social and human-made complex environments and the limitations of this approach. Further, the chapter provides an overall understanding of data streams and knowledge discovery from data streams. Moreover, the chapter provides a complete theoretical background of concept change.

Chapter 3 begins by describing several instances of natural equilibrium to understand how nature handles change. Following this delineation, theories of system dynamics are utilised to design a conceptual model to detect and understand the causality for concept change. The proposed model proposes a novel artificial intelligence technique based on self-structuring artificial intelligence capable of detecting changes that lead to disequilibrium and detecting the causality of such change in an HCDE.

Chapter 4 presents Self-Structuring AI algorithm for change detection. The chapter discusses in detail the learning paradigms of the proposed techniques; incremental, decremental and online. Then the chapter provides the algorithmic contribution followed by a demonstration of the technique with SEA benchmark dataset.

Chapter 5 presents the empirical evaluation of the proposed Self-Structuring AI algorithm for change detection proposed in chapter 4. The algorithm was applied to six scenarios that are representatives of an HCDE settings, including four real-world datasets in air traffic, smart energy, smart city traffic, physical activity monitoring, and two real-world case studies on the arterial road network of VicRoads and an annotated driving recordings of autonomous vehicles.

Chapter 6 presents Self-Structuring AI algorithm for concept change causality, based on several existing research on understanding a system's behaviour as sequences of behaviours and causal relationships between several data streams. The techniques are; generalised suffix trie for exploration of sequences of behaviour in data streams, behaviour tree method for identifying causal relationships between multiple data streams. The experiments were conducted on digital health dataset mentioned in chapter 5 and on a publicly available video stream of a volleyball game. The chapter provides a comprehensive view of the proposed Self-Structuring AI algorithm applied to understand digital equilibrium.

Chapter 7 concludes with a discussion of the research contributions in response to the research questions and directions for future work.

# Chapter 2
# Literature Review

The chapter aims to characterise and examine the constructs of a Hyper-Connected Digital Environment (HCDE) by studying the concepts and theories of complex environments. The chapter elaborates how theories of system dynamics are used to find a solution in the natural, social and human-made complex environments and the limitations of this approach. It presents a real-world scenario from a road traffic environment analysed using system dynamics, the study of information flows in the feedback relationships. The section delineates the challenges that make a system extremely complex and properties that maintain the stability of the system. The chapter further investigates how a natural system is sensed through events and how they accumulate into the system behaviour, which provides information on the underlying system structure. The chapter also provides a detailed literature study on knowledge discovery from data streams in HCDE and summarise research problems and challenges. A detailed discussion of theories of concept change is undertaken while two types of concept change and rate of change are presented. The existing work on concept change is discussed in line with data management, forgetting mechanisms, detection methods, adaption methods and learning methods.

## 2.1 Hyper-Connected Digital Environments (HCDE)

An HCDE is a conception of the hyper-connectivity of the Age of Connections where digital representations of the environments surrounding us (natural, human-made, virtual, social and artificial) are interconnected across their corresponding siloes through intelligent and automation technological innovations. In the current socio-technical domain, we have become increasingly connected physically, socially, and virtually due to improved information and communication technologies. Some of the applications showcase the opportunity to capture data in HCDE and generate analytical insights in real-time. For example (1) by applying analytical models in real-time smart city IoT infrastructure, (a) A city can improve the efficiency of urban systems such as highways and arterial roads. Real-time detection of traffic congestion and providing alerts to the commuters to take alternate routes identified by optimisation systems. (b) Optimise the power grid networks based on existing power demand and projections. (c) Monitor water systems for the detection of leaks and understanding the impact of water usage in the community. (d) Manage a wide range of infrastructures related to public safety and security, parking management, streetlight management. (2) Industry 4.0 connects smaller objects such as components within a machine on a production line (Shrouf et al., 2014). The manufacturing process becomes more effective due to gathering, analysing and utilising information from these objects. The dynamic interaction between people, organisations and businesses have given rise to a massive surge in complexity; hence, created complex systems which are difficult to interpret or predict (Cortada, 2020).

### 2.1.1 Understanding Complex Environments

By their very nature, complex systems have agents compete for limited resources, such as food, space, power, energy or wealth (Johnson, 2009). The complexity is created with a large number of interactions between these heterogeneous agents, where many of those interactions result in new phenomena (Johnson, 2009). Further, all complex systems exist within their environment and are part of that environment (Ball, 2012). As the environment changes, the adaptation affects the behaviour of the whole environment. The emergent behaviour of the whole cannot be construed by the sum of individual agents (Ball, 2012). For the easier explanation; if we look into a traffic example where cars (agents) consuming roads (limited resources) create waves of traffic jams (emergent behaviour). The traffic is not merely about the number and speed of cars but the roads, traffic lights and even potholes. Changes of those components will result in the traffic pattern to change, and the agents adapt to their environment resulting the system and its environment to co-evolve. Drivers of the cars stop the car by seeing the taillights of the car in front of them, meaning that the agents are interconnected and have non-linear interactions (Ball, 2012). This interconnectivity can be seen even within environments (Mitchell, 2011). For example, (1) a pandemic involves the interplay of people, viruses, travel, social interaction and medical care, (2) financial system constantly fluctuates due to the interaction between financial instruments, banks and the investor psychology. The major challenge of these complex systems is to recover quickly from the next unexpected event in our natural, social and artificial systems (Axelrod & Cohen, 1999).

## 2.2 System Dynamics

'System Dynamics' is the study of natural and human-made complex and dynamic systems (Springael & Kunsch, 2002). To understand how system dynamics theories can be applied in HCDE, let's look at a study that uses data from a digital representation of a human-made environment. This study has been conducted in Accra (5° 33′ N and 0° 13′ W), the capital city of Ghana to understand the traffic congestion (Armah et al., 2010). The reason for the study to be conducted in Accra is the poor urban transport system as opposed to the increased population. The transport system is characterised by long commuting times, extended journey delays, lengthy waiting lines for public transport, high accident rate and poor environment standards (Obeng-Odoom, 2009). This situation can be seen in many cities of developing countries across Asia and South-America (Faiz et al., 1995; Kutzbach, 2009; Mahendra, 2008; Obeng-Odoom, 2009), and even worse conditions in megacities such as Bangkok, Manila, Sao Paulo, Shanghai and Mexico City (Cervero & Golub, 2007; Mahendra, 2008). To overcome this situation, Armah et al. (2010) have created a causal loop to understand the complexity of the system (Figure 2.1).

In Figure 2.1, everything else being the same, a (+) sign indicates that changes are reinfored and (-) sign indicates changes are resisted. Reinfored effects are not necessarily good and resisted effects are not necessarily bad. Delays, shown by the double line on an arrow, occur in all systems and maybe in in seconds, minutes, hours, months or years. The delay marked on road construction and highway capacity is in years. The two counter-scting processes, pressure to reduce congesition and need for road contruction results in a closed-loop reinforcement. As far as the behaviour is concerned, there is compensating feeback as a response to the congestion. Decreased congestion makes driving more attractive, people use less public transport and public transport service go down. Hence, in the end, the whole system is caught in a feedback structure where public transport degrades, traffic increases, and congestion increases more and more.

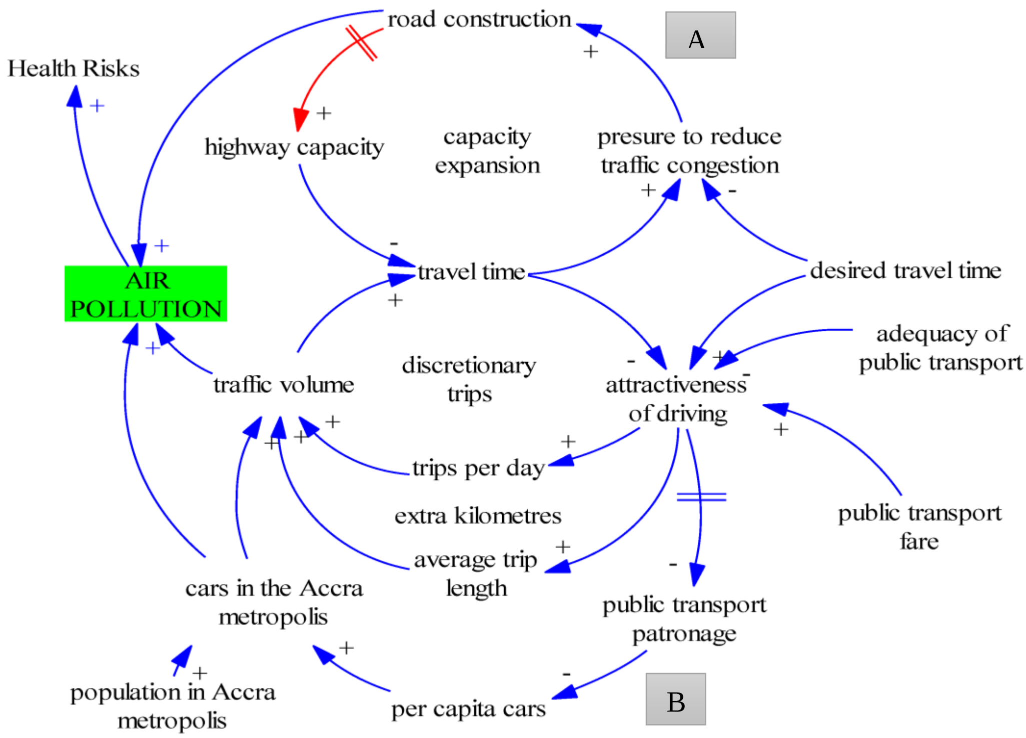



Figure 2.1 Traffic volume dynamics of traffic congestion in Accra (Armah et al., 2010). A plus sign (+) over an arrow from X to Y implies that if X increases so does Y, or if X decreases, Y also decreases. A minus sign (–) indicates an inverse effect.

Further, in Figure 2.1 (B), attractiveness of driving coupled with inadequate public transport is increased by reduced travel time. Increasing the attractiveness of driving leads to trips per day and average trip length. These two factors feed into increased traffic volume. The delay in public transport patronage is arisen by the time taken by an individual to save money to buy a car as well as readiness to change lifestyle. Gross Domestic Product (GDP) and livelihoods take significant time to improve. Therefore, the shift from public transport to personal cars often goes unobserved. Eventually, the number of cars in the Accra increases, leading to an increase in traffic volume. Therefore, even though the original goal was to reduce traffic congestion, the outcome indicates that the feedback loops rather reinforce the problem.

The resulting causal loop diagram from the case study helps us understand the robustness in the traffic environment and understand the complexity of the smart city system. Even though Armah et al., (2010) are successful in understanding the complexity of this transportation and traffic congestion, we believe that the transforming this environment to an HCDE managed by artificial intelligence can be used to understand and intervene any adverse situations such as traffic congestions.

### 2.2.1 What is a Natural System?

A system is defined as "a set of elements, interconnections and functions coherently organised to achieve a function or purpose" (Meadows et al., 1992). We explain this through three different use cases; natural ecosystem, a sports game and a novel smart city environment.

**Use case 1**: A natural predator-prey system consists of *elements* such as predators, preys, food etc.; *interconnections* such as sensing, running, attacking, hiding; and the *purpose* is the survival.

**Use case 2**: A volleyball team is a system with *elements* such as players, ball, coach and field; *interconnections* are the rules of the game, coach's strategy, players' communications and the laws of physics which govern the motion of the ball and players; and *purpose* of the team is to win the game, or have fun, or get exercise, make money, or all of the above.

**Use case 3**: A smart city traffic environment consists of subsystems with *elements* such as vehicles, roads and traffic lights, *interconnections* such as road rules, speed limits and driver behaviours and the *purpose* could be safety or minimising travel time or both.

Systems can change, adapt, respond to events, seek goals, mend injuries, and attend to their survival in realistic ways, although they may contain or consist of nonliving things (Figure 2.2)(Meadows, 2008). Systems are self-organising, self-repairing, resilient, and most of them are evolutionary. Completely new, unimaginable new systems can evolve from one system.

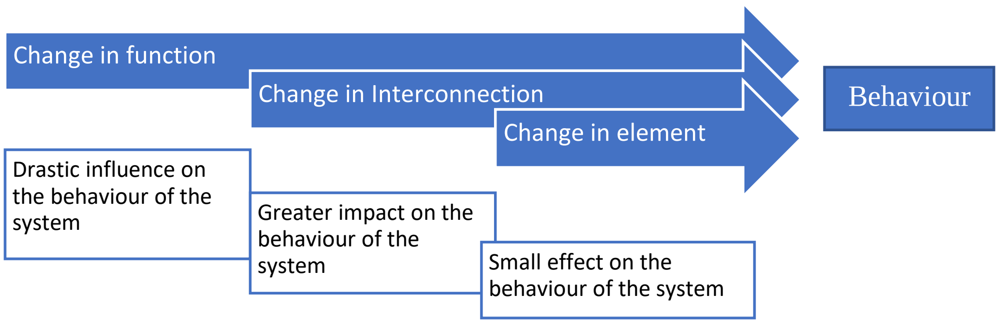



Figure 2.2 Changes in Natural System: as for the natural equilibrium/phenomena, changes can occur in the system's elements, interconnections and functions. Changing elements usually has the least effect on the system. Changing interconnections will have a greater impact on the system. Changes in the function or purpose can have a drastic impact on the system.

**Use case 1:** Changes in the population of a predator will affect the prey population. Changes in the interactions such as sensing will change how a predator hunt prey or ability of prey to escape from a predator. What if we keep the prey and predator population and the interactions the same, but change the purpose to reduce the population?

**Use case 2**: It is still recognised as a volleyball game even if all the players in a team are changes, although, this might have an impact on the performance of the team. Changing the rules of the game will change the volleyball game to a basketball game. What if you keep the player and the rules the same but change purpose of the game from winning to losing?

**Use case 3:** In a road traffic environment, changing the vehicles or the drivers will not change the traffic situation drastically although, this might have an impact on the traffic condition. A change in the road rules will have a bigger influence on a traffic situation. What if we keep the drivers and the roads rules the same and change the purpose?

A change in function changes a system profoundly, even if every element and interconnection remain the same. Changes in the function, interconnections and elements change the behaviour of the system and set the pace of the dynamics. With the changes, a system comes to a state of dynamic equilibrium and creates mutual causal interaction, where x affects y and y affects x, and so on. State of dynamic equilibrium creates an ongoing process called feedback loops. The natural environment is full of these mechanisms formed by the links between living and non-living things. For instance, this builds resilience by governing the way populations and food webs respond to events.

**Stabilising Loops – Balancing Feedback**

One common type of feedback loop that stabilises the system or sub-system is called a balancing feedback loop, and it aims at goal-seeking or equilibrating (Meadows, 2008). A balancing feedback loop opposes whatever direction of change that is imposed on the system. For example, if you push the number of elements too far up, a balancing feedback loop will try to pull it back down; if you shove the number of elements too far down, a balancing loop will try to bring it back up. This behaviour pattern can be seen in many systems in the world such as when (a) your body adjusts to blood-sugar concentration, (b) you pull your car to a stop at a stoplight, (c) a reservoir is brought up or down to its desired level, (d) a missile finds its target.

**Reinforcing Feedback Loop**

The second kind of feedback loop is called a reinforcing feedback loop that enhances whatever direction of change imposed on it (Meadows, 2008). Reinforcing feedback loops can be found wherever a system element can reproduce itself or grow as a constant fraction of itself, for example, money in the bank or pests in a cornfield.

Feedback loops are linked together in natural systems, often in tremendously complex patterns (Meadows, 2015). Many feedback loops in a complex system tug against each other, maintaining the natural equilibrium and denoting the cause and effect in a system. This nonlinear relationship between cause and effect does not produce a proportional effect. For example, as traffic flow on a highway increases, car speed is affected only slightly over a large range of car density (Figure 2.1). Eventually, however, small further increases in density produce a rapid drop-off in speed. When the cars on the highway build up to a certain point, it can result in a traffic jam, and car speed drops to near-zero.

Figure 2.3 summarises the components of a system structure. System structure consists of three components; elements, interconnections and purpose/function. Change in each component influences the system in different levels: *element* has the smallest influence and *purpose/function* has the biggest influence from a change. Changes in the *elements*, *interconnections* or *functions* set the pace for mutual causal relationships and create balancing and reinforcing feedback loops to maintain the equilibrium. As an HCDE is the digital representation of the natural environment, the digital environment can be represented with the same underlying system structure.

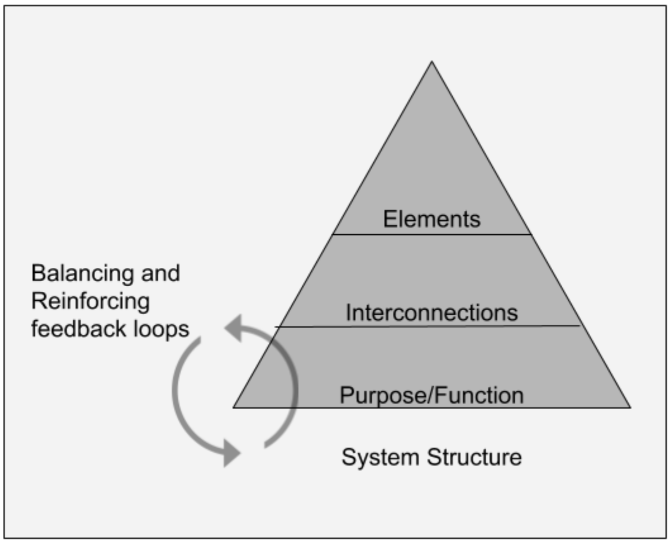



Figure 2.3 A system structure: Changes in elements have the lowest influence, whereas changes in the purpose or function have the biggest influence in the whole system structure. Change creates mutual causal relationships and identified as balancing and reinforcing feedback loops.

### 2.2.2 How Do Systems Keep Stability/Equilibrium?

In any highly dynamic systems such as ecosystems, stability is maintained through one of the three characteristics; resilience, self-organisation or hierarchy (Meadows, 2015) (Figure 2.4). These three characteristics lend systems the ability to function well over the long term and to be sustainable. As explained above, the digital environment represents the natural environment; hence, the properties of a natural environment that keep stability is applicable in the digital environment.

**Resilience**: "The ability to bounce or spring back into shape, position, etc. after being pressed or stretched." Resilience is a measure of a system's ability to stabilise within a dynamic environment. Resilience arises from a rich structure of many feedback loops that can work in different ways to restore a system even after a large perturbation. Ecosystems "learn" and evolve through their incredibly rich genetic variability. Awareness of changes in the system enables one to see many ways to preserve or enhance a system's resilience powers.

**Self-Organisation:** An impressive characteristic of a complex system is their ability to learn, diversify, complexify and evolve, known as self-organisation. Like resilience, the purpose of self-organisation is to provide short-term productivity and stability. Self-organisation produces heterogeneity and unpredictability, allowing and creating whole new structures. Ecosystems are remarkably self-organising, with multiple species holding each other in check, moving around in space, multiplying or declining over time in response to weather and food availability. The world is organised in self-organised subsystems aggregated into larger subsystems.

**Hierarchy:** In creating new structures and increasing complexity, one thing that a self-organising system often generates is a hierarchy. Hierarchies give system stability and resilience and reduce the amount of information that any part of the system has to track. Hierarchical systems are partially decomposable. When hierarchies break down, they usually split along their self-organised subsystem boundaries. Hierarchical systems evolve from the bottom up. The purpose of the upper layers of the hierarchy is to serve the purposes of the lower layers.

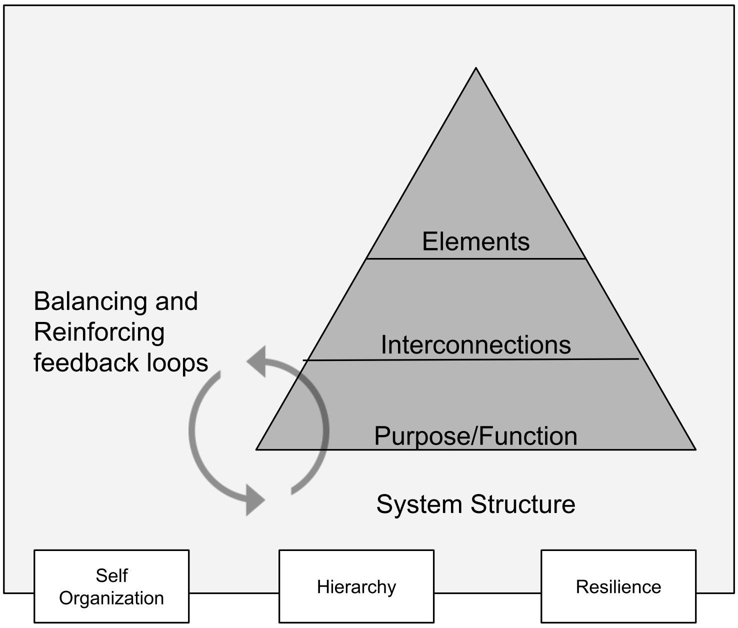



Figure 2.4 Properties of a System: self-organising, hierarchy and resilience.

### 2.2.3 Challenges of Systems

The systems are more complex due to non-linear relationships, non-existent boundaries, ubiquitous delays and layers of limits (Meadows, 2015) (Figure 2.5). The digital environment that represents the natural environment comprises of these challenges. These challenges depict the need for an AI algorithm to understand the digital environment.

**Non-linear Relationships**

To understand the nature of the relationships, we tend to look at the world as having linear relationships. However, the world is full of non-linear relationships where the cause and effect do not produce a proportional effect. Below are a couple of examples for non-linear relationships explained by (Meadows, 2008).

- As traffic flow on a highway increases, car speed is affected only slightly over a large range of car density. Eventually, however, small further increases in density produce a rapid drop-off in speed. And when the number of cars on the highway builds up to a certain point, it can result in congestion, and speed drops to zero.

- Soil erosion can proceed for a long time without much effect on crop yield—until the topsoil is worn down to the depth of the crop's root zone. Beyond that point, a little further erosion can cause yields to plummet.
- A little tasteful advertising can awaken interest in a product. A lot of blatant advertising can cause disgust for the product.

Nonlinearities are important because they change the relative strength of feedback loops. It changes the behaviour of a system from one mode to another.

**Non-existent Boundaries**

A system is connected to everything else, but not neatly. That is, systems are not separate as the world is a continuum. There are only boundaries of word, thought, perception, social agreement and mental model. A boundary is defined based on the purpose or task. If a boundary is defined too narrowly, that is, if we separate a system from everything else, the system will surprise us. For example:

- We try to separate and solve the urban traffic problem by building highways, attracting new housing systems and more cars coming to the road. As a result, the urban traffic problem will continue.
- We try to solve the sewage problem by throwing the waste into the river; the boundary for sewage problem becomes the whole river, soil and groundwater surrounding the river.
- The boundary for a national park does not end at the park's physical border, but rather by migrating wildlife, by waters that flow in and out of or under the park, by the effects of economic development at the park's edges and even the climate change.

Finding the correct boundary is often challenging. The boundaries are of our own making and that they can and should be reconsidered for each new discussion, problem, or purpose.

**Layers of Limits**

The world comprises of many causes that consistently work together resulting in many effects. In other words, multiple inputs produce multiple outputs, and therefore virtually all of the inputs and outputs are limited. For example:

- A manufacturing process needs labour, capital, energy, land, raw materials, water, credit, technology, insurance, management, customers, public-funded infrastructure and government services such as health, police, and fire protection, a healthy ecosystem.

The concept of layers of limit is simple yet widely misunderstood. Any physical system with multiple inputs and outputs such as an economy, a production process or a population is surrounded by this limiting factor. As a system grows, it interacts with itself and influences its limits. A coevolving dynamic system is created by the growing entity and its limited environment together.

**Ubiquitous Delays**

Delays are ubiquitous in systems. For example:

- The delay between catching an infectious disease and getting sick enough to be diagnosed, for example, COVID-19.
- The delay between pollution emission and the purification or diffusion or absorption of the pollutant to the point at which it harms the ecosystem.
- The gestation and maturation delay in breeding animals or plants.
- The delay in changing the social norms for desirable family size.
- The delay in retooling a production stream and the delay in turning over a capital stock. The production of a new car takes about three to eight years. That car may stay on the road for about ten to fifteen years.

Delays determine how timely is the information passed around a system, how accurately they hit their targets, and how fast systems can react.

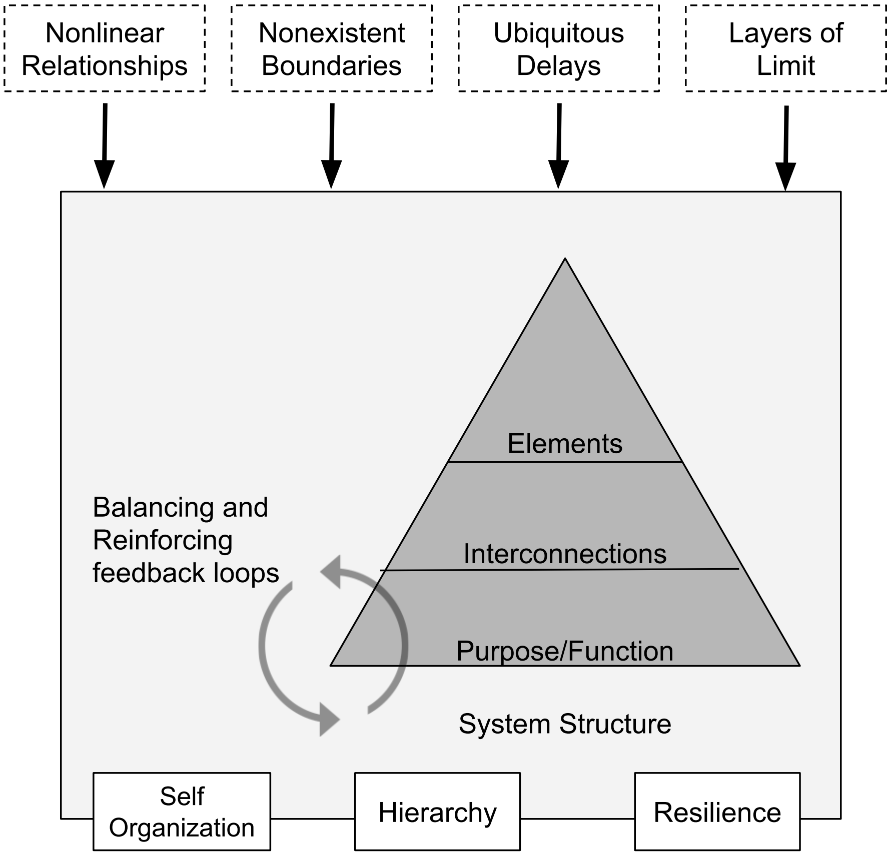



Figure 2.5 Challenges of a system: Nonlinear relationships, non-existent boundaries, ubiquitous delays and layers of limit

### 2.2.4 Events and System Behaviour for Understanding the System Structure

Human often observes the interconnected, feedback dominated world as a series of events (Meadows, 2008) (Figure 2.6). Daily news provides us glimpse of natural disasters, civil unrests, wars, political agreements, real estate booms or busts, technological advancements etc. These events, each one different from another, can be fascinating and does not hold much explanatory or predictive value. Events are the most visible aspect of a larger complex like the tip of an iceberg rising above the water but not always the most important.

Events accumulate into a dynamic pattern of behaviour that provides more explanatory and predive value (Figure 2.6). A system's behaviour can be seen by its growth, stagnation, decline, oscillation, randomness, or evolution. A better behaviour level understanding can be achieved if the above-mentioned daily news is provided with a historical perspective. This behaviour relates information on the underlying system structure and structure is the key to understanding what is happening and *why* (Meadows, 2008).

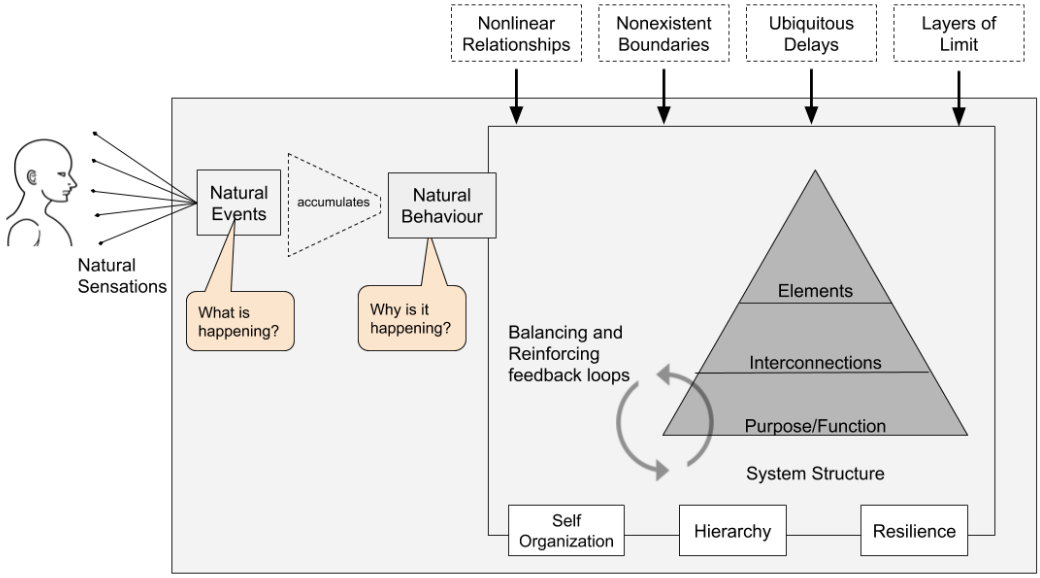



Figure 2.6 Overall view of a system - we sense the changes in the system as events, events accumulate into system behaviour, system behaviour provides information to the system structure.

System thinking looks at the structure of the system with its interlocking stocks, flows, and feedback loops and at the behaviour with time graphs. System structure determines the behaviours that are hidden in a system. A goal-seeking balancing feedback loop holds a dynamic equilibrium. A reinforcing feedback loop generates exponential growth. The two of them linked together are capable of growth, decay, or equilibrium. If it contains delay, it may produce oscillations.

## 2.3 Related Work in Conventional Digital Environments

Many data stream applications work in dynamic environments where the underlying process is not strictly stationary. This results in variability, a research challenge that is discussed in detailed in this section. As our environment is naturally dynamic and constantly changing over time, the big data generated in the current digital environments have variability embedded in them. Examples of such systems are real-time surveillance systems, telecommunication systems, sensor networks. For an accurate result, learning algorithms that learn from these data must track this changing behaviour and adapt the decision models accordingly.

### 2.3.1 Knowledge Discovery from Conventional Digital Environment

Extracting potentially useful knowledge from data streams is challenging. Machine learning and knowledge discovery techniques in research and practice focused on small datasets where the whole training dataset is available to the algorithm. These algorithms usually process the training data multiple times and output a decision model. The rationale behind this practice is that examples are generated at random according to some stationary probability distribution (stochastics) (Brain & Webb, 2002). Most of these machine learning algorithms use a greedy, gradient descent or ascent search in the space of the learning model and are prone to high-variance and overfitting problems. In current big-data applications, learning algorithms need to learn in dynamic environments, where the data are collected over time. A desirable property of these algorithms is the ability to incorporate new data resembling new concepts. If the process is not strictly stationary, as are most real-world applications, the target concept could gradually change over time.

This new world in movement induced by ubiquitous environments redefines the characteristics for the data (Gama, 2010):

- Data are made available through unlimited streams that continuously flow, eventually at high speed, over time.
- The underlying regularities may evolve over time rather than be stationary.
- The data can no longer be considered as independent and identically distributed.
- The data are now often spatially as well as time situated.

These new characteristics of data affect even the very basic operations at the core of learning methods. For example, (1) when all data are available and stored in a working matrix, we can apply any clustering algorithm over the transpose of the working matrix. This is not applicable when the data evolve over time as the transpose operator is a blocking operation where the first output tuple is available only after processing all the input tuples (Barbará, 2002); (2) computation of entropy of a data stream which is no longer finite, the number of variables is huge, and the target classes are not known prior; (3) continuous maintenance of the k-most frequent items in a retail data warehouse with three terabytes of data, hundreds of gigabytes of new sales records updated daily with millions of different items. Solutions to these problems require new sampling and randomising techniques, together with new approximate and incremental algorithms (Aggarwal, 2007; Gama & Gaber, 2007; Muthukrishnan, 2005).

### 2.3.2 Research Problems and Challenges in Data Stream Analytics

There are many research problems and challenges addressed in data stream learning (Gaber, Krishnaswamy, et al., 2005; Gaber, Zaslavsky, et al., 2005; Golab & Özsu, 2003). The major challenge in data stream mining is the **variability** discussed in detail in section 2.3.3. Another main issue addressed is the **continuous flow of data streams** where traditional database management systems cannot deal with a high **velocity** of data. Novel indexing, storage and querying techniques are required to handle this fluctuated flow of information streams. **Scalability** is another issue addressed in data stream analytics. A large amount of streaming data are generated in resource-constrained networks such as sensors networks (Bhargava et al., 2003). Scalability is a crucial issue as the generated streams are sent to a central site. With the scalability, **unbounded memory** must be addressed as well. Most of the machine learning methods require data to be present in memory while executing learning algorithms. Due to the high **volume** of data generated from the streams, machine learning algorithms have to be executed online.

Apart from the data management issues in data streams analytics, issues related to the results also need to be addressed. Techniques based on space and time must be accompanied with acceptable **accuracy levels**. Approximations algorithms (Muthukrishnan, 2005) can guarantee error bounds and sampling techniques such as VFML (Domingos & Hulten, 2001) allows adaptation to the concept seen before. Further, how the results changed over time would provide an insight into the **dynamics of the data streams** and benefit many temporal-based analysis applications. This issue has been addressed in MAIDS (Cai et al., 2004). Finally, the **visualisation of the data mining results** is also quite challenging and is addressed in (Kargupta et al., 2002).

There are many more research issues and challenges that have not been addressed (Gaber, Zaslavsky, et al., 2005). The **integration between data stream management systems and the ubiquitous data stream mining** approaches is an essential issue that needs to be addressed for a fully functioning ubiquitous mining. Further, the possibility of **data pre-processing** in stream mining process has not been addressed so far in the literature. Data pre-processing consumes major effort in the data mining process, and it is challenging to automate this process. Limitation of **data stream mining technologies** is also an important issue that needs to be addressed for **real-world applications**. Thus far, techniques proposed to improve the computational complexity of the mining algorithms within some margin of error without noting the real needs of the applications. Providing the user with the environment's real-time situation will be more useful than achieving better computational accuracy.

In addition to the data stream mining problems and challenges addressed above, a major issue in data stream mining is the variability known as concept change elaborated next in this chapter.

### 2.3.3 Concept Change

Due to the variability of the data stream, the underlying concepts could *evolve* from time to time, each time after minimum permanence. The change is reflected in the feature vector, where old observations that reflect the behaviour becomes irrelevant to the current state of the phenomena. Formally, concept change (*known widely as concept drift)* between time point $t_o$ and time point $t_1$ can be defined as equation 2.1,

$$\exists x: P_{t_0}(\vec{x}_i, y) \neq \; P_{t_1}(\vec{x}_i, y) \tag{2.1}$$

where $P_{t_0}$ denoted the joint distribution at time point $t_0$ between the set of input variables $\vec{x}_i$ and the target variable $y$ (Gama et al., 2014).

To explain the change detection further, suppose we have a sequence of pairs $(\vec{x}_i, y_i)$ where $y_i \in \{C_1, C_2, \dots, C_k\}$. At each time stamp $t$ the learning algorithm outputs a prediction class $\hat{y}_t$. If the examples are independent and generated at random by a stationary distribution $D$, a traditional supervised machine learning algorithm (decision tree, regression etc.) learning from the sequence can approximate the class label. If $D$ is not stationary, time to time, the distribution that is generating the examples changes. In this case, the data stream can be defined as sequences $\langle S_1, S_2, \dots, \; S_k, \dots \rangle$, where each element is a set of examples generated by a stationary distribution $D$. A traditional machine learning algorithm learning from this data stream cannot guarantee arbitrary precision. The main problem of learning from this stream is not knowing when the change occurs. Figure 2.7 illustrates two different examples of change.

A *concept* can be defined as the relationship between a set of independent variables, $\vec{x}$, and a dependent variable, $y$. The joint probability $P(\vec{x}, y)$ can be decomposed in (equation 2.2):

$$P(\vec{x}, y) = P(y/\vec{x}) \times P(\vec{x}) \tag{2.2}$$

Lazarescu et al. (2004) define concept change in terms of consistency and persistence. Consistency refers to the change $\epsilon_t = \theta_t - \theta_{t-1}$ that occurs between consecutive examples of the target concept from time $t-1$ to $t$, with $\theta_t$ being the state of the target function in time $t$.

A concept is consistent if $\epsilon_t$ is smaller or equal than a consistency threshold $\epsilon_c$. A concept is persistent if it is consistent during $p$ times, where $p \geq \frac{\omega}{2}$ where $\omega$ is the size of the window. If a concept change is both consistent and persistent, it is considered permanent or as a real concept change. If the concept change is consistent but not persistent, it is a virtual concept change. Noise has neither consistency nor persistency.

In the setting that we are considering, the nature of change is diverse and abundant. Existing machine learning algorithms learn from observations described by a finite set of attributes. However, in IoT applications, important properties that influence the behaviour of nature could be *hidden.* As a result, concepts learned at one time can become inaccurate. There could be two implications of these changes:

1. The data distribution $P(y/\vec{x})$ changes and affects the predictive decision. This is known as Real Concept Change (or Real Concept Drift). Such changes can happen with or without changes in $P(\vec{x})$. An illustrative example of a real concept change in two-dimensional feature space with two classes is presented in Figure 2.7 Real Concept Change - example demonstrates how the decision boundary changes in a two-dimensional feature space. Different colours represent different classes.

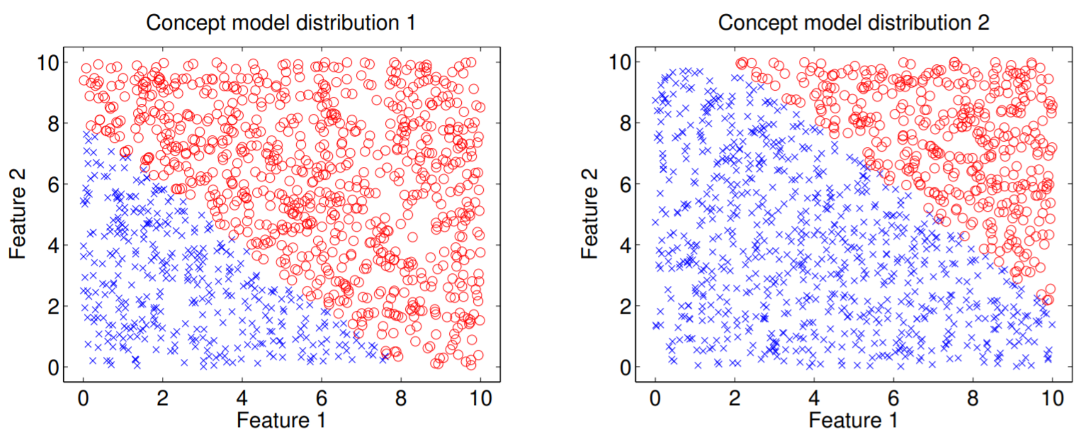



Figure 2.7 Real Concept Change - example demonstrates how the decision boundary changes in a two-dimensional feature space. Different colours represent different classes.

2. The changes are visible from the data distribution without knowing the true labels, that is, changes in the distribution of incoming data $P(\vec{x})$ without affecting $P(y/\vec{x})$. This is known as Virtual Concept Change (or Virtual Concept Drift). An example of virtual concept change is spam filtering applications, where the data priors change but not the meaning (Figure 2.8).

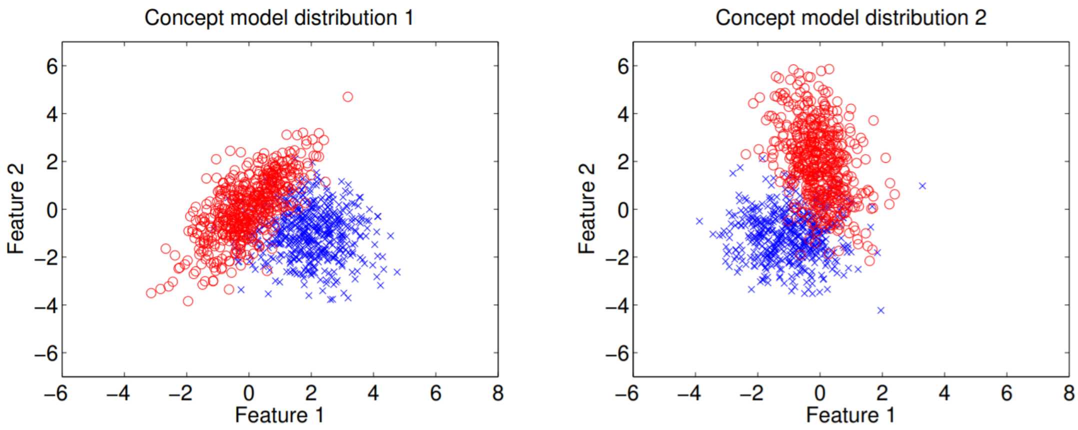



Figure 2.8 Virtual Concept Change - corresponds to a change in data distribution that leads to changes in the decision boundary. Different colours represent different classes.

Further, the rate of change over time also provides important information regarding the concept change. Figure 2.9 illustrates different forms of change on one-dimensional data where the change happens in the data mean.

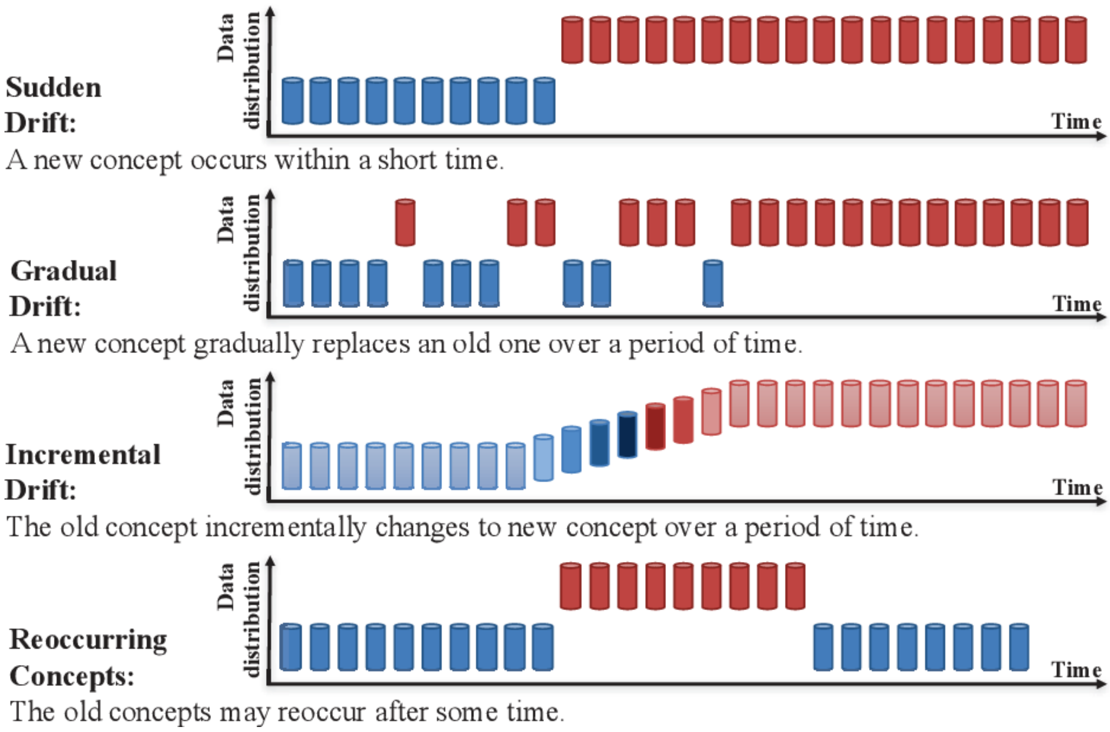



Figure 2.9 A demonstration of concept change types (A. Liu, 2018). **Sudden Drift -** A concept change may happen suddenly or abruptly by switching from one concept to another. In this case, drifts may introduce new concepts that were not seen before. **Gradual Drift** – A concept change may happen gradually with going back to the previous concept from time to time, for some time, before changing completely to a different concept. **Incremental Drift** – A concept change may happen incrementally with many intermediate concepts in between. **Reoccurring Drift -** A concept change may introduce previously seen concepts may reoccur after some time.

Most commonly found concept changes are abrupt drifts, which is also known as concept shift, and reoccurring drifts, which denotes a pattern (Gama & Gaber, 2007). Gradual and incremental changes in the target concept, such as the rate of changes in price, are another types of concept change. However, gradual drifts and incremental drifts are not common in the current digital environment (Gama, 2010). Further, slow changes can be confused with stationarity.

A challenge faced by concept change detection algorithms is that they must differentiate noise or outliers from change (Gama et al., 2014). The difference between noise and a concept change is persistence, where there would be a consistent set of examples in the new concept (Gama et al., 2014). Algorithms for change detection must combine robustness to noise with sensitivity to concept change (Gama et al., 2014).

### 2.3.4 Characterisation of Change Detection Methods

Most machine learning techniques that deal with changing concepts assume that most recent examples are the most relevant ones (Klinkenberg, 2004; Widmer & Kubat, 1996). In this section we investigate a taxonomy for adaptive algorithms that learn a predictive model from evolving data with unknown dynamics; (1) data management, (2) forgetting mechanisms, (3) detection method, (4) adaptation method.

#### *2.3.4.1 Data Management*

Learning under concept change requires updating the underlying predictive model with new information and forgetting the old and irrelevant information (Gama et al., 2014). Data management methods represent how the information is stored and used by machine learning techniques. Data can be characterised as short-term memory represented as data and long-term memory represented as a generalization of data. The short-term memory or data will be consumed as a full dataset or partial dataset by the machine learning algorithms. Learning from partial memory typically aims at learning from most recent data: either a single example or multiple examples (Figure 2.10).

**Short-term Memory (Data) – Full Dataset**

Some machine learning algorithms use the full dataset stored in memory to learn and detect concept changes. In this data management method, weighting to the examples is included based on the age so that high importance is added to the most recent data and decrease the importance with time. Therefore, the oldest information has less importance. The strategy can be implemented using linear decay (Koychev, 2000, 2002) or exponential decay (Klinkenberg, 2004).

**Short-term Memory (Data) – Partial Dataset**

This data management method stores only the most recent examples in a *first-in-first-out (fifo)* data structure, where *fifo* define a time window over the stream of examples. At each processing time, the learning algorithm uses only the examples that are available in the current time window.

**Single Examples**: Online learning algorithms hold the base of using only one example at a time in the order of arrival (*fifo*). In this approach, the learning algorithm does not store the training dataset in memory and does not access the previous examples. The online learning algorithms are recognised as naturally adaptive to evolving concepts. These algorithms update to the model-driven by the error; hence, the model will be continuously updated with the most recent examples. However, online learning systems do not have explicit forgetting mechanisms. Adaptation to new concepts would be visible only as the old concepts are diluted due to the new incoming data. Such existing machine learning algorithms that use single instance memory systems include STAGGER (Schlimmer & Granger, 1986), DWM (J. Zico Kolter & Maloof, 2007, 2003), SVM (Syed et al., 1999), IFCS (Bouchachia, 2011), and GT2FC (Bouchachia & Vanaret, 2014). Although algorithms such as WINNOW (Littlestone, 1988) and VFDT (Domingos & Hulten, 2000) can adapt to slow changes over time, their slow adaptation affects the detection of abrupt concept changes.

**Multiple Examples**: This data management method aims at learning from a set of recent examples. FLORA (Widmer & Kubat, 1996) is a supervised incremental learning algorithm for detecting concept changes in evolving data that uses a window of data based on *fifo*. In general, the training window size can be fixed or variable. Fixed-size sliding windows will store a fixed number of most recent examples where the oldest example will be discarded when a new example arrives. Variable size sliding window varies the number of examples in a window over time, depending on the indication of changed concept.

These learning algorithms are updated following two processes, (1) learning process in which the algorithm builds a new model based on the examples available on the new window (2) and a forgetting process in which the data that are moving out of the window are discarded. The key limitation of the sliding window approach is the need for defining the appropriate window size. Although a short window would reflect the current distribution more accurately and ensure fast adaptation to concept changes, it could affect the system's performance during the stable periods. A large window will provide a better performance in a stable time, but detection and adaption to concept change will not be efficient. There are two implementations of windows:

I. **Fixed-size windows** – This method stores a fixed number of examples in memory where new examples are available oldest examples are discarded. This is the simplest method in handling concept changes and can be used as a baseline for performance.

II. **Adaptive size windows** – This method stores carriable number of examples in the window and is decided based on the detection model. The addition of new data and the deletion of old data keep the window consistent with the current concept. The most common implementation is to decrease the window size in case of change detection and increase the window size otherwise.

    FLORA2 (Widmer & Kubat, 1996) was the first algorithm to use adaptive windows. Modifications of these algorithms deal with recurring concepts (FLORA3) and noisy data (FLORA4). A study that uses support vector machines (SVMs) for detection and adaptation concept change maintains an appropriate size window, adjusting the window size based on the estimate of the generalization error. João Gama et al., (2004), Klinkenberg (2004), and Maloof & Michalski (1995) have also proposed similar algorithms for concept change detection and adaption based on learning window of variable length. Although these methods assume that recency of the data is associated with importance and relevance, this assumption may not be true in every circumstance (when the data is noise or concept reoccur). Also, windowing may fail if a change lasts longer than the window size.

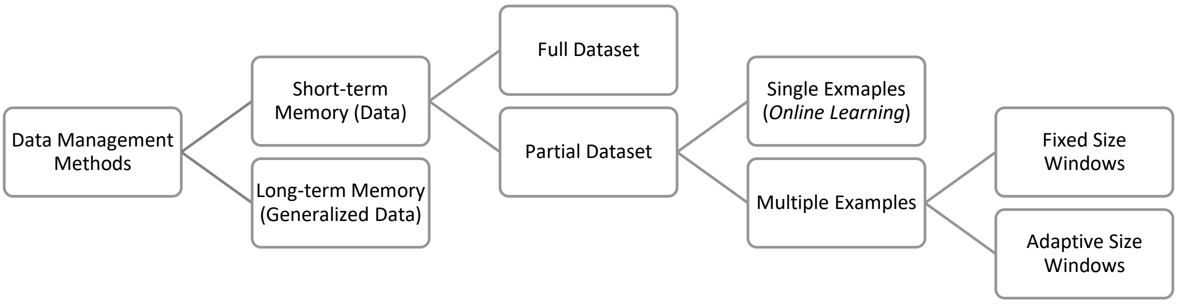



Figure 2.10 Data management methods – how the data processed as a data stream are stored or retrieved by the learning algorithm to detect and adapt to concept change.

#### *2.3.4.2 Forgetting Mechanisms*

Dynamic environments with non-stationary distributions must forget the observations that are not consistent with the current natural behaviour (Gama et al., 2014). Change detection algorithms will only adapt to the new concepts only if old information is forgotten. The data management models also need to address forgetting mechanisms. Weighting examples corresponds to gradual forgetting, and time windows correspond to abrupt forgetting (Figure 2.11). It is possible to combine both forgetting mechanisms by weighting the examples in a time window (Klinkenberg, 2004).

There are several models based on abrupt forgetting have been presented in the literature which can be separated into implementations of two basic types of sliding windows;

I. sequence-based – where the number of observations defines the window size. There are different models for sequence-based windows;
    a. sliding windows of size $j$ which stored only the most recent examples in a *fifo* data structure.
    b. landmark window which stored all the examples during a given timestamp, resulting in a variable size window.
II. timestamp-based – where the window size is defined by the time duration $t$ includes all the elements that arrived within the time.

Gradual Forgetting is a full memory approach where no examples are completely discarded from memory. Examples in memory are associated with weights that reflect their age where the importance of an example in the training set decreases over time. Suppose at time $t$, the stored sufficient statistics is $S_t - 1$ and the example is $X_t$. If the aggregation function is $G(X, S)$, the new sufficient statistics are computed as $S_t = G(X_t, \alpha S_i - 1)$ where $\alpha \in (0,1)$ is the fading factor. Further, Koychev (2000, 2002) has presented a technique based on linear decay and Klinkenberg (2004) has proposed a technique based on exponential decay where weights are assigned to examples according to their age using an exponential aging function $w\beta(X) = \exp(-\beta k)$, where the example $X$ appeared $k$ timestamps ago and the parameter $\beta$ defines how fast the weight decrease.

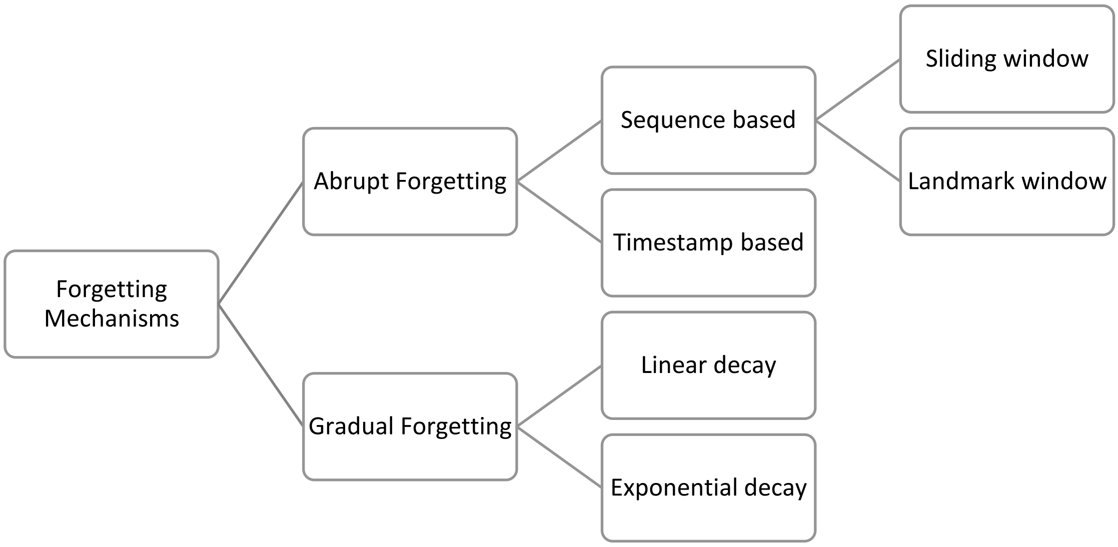



Figure 2.11 Forgetting mechanisms - how the information that is no longer relevant, are removed.

### *2.3.4.3 Detection Methods*

Detection methods characterise the techniques and mechanisms of concept change detection. Change detection algorithms can provide a meaningful description indicating change-points or small time-windows where the change occurs (Gama et al., 2014). Two different approaches are used in literature (Figure 2.12);

I. Monitoring the evolution of performance indicators. Indicators such as performance and properties of data are monitored over time for a good overview of these indicators.

II. Monitoring distributions between a reference window summarises past information and a window representing the most recent examples.

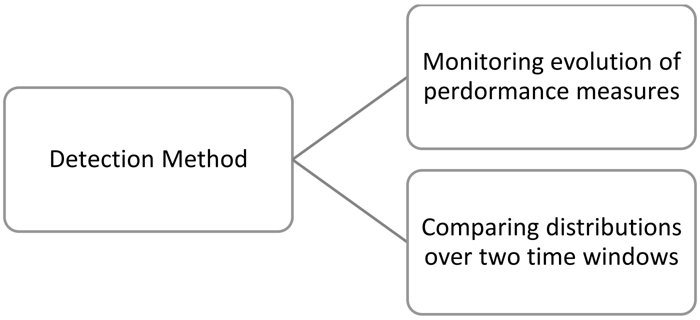



Figure 2.12 Concept change detection methods

**Change Detection based on Performance Indicators**

Majority of the work in literature are based on performance indicators. These algorithms monitor the changes in the performance indicators such as accuracy or error-rate of the learning algorithm and detects as a concept change if there is a statistically significant increase (error rates) or decrease (accuracy). Widmer & Kubat (1996) have developed a rule-based classifier based on window adjustment heuristic. To detect a concept change, the accuracy of the algorithm is monitored over time where the window size is adjusted accordingly. Klinkenberg & Renz (1998) propose monitoring accuracy, recall and precision over time and later compare with the confidence interval of standard sample errors for a moving average value of each indicator. Klinkenberg & Joachims (2000) have proposed an effective and efficient method of detection using properties of Support Vector Machines in which the window size is selected to minimise the generalization error on new examples.

Drift Detection Method (DDM) (Gama et al., 2004) defines warning levels and drift levels for concept change detection. DDM monitors the online error-rate of the algorithm in a fixed window while the training examples are presented in sequence. The algorithm will classify the newly arriving examples with the current model. Although the algorithm expects a low error value in this classification, the error will increase if the distribution changes. Here the algorithm defines a warning level and subsequently a drift level. When the warning level is reached, DDM starts training another model while the current model continues to provide classifications. When the drift level is reached, the old model is replaced by the new model. This principle has been adopted and applied in Learning with Local Drift Detection (Gama & Castillo, 2006), Heoffding's inequality based Drift Detection Method (Frías-Blanco et al., 2015), Fuzzy Windowing Drift Detection Method (A. Liu et al., 2017), and Dynamic Extreme Learning Machine (S. Xu & Wang, 2017).

In contrast to DDM, Nishida & Yamauchi (2007) have proposed a statistical test of equal proportions based method where the concept change is detected by comparing the examples of the most recent time window with the overall examples available. The time window is defined by the user, and each timestamp comprises of two-time windows. ADaptive WINdowing (ADWIN) (Bifet & Gavaldà, 2007) is another popular drift detection method that uses two-time windows. Unlike the above statistical method, ADWIN uses sliding windows whose size does not need to be defined in advance. Sliding window size is recomputed online based on the rate of change observed from the data in the current window. There are many concept change detection algorithms/methods that are derived from ADWIN: (Bifet & Gavaldà, (2009), Bifet, Holmes, Pfahringer, & Gavaldà, (2009), Bifet, Holmes, Pfahringer, Kirkby, et al., (2009), Gomes et al., (2017)).

**Change Detection based on Distribution**

Methods that use distribution for concept change detection use a distance function or metric to quantify the dissimilarity between the distribution between the old window and new window (Dasu et al., 2006; Lu et al., 2014; Shao et al., 2014). If there is a statistically significant dissimilarity, the learning model needs to be changed. The advantage of this method is that the algorithms address the concept change from the root sources, which is the distribution of the data. With this drift detection method, more information regarding the drift such as the time the drift occurred can be identified accurately. However, these algorithms need the user to predefine the old window size and new window size and keep the historical data in the memory. This is infeasible in the current digital environment in which high volumes of data are processed in high velocity. Further, these algorithms are known to incur a high computational cost.

Kifer et al. (2004) have proposed a method that uses the distribution for concept change detection. They propose algorithms that monitor two probability distributions drawn from samples and decide whether these distributions are different. VFDc is another algorithm that uses the same approach continuously monitor the differences between two class distributions of the examples.

#### *2.3.4.4 Adaptation Methods*

The adaptation methods characterise the adaptation of the learning models to the changing concepts. In literature there are two different approaches (Figure 2.13);

I. Blind methods where the learning algorithm is adapted at regular time intervals without considering whether changes have occurred. This is implemented with the methods that weight the examples according to their age and the methods that use time window of fixed size explained earlier (Klinkenberg & Joachims, 2000; Klinkenberg & Renz, 1998; Maloof & Michalski, 1995; Widmer & Kubat, 1996).

II. Informed methods where the learning algorithm is modified only after a change has occurred (Gama & Castillo, 2006). This method is used in conjunction with a detection method.

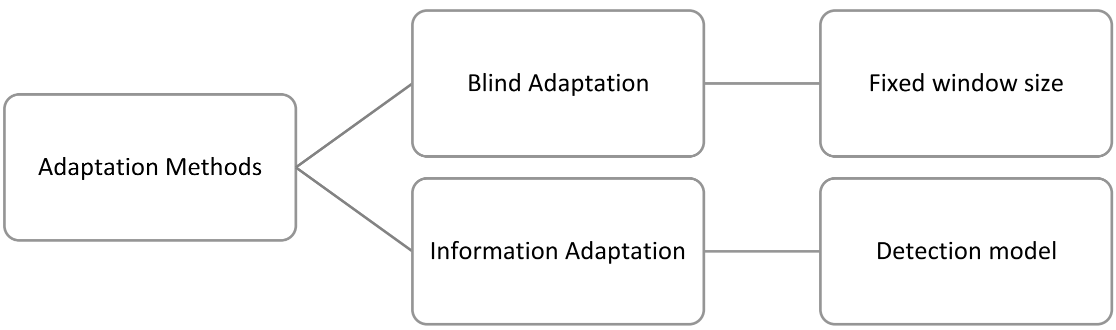



Figure 2.13 Adaptation methods – how the learning algorithm is adapted to the new concept.

#### *2.3.4.5 Learning Methods*

The learning methods characterise techniques and mechanisms for generalising from examples and updating the predictive models from evolving data. Learning can carryout whenever new examples are available. The literature discusses two different learning modes (Figure 2.12);

I. **Retraining** where the current model is discarded, and a new model is built from scratch using buffered data. In the beginning, the learning model is trained with all the available data. Whenever new data are available, the previous model is discarded, and a new model is learning from all the available data (Klinkenberg & Joachims, 2000; Street & Kim, 2001; Zeira et al., 2004). Retraining has to emulate incremental learning with batch-learning algorithms (Gama et al., 2004).

II. **Incremental** where the model is incrementally updated and adapted to the new data. Incremental learning algorithms process input data one by one or in batches and update the statistics (from previous data or summaries of data) of the existing model. Implementations of incremental learning in concept change detection and adaption include WINNOW (Littlestone, 1988) and MBW (Carvalho & Cohen, 2006). MBW is developed based on a well-known algorithm including perceptron and multilayer perceptron. Although they have usually been using several passes through the training data when restricted to a single training passing over the data, they are particularly relevant for massive streaming data. The learning model is updated with the current data and with time, newly arrived data tend to erase the prior knowledge. In models such as artificial neural networks, learning in inevitably connected with forgetting. The ability to continuously learn from a stream of examples while preserving previously learned knowledge is known as the stability plasticity dilemma [Carpenter et al. 1991a]. It is a dilemma because there needs to be a balance between being stable to handle noise and being able to learn new patterns.

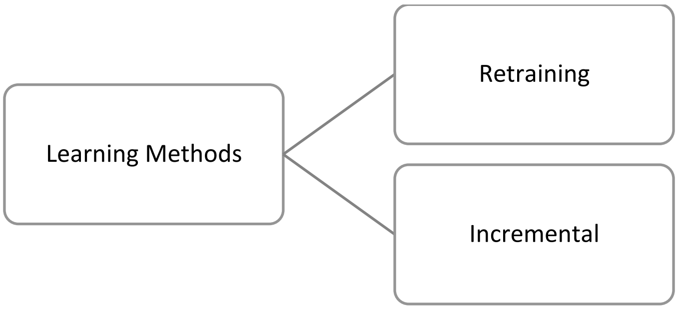



Figure 2.14 Learning methods for concept change detection and adaptation

## 2.4 Chapter Summary

The chapter presented a comprehensive literature survey on HCDE, system dynamics and current state of the art machine learning techniques for concept change detection on conventional digital environments. The chapter characterised an HCDE by studying the theories of complex environments and elaborated on how theories of system dynamics are used to characterise natural, social and human-made complex environments and the limitations of this approach. The latter delineated the challenges that make a system extremely complex and properties that keep a system's stability. The chapter elaborated how a natural system is sensed through events and how the events accumulate into system behaviour. Understanding the system behaviour relates information on the underlying system structure.

The chapter further provided a detailed literature study on knowledge discovery from data streams in an HCDE and summarized research problems and challenges. It discusses the theories of concept change and existing work on change detection in line with data management, forgetting mechanisms, detection, adaption and learning methods. Data management methods represent how the information is stored and used by machine learning techniques and categorized mainly into short term memory and long-term memory. Forgetting mechanisms represent how to forget the observations that are not consistent with the current natural behaviour and mainly categorized into abrupt forgetting and gradual forgetting. Detection methods characterise the techniques and mechanisms of concept change detection and are based on monitoring the evolution of performance indicators and monitoring the distributions between reference windows. The adaptation methods characterise the adaptation of the learning models to the changing concepts and are categorized into blind adaption and information adaptation. Learning methods characterise techniques and mechanisms for generalizing from examples and updating the predictive models with evolving data where retraining and incremental methods are discussed.

# Chapter 3

# A Conceptual Model of Self-Structuring AI

The constituents of an HCDE, the natural, social, virtual and artificial environments, are dynamic and continuously evolving. An HCDE wherein electronic devices sense an environment will contain the changes that occur in that setting. Drawing inspiration from nature to understand an HCDE, firstly, this chapter elaborates on natural equilibrium; (a) a classic example of natural equilibrium in eco-systems, (b) the stability in internal, physical, and chemical conditions maintained by living systems and (c) a situation in which supply and demand decide equilibrium values of economic variables. Next, the chapter investigates the need for equilibrium in HCDE and conceptualises the equilibrium in an HCDE using system dynamics theories. This section explores how the digital representation of natural environment is generated by sensing the natural environment. Finally, a conceptual model materialized in Self-Structuring AI is proposed for concept change detection and understanding the causality for concept change.

# 3.1 Equilibrium in Nature

The idea of 'Balance of Nature' has been a long-standing concept where the entire earth has been viewed as a potentially self-regulating system kept in stable equilibrium by predictable forces if left alone. It is usually implied that undisturbed nature is ordered and harmonious, and with feedbacks, systems return to equilibrium/stabilization after disturbances (Wu & Loucks, 1995). The natural world is full of such systems trying to stabilize with feedback mechanisms formed by the links between living and non-living things. This can be observed in areas such as ecosystems, biological systems, societies, economies etc. For example, in natural ecosystems, nature builds resilience by governing how populations and food webs respond to events. That is, when predators hunt preys, prey population drops, causing the predation rate to drop and allowing the prey population to grow again (Figure 3.1). Ecosystems would maintain a long-term equilibrium allowing food chains to persist over time. Both populations' dynamic equilibrium is interesting because it shows a direct cause and effect relationship between different species in ecosystems.

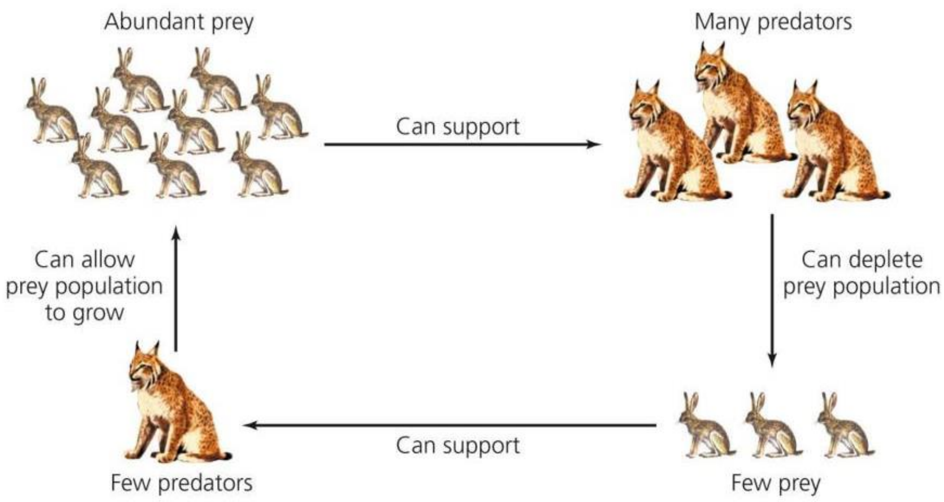



Figure 3.1 Ecosystem feedback is the effect that change in one part of an ecosystem has on another and how this effect then feeds back to affect the source of the change inducing more or less of it. These feedback loops form the basic dynamics for regulating the state of the ecosystem. (Green, 2019)

## Tropical Cascade – "Wolfs Change Flow of Rivers"

Tropical cascade is an ecological process that starts at the top of the food chain and tumbles down to the bottom, involving predators, preys and plants. A classic example of the topical cascade occurred in Yellowstone National Park (YNP), the USA in 1995-96 when grey wolves were reintroduced after seven decades of absence (Fortin et al., 2005). The historical presence, then absence, and now the presence of wolfs in YNP represents natural experiment through time and an opportunity to study cascading trophic interactions. Even though wolfs are known to be predators, the ecological study showed that they gave life to many more (Figure 3.2).

The 70 years of the wolf-free period significantly impacted wildlife habitat, soils and woody plants resulting in a collapse of a tri-trophic cascade. Due to the absence of wolfs, there was a significant increase in the number of elk in the YNP. It was very difficult to manage the elk population despite the human effort to control them. As a result, the natural vegetation in YNP reduced to almost nothing.

The reintroduction of wolfs, even in a small number, made a remarkable change in the environment. Other than been eaten by the wolfs, the behaviour of the elk changed significantly. They started to avoid some parts of the park such as valleys and gorges, areas that are a problem of crypsis. As a result, these areas started to regenerate with more biodiverse plant communities leading to an increase in beavers, who are known to be ecosystem engineers. When the beaver number increased, the number and diversity of insects, reptiles, fish and amphibian species increased. Wolfs also killed small predators like coyote; hence the pronghorn and small mammals such as rabbits and mice increased, resulting in more hawks, weasels, badgers, foxes and eagles. Scavengers such as bears increased not only because there was more left-over food by wolfs, but because there were more plants for them to eat due to regenerated crops. The bears reinforced the impact of wolfs by killing some of the calves of the elk. The most interesting observation due to all these biodiverse changes was the rivers' behaviour change, such as water flow. In conclusion, the reintroduction of wolves transformed the ecosystem of the YNP and its physical geography.

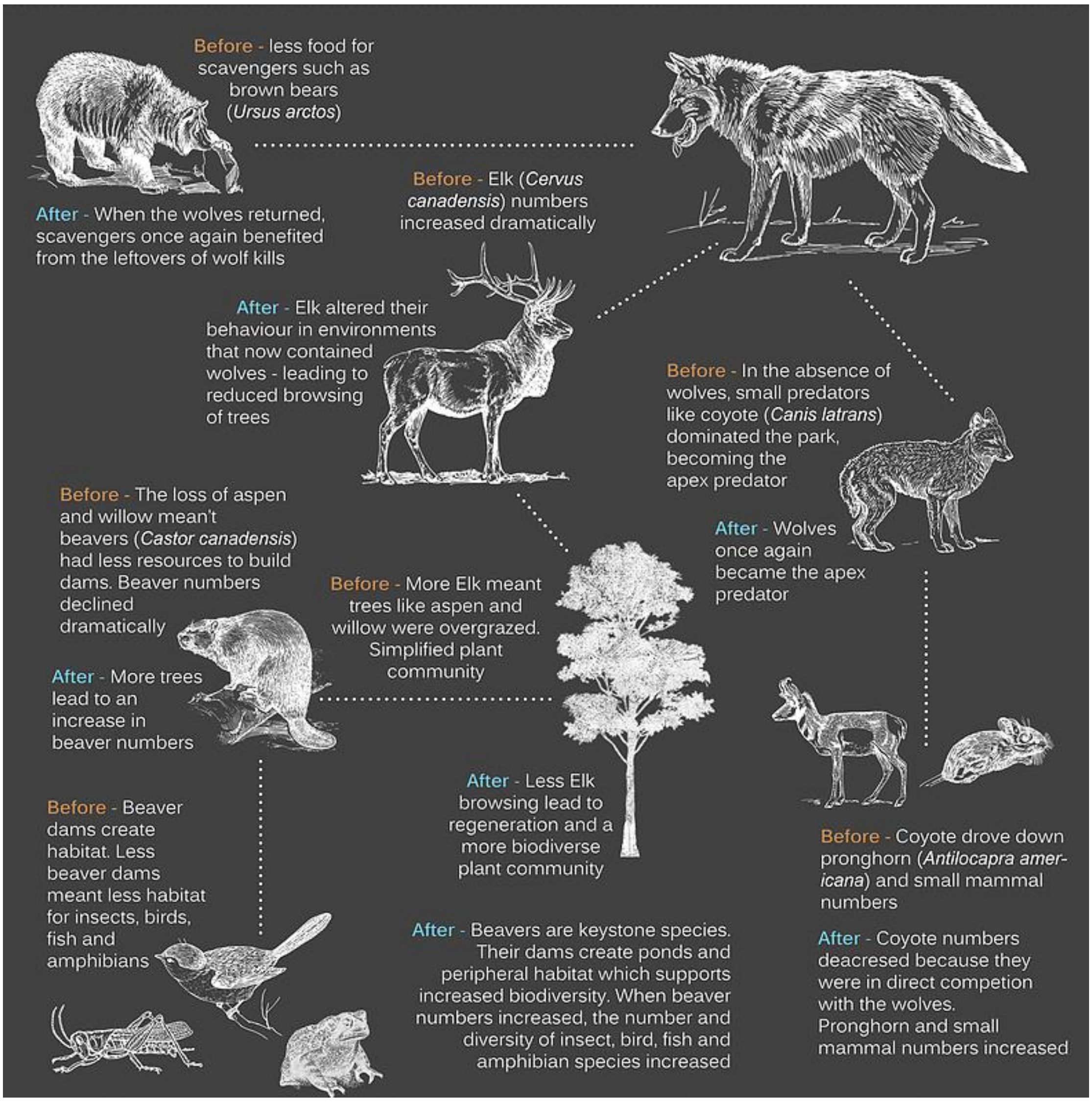



Figure 3.2 Trophic Cascade – an example from Yellowstone National Park: The reintroduction of grey wolves into Yellowstone National Park in the USA is a classic example of a terrestrial trophic cascade. The wolf's absence had a huge impact on the park's ecology. Elk populations began to rise, and subsequent overgrazing had a knock-on effect on other organisms. The image describes how the park's ecology changed before and after the reintroduction of wolves (image: (*Eco Sapien - An Infographic Exploring Yellowstone National Park...*, 2015)).

## Homeostasis

The previous example highlighted equilibria that exist externally and at a macro level. However, internal equilibrium, known as homeostasis, is prevalent and crucial for all living beings. Homeostasis refers to maintaining chemical and physical conditions such as body temperature, fluid state, chemical concentrations etc. in steady-state inside living systems for its optimal functioning. The internal chemical and physical conditions that are essentially needed to be maintained include body temperature, fluid balance (amount of water), pH level, blood sugar level, blood oxygen level and concentration of calcium, sodium, potassium, iron, copper.

The equilibrium is maintained by a dynamic process. That is, regulation feedback is brought about as a response to any change of the above variables that are already in its optimal range. Individual regularization mechanisms include a) receptor(s) to sense changes in the variable being monitored and trigger action potential for the regularization feedback in response to a substantial change in the condition, b) a control centre that sets the desired range to maintain the equilibrium and c) effector(s) that carry out compensatory actions to bring the variable to an equilibrium state. For example, osmoreceptors in the median preoptic nucleus work as the receptors for the fluid level in human beings. At the same time, the hormone system known as the renin-angiotensin-aldosterone system initiates the regularization feedback. The compensatory action is carried by kidney acting as the effector, which reabsorbs water to reduce water loss as urine.

There have also been efforts to formalize homeostasis from the theoretical perspective of thermodynamics and systems theory (Bailey, 1984; Recordati & Bellini, 2004). They describe internal homeostasis from a thermodynamic perspective as a stationary state of nonequilibrium since equilibrium in thermodynamic is well and strictly defined. They formalize the actual state of rest, $\beta(t)$, as

$$\beta(t) = \beta_s + \rho(t) \tag{3.1}$$

where $\beta_s$ is the time-independent steady-state of reference and $\rho(t)$ is the time-dependent fluctuations of the state variables. They note that, amongst these states of rest, quiet wakefulness seems to be only locally stable. In contrast, sleep stages III and IV are the globally stable and has the nearest thermodynamic equilibrium.

## 3.2 Equilibrium in Human Society

### Economic Equilibrium

Another example of a system that demonstrates ideal/close to ideal equilibrium characteristics is the macroeconomy. Competitive equilibrium is the traditional formulation of ideal equilibrium conditions valid for markets with flexible prices and a large number of buyers and sellers. In a competitive equilibrium, economic variables remain unchanged as the economic forces such as the supply and demand for goods are balanced. That is, the economic equilibrium occurs at the point where the quantity of goods supplied equals the quantity of goods demanded. Moreover, the prices of the goods are established by this equilibrium state and referred to as the competitive price. This ideal equilibrium condition assumes that each buyer/seller decides upon only a small quantity compared to the overall volumes of the market-leading to no/minimal influence on the overall price.

Hux Dixson (2001) identifies three properties of economic equilibrium; a) The behaviour of agents is consistent, b) No agent has an incentive to change its behaviour, and c) Equilibrium is the outcome of some dynamic process. Competitive equilibrium satisfies all three conditions. Since the supply equals demand, the property (a) is fulfilled. At the market price, neither the buyers nor the sellers have any incentives to demand/supply any more/less as the change of equilibrium price negates any gains. This satisfies the property (b). Finally, competitive equilibrium satisfies property (c) as there would be downward/upward pressure on the market prices in the case of an excess/short supply leading to self-adjustments in the market to bring the economy back the equilibrium state.

While competitive equilibrium is the ideal version of the economic equilibrium, it is more theoretical than practical due to its hard assumption of individual traders' influence. Research discipline named ‘game theory’ has been advanced to model the equilibrium shown in macro-economic environments. John Nash (1950), who worked on game theory, showed that the economy would still be in an equilibrium state even in the absence of hard assumptions of competitive equilibrium.

Each trader in the economy is trying to optimize given its anticipation of others’ strategies. This was shown to lead to an equilibrium - now referred to as Nash equilibrium - which is a configuration of strategies - one for each trader - such that no trader gains by unilaterally changing its strategy. More formally, if there are $n$ traders, where each trader $i$ $(i = 1 \dots n)$ has a strategy space $S_i$, the economy is a function,

$$g: S_1 \times \dots \times S_i \times \dots \times S_n \to \mathbb{R}^n \tag{3.2}$$

and $g_i(s_1, \dots, s_i, \dots, s_n)$ is $i^{th}$ trader’s profit when strategies $(s_1, \dots, s_i, \dots, s_n)$ are followed. John Nash proved that there exists a configuration of strategies $(s_1^*, \dots, s_i^*, \dots, s_n^*)$ such that,

$$\forall\, s_i \in S_i, \qquad g_i(s_1^*, \dots, s_i^*, \dots, s_n^*) > g_i(s_1^*, \dots, s_i, \dots, s_n^*), \quad i = 1 \dots n \tag{3.3}$$

That is, no trader profits more by unilaterally deviating from the equilibrium strategy $s_i^*$ to $s_i$.

## 3.3 Equilibrium in an HCDE

As explained from the above examples, it is evident that 'natural equilibrium' exists in natural and human-driven environments. The undisturbed environment is ordered and harmonious, and the systems return to previous equilibrium after disturbances. The ability to understand the natural systems and artificial environments were studied extensively by Jay Forrester of Massachusetts Institute of Technology in the 1950s. Jay W. Forrester (1971) proposed a methodology and a mathematical modelling technique to frame, understand, and discuss complex issues, especially in artificial systems. This approach aims at understanding the non-linear behaviour of complex systems over time using stocks, flows, internal feedback loops, table functions and time delays.

Sterman (2001) proposed an extension to System Dynamics which focuses on human-driven complex systems (such as a living being, a corporation, a city, an economy or an ecosystem) where a small change in one element can produce a big change in the whole system. Sterman (2001) elaborates how change, such as the effects of information technology or the effects of greenhouse gases on the global climate, is transforming our world. Some of these changes amaze and delight us, but others impoverish the human spirit and threaten our survival. All too often, well-intentioned efforts to solve pressing problems provoke feedback and create unforeseen reactions. Some of the examples of such situations are California's failed electricity reforms (Joskow, 2001), road building programs that create suburban sprawl and increase traffic congestion (Downs, 1999) and failed change initiatives in organizations (Sterman, 2001).

The classical natural equilibrium view, however, has not been studied with respect to the current HCDE. The current digital environment is a globally dynamic infrastructure that refers to connectivity with their internal and external environments (Devi & Rukmini, 2016). This environment consists of physical objects called "things" connected through information technology. The recent advances in this digital environment have accelerated the deployment of billions of interconnected, smart and adaptive applications in critical infrastructures such as healthcare, transportation, environment control and home automation (Stoyanova et al., 2020; Zou et al., 2020).

The HCDE bridges the gaps between the physical and digital environment and will represent the disturbance present in the natural environment. To understand the digital environment, we need to study the information flows in the feedback relationships. As a change in the natural/human-driven environment occurs, surrounding information changes. Flows of information in the digital environment are thus analogous to flows of matter and energy in natural environments. Due to this correspondence, a 'Digital Equilibrium' can be described using the theories of system dynamics.

## 3.4 A Conceptual Model for Understanding Digital Equilibrium using for Self-Structuring AI (SS-AI)

The above section explains how natural equilibrium maintains any natural system stable with continuous changes. In this section, we look at how the equilibrium is maintained in an HCDE. The system structure represented by the natural behaviours and events holds the same underlying structure for an HCDE. As explained in the previous sections, it is important to understand the system structure to get a better idea of how the system works and as a result being able to predict the environment. Therefore, we assume and conceptualize that the HCDE comprises of digital representations of the natural events and digital representations of natural behaviour (Figure 3.3). Similar to how we sense the natural environment with our five senses, an artificial being observes the environment with digital data. In an HCDE, digital data is characterised by high volume, velocity and variety. Therefore, the detection of digital representation of the natural events will need to be facilitated by artificial intelligence. Similar to how events accumulate into behaviours, digital representation of the natural events is assumed to be accumulated into a digital representation of natural behaviour. This digital representation of natural behaviour will reveal information about the system structure and understanding the system structure provides the base for prediction.

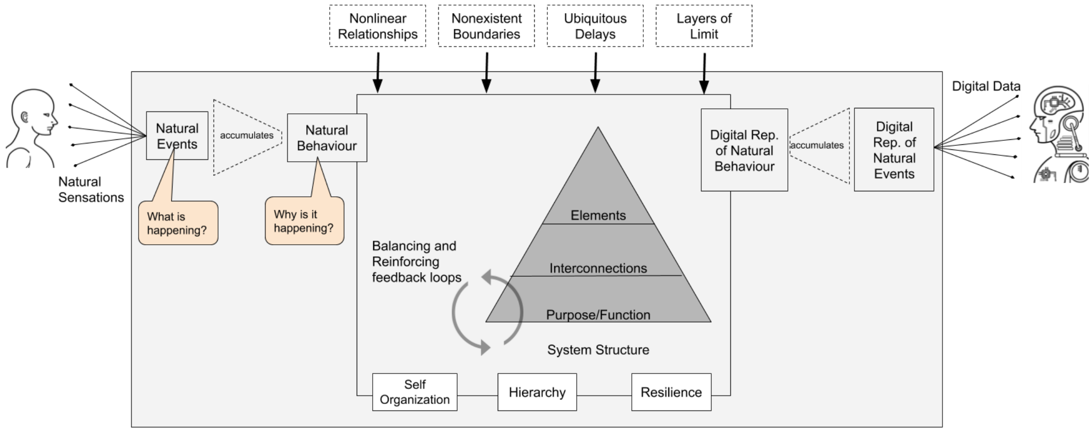



Figure 3.3 Conceptualization of a Digital System: a digital representation of the natural environment.

With the advent of HCDE, there are many examples of digital systems such as smart homes, smart city, industry 4.0. These systems generate dynamic, unlabelled, continuous data in the form of structured and unstructured digital data streams.

When learning from data streams, the following challenges need to be considereded,

- Volatility – continuously and autonomously evolving data streams,
- Velocity – real-time analysis from data generated from IoT, text, image and video streams,
- Volume - access, integrate, store/process analyse the massive amount of data,
- Variety - manage different types of data and varied data formats such as structured, unstructured, semi-structured,
- Veracity – create reliable data as the basis for data-driven decision making by filtering, validating, profiling and cleansing.

On top of these, as an HCDE consists of digital representation of the natural environment, digital systems inherit the properties of the natural environment. Therefore, the properties of a system: self-organization, hierarchy and resilience can be used to build a machine learning technique to detect the natural events and interpretation of these events automatically. Resilience represents the ability of the components in the digital environment to adapt from a change or disturbance. Self-organization property organizes the digital data into concepts in the feature space. The hierarchy represents the arrangement of objects or behaviours in the digital environment linked directly and indirectly. Artificial intelligent algorithms can be implemented using self-organization, hierarchy and resilience as properties of the environment.

Understanding the system structure through digital data face major complexities due to the challenges imposed on a system structure such as non-linear relationships, non-existent boundaries, ubiquitous delays and layers of limit. The relationships comprehended in data streams are non-linear. These challenges make it challenging to analyse the digital environment without the use of artificial intelligence.

These challenges and complexities are addressed by Self-Structuring AI (SS-AI) (Figure 3.4). Self-Structuring AI is an emerging paradigm of artificial intelligence defined as intelligence structures that autonomously evolve based on the unstructured and unlabelled nature of data, spatially, temporally, laterally, and semantically (De Silva et al. 2020). In SS-AI, the digital representation of the natural events will be captured through concept changes, and digital representation of natural behaviours captured through sequences and causality. Concept changes will denote ‘what’ is happening in the environment, and sequences and behaviours will provide an understanding of ‘why’ it is happening.

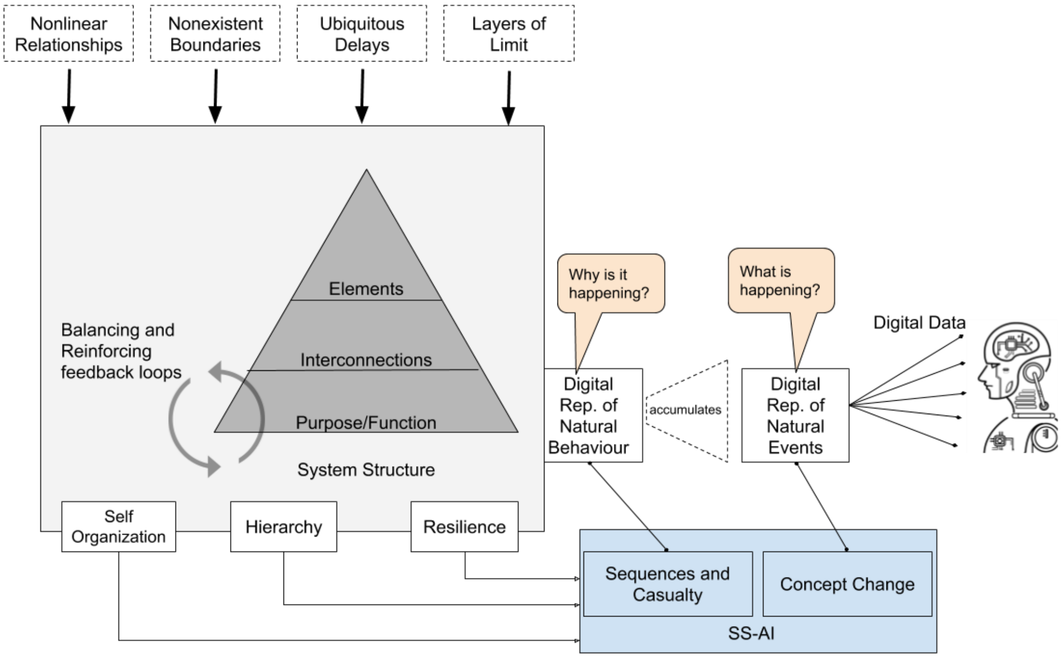



Figure 3.4 Conceptual model for understanding digital equilibrium using Self-Structuring AI (SS-AI)

**Self-Structuring AI for Concept Change**

To detect the concept changes, a human observes a change in the natural environment as events and an artificial system or an artificial being observes the natural environment through digital data. Changes in elements, interconnections or purpose/function (the system structure) are manifested as an event. These changes impact the environment at different levels. The digital representation of a natural event can be represented as a concept change. Duration of an event or a concept change cannot be pre-determined, and time of an event or a concept change cannot be pre-defined.

The digital environment comprises of data streams. As explained in chapter 2, one major challenge in working with data streams is their evolving nature, i.e., the concept at a given time, $t$, will evolve to another concept at a time, $t + \alpha$ (Lam & Mostafa, 2001). This is known as concept change, or concept drift in literature. The majority of existing work on concept change detection is based on supervised learning applied in traditional environments where the data are infrequent, small, isolated, sparse and labelled. The change detection algorithm monitors the accuracy of the underlying predictive model, and a reduction of the accuracy (lower than a certain threshold) is detected as a concept change. Detection of a concept change will trigger the predictive model to retrain. This implementation is not feasible in the HCDE where large volumes of data are executed in high velocity, connected, dense and unlabelled. Therefore, this thesis aims at developing an unsupervised method for change detection and adaptation (Figure 3.5).

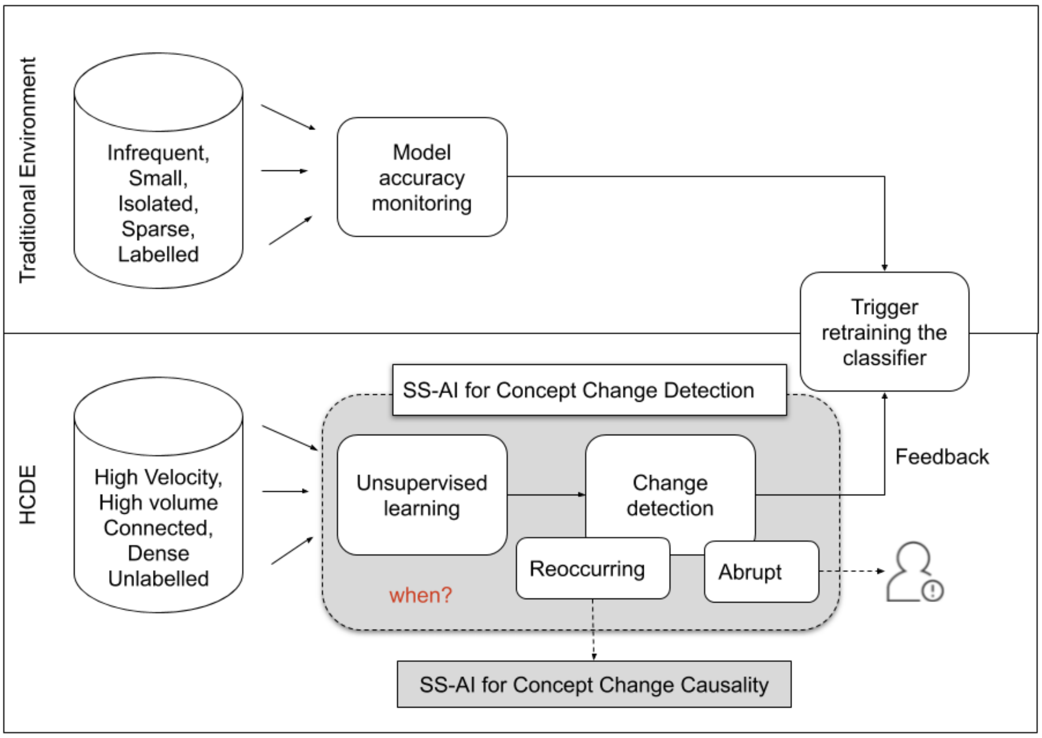



Figure 3.5 Self-Structuring AI (SS-AI) for Change Detection in HCDE. In a traditional environment, concept change is detected through monitoring the classifier accuracy. An HCDE, detection of a concept change triggers retraining of the classifier. Alerts can be generated by distinguishing abrupt concept changes from reoccurring concept changes. Reoccurring concept changes are further used in Self-Structuring AI (SS-AI) algorithm for concept change causality.

As shown in Figure 3.5, it is important to distinguish between reoccurring and abrupt concept changes. Reoccurring concept changes allows us to know the general patterns in the environment, hence apply general rules in the environment. For example, in a big data driven smart city traffic environment, understanding peak/off-peak travel patterns will facilitate traffic light management. This pattern would depend on many dynamic factors such as school day, holidays, city events and weather. The behaviour pattern provides an understanding of natural behaviours and is further examined in this thesis. Abrupt concept changes allow us to detect sudden changes in the environment which would need quick responses. For example, in the smart city traffic environment – a road accident will affect the traffic flow not only in the surrounding environment but possibly the whole system. Abrupt changes will need further attention from a human. Feedback is sent to the underlying supervised learning algorithm (classifier) when a concept change is detected. Concept change detection allows the supervised learning algorithm to trigger retraining to facilitate successful and efficient adaptation to the new concept.

**Self-Structuring AI for Sequences and Causality Detection**

The pattern of behaviour can be determined based on an event or a concept as natural events accumulate into system behaviour. Similarly, digital representation of the natural behaviour provides information on the underlying system structure and can be predicted using sequences and causality.

Perceiving the sequence of events or concepts provides an understanding of the behaviour of an individual component in the environment. For example, in a smart city traffic environment, there is a peak and off-peak behavioural pattern. The sequence of behaviours defines the temporal dynamic of the environment and explains why a change occurred. An AI algorithm is proposed to detect the sequence of behaviour when a concept change is detected (Figure 3.6). The algorithm creates a stabilized sequence tree that will learn from reoccurring concept changes in individual data streams and would explain 'why' a concept change in the data stream occurred. The proposed algorithm will output a stabilized sequence tree at each time $t$.

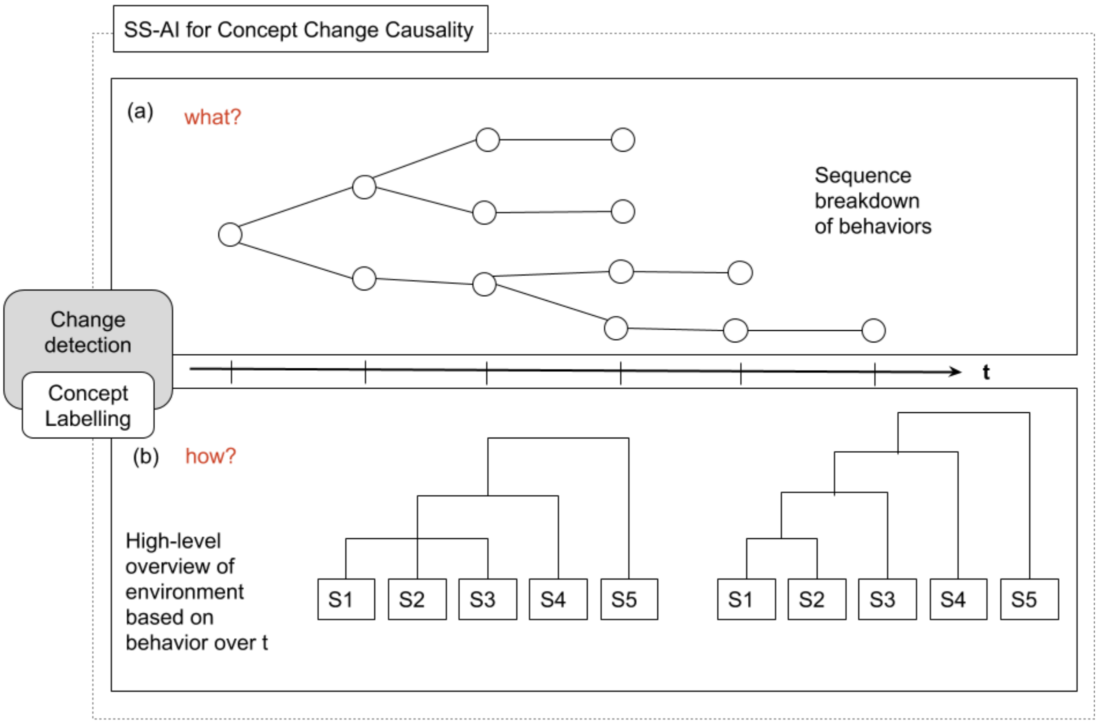



Figure 3.6 Self-Structuring AI (SS-AI) for Concept Change Causality. $t$ is the time a concept change is detected. (a) sequence breakdown of the behaviours – each point is a unique event/action which is stabilized over time based on the repetition of behaviours. (b) behavioural tress provides an abstract view of the environment – S1, S2, S3, S4 and S5 are five different data streams and figure demonstrated the distance/similarity between each data stream over time. For example: at $t = 3$, behaviour of S1, S2 and S3 are similar than S4 and S5; at $t = 6$, behaviour of S1 and S2 are still similar, but the behaviour of S3 has changed.

Understanding the causalities in the environment will denote the non-linear dynamic relationships existent within the environment. For example, traffic congestion in one road might affect some neighbouring roads in the road network in a smart city traffic environment. Causality defines the spatial dynamic of the environment and provides an overview of the groupings in the environment, explaining how a change occurred.

An AI algorithm is proposed to execute at time $t + \alpha$ to provide a high-level overview of the environment based on multiple data streams (Figure 3.6). The algorithm creates a behavioural tree that indicates the similarity or distance between different data streams and explains 'how' the concept change influences the overall environment.

# 3.5 Chapter Summary

The chapter elaborated on the natural equilibrium as the inspiration to understand an HCDE. Further, this chapter presented examples of natural equilibrium in eco-systems, homeostasis, and economic equilibrium to study the information flows in the feedback relationships.

An HCDE bridges the gaps between the physical and digital environment and will represent the disturbance present in the natural environment. As a change in the natural/human-driven environment occurs, surrounding information changes. Therefore, the chapter justified the need for equilibrium in HCDE called ‘digital equilibrium’ and conceptualized the equilibrium in an HCDE as opposed to the natural environment. The proposed conceptual model used Self-Structuring AI to detect digital representation of natural events and behaviours.

# Chapter 4

# A Self-Structuring AI Algorithm for Change Detection

This chapter proposes the Self-Structuring AI algorithm for concept change detection outlined in chapter 3. Self-Structuring AI is defined as intelligence structures that autonomously evolve based on the unstructured and unlabelled nature of data, spatially, temporally, laterally, and semantically (De Silva et al., 2020). The proposed algorithm is based on three learning features that are fundamental for concept change detection from unlabelled data streams. They are, 1) incremental learning, 2) decremental learning, and 3) online learning. With these learning features, the proposed algorithm facilitates unsupervised, self-adaptive learning in unlabelled big data streams. As explained earlier in chapter 3, the chapter will distinguish between reoccurring and abrupt concept changes. This will allow further actions or automation, as explained in chapter 6.

The chapter is organized as follows. Section 4.1 presents how the change detection can be used in the digital environment and provides a high-level design for automatic detection of concept changes in a data stream. Section 4.2 described the learning paradigms; incremental, decremental and online learning of the proposed technique. Section 4.3 explains the incremental knowledge acquisition and self-learning (IKASL) algorithm, upon which the proposed algorithm built on, in detail. Section 4.4 provides a detailed description of the proposed change detection algorithm, where abrupt and reoccurring change detection is distinguished. Section 4.5 presents the empirical evaluation using benchmark SEA dataset.

# 4.1 Learning Paradigms of the Proposed Algorithm

The proposed technique is based on three learning paradigms that are fundamental for concept change detection from unlabelled data streams. They are, 1) incremental, 2) decremental, 3) online learning.

## 4.1.1 Incremental Learning

Incremental learning is necessary for learning from data streams as it effectively addresses both time and memory constraints (Furao & Hasegawa, 2006; Mouchaweh et al., 2002; Navarro-Gonzalez et al., 2015). Incremental learning methods do not require an initially labelled dataset for training since they continue to learn from an incoming data stream. They assume that the hypotheses (source, concept, distribution, etc.) learned before are always valid for the new incoming data (Sayed-Mouchaweh, 2016). Therefore, incremental learning can be used to adapt to a known concept and detect concept change by differentiating between previously known concepts and new concepts.

IKASL (De Silva & Alahakoon, 2010) is an implementation to facilitate incremental learning, preserve acquired knowledge and apply knowledge gained for subsequent learning. Learning outcomes of past data form the foundation for self-organization of new data. The actualization of incremental learning in this manner is computationally reasonable due to low resource requirement for maintaining hit node lists from past learning. Achieving these aims enables the IKASL algorithm to overcome the stability-plasticity dilemma. IKASL algorithm is described in detail in section 4.4.

### 4.1.2 Decremental Learning

Decremental Learning is used to unlearn (to forget) representations of the data stream, which are no longer relevant (Gama et al., 2014). Learning from data streams should be continuous while preserving the previously known useful knowledge. Natural cognitive systems gradually forget previously learned information. Plausible models of human cognition should, therefore exhibit similar patterns of gradual forgetting of old information as new information is acquired (French, 1999). Only rarely does new learning in natural cognitive systems completely disrupt or erase previously learned information. That is, in general, natural cognitive systems do not forget 'catastrophically'. However, catastrophic forgetting does occur under certain circumstances in distributed connectionist networks. The same feature gives these networks their remarkable ability to generalize, to function in the presence of degraded input etc.

Neuronal rigidity of this nature leads to an unstable network (stability dilemma) in a continuous learning environment. An unstable network starts forgetting learned patterns (catastrophic interference) resulting in rigid/partially forgotten (plasticity dilemma) learning outcomes. Therefore, to overcome the limitations of data stream analysis, translating from the biological process of learning an adaptive learning algorithm should be able to learn from new data without requiring access to previous data, preserve acquired knowledge, unlearn knowledge that is no longer relevant, accommodate new learning outcomes and relate these to previously acquired knowledge.

### 4.1.3 Online Learning

Data streams tend to generate data at high speed and in large quantity. This can become a limitation for an offline iterative machine learning process where the whole training data must be available at the time of model training. In online learning, algorithms process input data sequentially as the data become available, and the model is continuously updated. It is important to continuously update machine learning outcomes as concepts evolve over time. Online learning can be incorporated with incremental and decremental machine learning to keep up with high frequency and high-velocity data streams.

## 4.2 Incremental Knowledge Acquisition and Self-Learning (IKASL) Algorithm

The Incremental Knowledge Acquisition and Self-Learning (IKASL) algorithm is an unsupervised, incremental learning algorithm that continues to learn new data based on generalized layers of past learning outcomes. It has been successfully demonstrated on social media text mining (Bandaragoda et al., 2017) and smart electricity meter data for pattern classification and demand forecasting (D. De Silva et al., 2011a, 2011b, 2011c). Incremental learning in IKASL is initiated by aggregation of unsupervised machine learning outcomes with the formation of generalization layers. Each generalized node expands into its own feature map to generate a topological representation of subsequent input vectors.

Morphologically, the IKASL model is an n-layer network structure, with n periods of incremental learning. The layers are virtual; they are not predefined and come into existence as required by the incremental learning process. Each layer is composed of two sub-layers, learning, $L_n$ and generalization, $G_n$.

The functionality of the learning layer is based on the GSOM algorithm (Alahakoon et al., 2000). The first learning layer, which is also the starting layer of the process, generates a dynamic feature map based on the growing, self-organizing process. The corresponding dataset is fed into the network, Starting with four randomly initialized output nodes. For each input, the closest output node is selected as the winner using the Euclidean distance measurement. The weight vector of the winner and its neighbourhood are adapted to reflect the win.

The weight adaptation rule is shown in Equation 1.

$$\Delta w_i = \rho \emptyset (r_j, r_{j*})(x_i^m - w_i) \tag{4. 1}$$

where;

| | |
|---|---|
| $\mathrm{i}$ | the number of attributes, |
| $\mathrm{x}^m$ | m-th input vector from a set of vectors X, |
| $\mathrm{w_i}$ | i-th weight of the winning node, |
| $j_*$ | index of the winning node, |
| $\rho$ | a predefined weight update ratio, |
| $\emptyset(\mathrm{r_i}, \mathrm{r_{i*}})$ | the neighbourhood function |

Separately, the error value of the winning node is also increased to distinguish it from the rest of the network. Node growth occurs when this error value exceeds a predefined growth threshold, $GT$. New nodes grow out of the node with the highest accumulated error, $N_e$ and are initialized to reflect the neighbourhood of $N_e$. Upon learning for a predefined number of epochs, a calibrating phase smooths out irregularities in recent weight adaptations. Learning outcomes of the dynamic feature map can now be identified.

(a) Primary learning outcome: The knowledge embodied in weight vectors of winning nodes. Taking the weight vector $w$ as the representative vector, the primary learning outcomes of the dynamic feature map can be shown as an integration, $k_w$,

$$k_w = \int_v^0 \propto (|x - w|)^2 p(x) dx \tag{4. 2}$$

where $V = U_{i \in S} V_i$ is the input data space with $S$ vectors, p(x) the probability distribution of input vectors $V$ and $\propto$ as the learning rate.

(b) Secondary learning outcome: Identified as a proportionate learning outcome, knowledge embodied in weight vectors of the winner's neighbourhood nodes. Given the proximity to the winning node, these vectors represent the variation of the primary learning outcome (i.e., the corresponding feature) as per the data space. These outcomes can also be shown as, $K_{nn}$,

$$K_{nn} = \int_{v}^{0} \propto \text{ŋ}\ (v,k)(|x - w|)^2 p(x)dx \tag{4. 3}$$

with ŋ $(v,k)$ is a representation of the proportion of learning being dispersed to the neighbourhood nodes.

Thereby, the total knowledge acquired by a dynamic feature map is expressed as per Eq. 4. It is $K_{Tot}$ that should be considered to maintain continuity in learning and not $K_w$ which is only part of learning outcomes.

$$K_{Tot} = \{K_w, K_{nn}\} \tag{4. 4}$$

The identified primary and secondary learning outcomes are combined into a single structure to form the generalization layer, $G_n$. Representing primary and secondary learning outcomes in a simple data structure which can form the basis for further learning, is the key purpose of the generalization layer. Aggregation refers to the process of combining values into a single outcome which takes into account, in a given fashion, all input values (Beliakov et al., 2007). The existence of a weak order relation on the set of all possible values is the minimal requirement to be satisfied to perform aggregation. An aggregation operation considers several aspects, such as the expected outcome from the aggregation operation, the nature of values to be aggregated and the type of scale being used.

The fuzzy integral has been selected as the aggregation function for the generalization layer, mainly due to its natural framework for inclusion of behavioural properties, such as the ability to express the importance of criteria, the behaviour of the decision-making requirement (tolerance, intolerance of criteria fulfilment) and interaction between criteria (redundancy and reinforcement of multiple criteria) (Yager, 1993). The fuzzy integral's nonlinear approach to combine multiple sources of information (which is reflective of human non-linear decision making (J. Li et al., 2004)) and ability to handle multiple sources (Lühr & Lazarescu, 2009) are further contributions. The fuzzy integral considers both objective evidence supplied by various sources and the expected worth of subsets of these sources in the fusion process (Beliakov et al., 2007).

Each node in the generalization layer has the potential to grow into a feature map. The subsequent learning phase, $L_{n+1}$, will start by identifying, for each input vector, the winning node from the generalization layer. Instead of generating a single map, each generalized node expands into its own feature map to generate a topological representation of the corresponding input vectors. The winning nodes, primary and secondary learning outcomes of the phase $L_{n+1}$ will be distributed among these maps. The following aggregation process will produce a $G_{n+1}$ generalization layer from the outcomes of the phase $L_{n+1}$. $G_{n+1}$ will form the basis for $L_{n+2}$ learning; the two phases will thus continue until all data sequences have been processed/learned.

When a generalized node is not the winner for any of the inputs in the dataset of the subsequent learning phase, the non-utilized node is added to the list of nodes in the new generalization layer. This allows the non-utilized node to continue learning in the current phase, even though it does not represent any inputs from the subsequent phase (D. De Silva & Alahakoon, 2010). The capacity to accommodate non-utilized nodes for further learning enables the algorithm to preserve knowledge, regardless of its relationship to the data space presented in current learning. Associations between nodes in the generalization layers will be persistent, leading to creating a memory-like structure based on the aggregated outcomes of the learning stages.

The main features of the algorithm are,

1. A continuous, self-learning algorithm.
2. A dynamic structure for acquisition and preservation of learning outcomes.
3. Generation and sustenance of computationally efficient, generalized representations of learning outcomes.

4. Regulation of generalized representations as for the basis for subsequent learning.

## 4.3 The HCDE Context for the Proposed Algorithm

In this section, we introduce an overview of the proposed Self-Structuring AI algorithm for concept change detection. As outlined in section 4.1, the algorithm uses online, incremental, and decremental learning for concept change detection and distinguishes between abrupt and reoccurring concept changes.

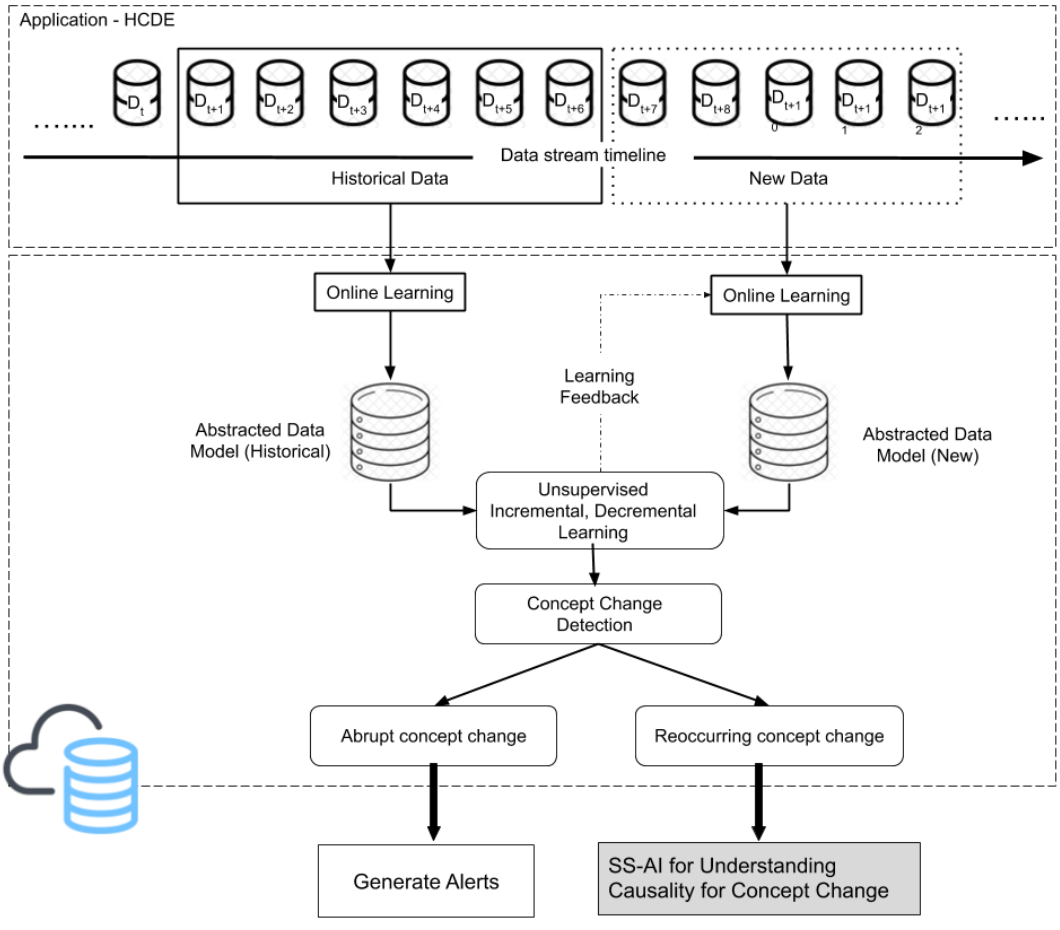



Figure 4.1 The HCDE context for the proposed Self-Structuring AI algorithm for concept change detection.

As shown in Figure 4.1, a data stream generated by an application in HCDE is monitored by the Self-Structuring AI algorithm executed in a cloud setting. The online learning algorithm captures the data in a time window defined by the algorithm. The output of the online learning algorithm is presented to unsupervised incremental and decremental learning. The incremental and decremental learning output is used for concept change detection and as learning feedback to the next learning iteration. A significant movement in feature space between the outputs of two corresponding learning iterations is detected as a concept change. A concept change detected is distinguished between abrupt and reoccurring. Detection of abrupt concept change is alerted to humans for further actions. Detection of reoccurring concept change is used to understand the concept change using Self-Structuring AI algorithm for concept change causality explained in chapter 6.

## 4.4 Proposed Self-Structuring AI Algorithm for Concept Change Detection

We build on the success of the IKASL algorithm by advancing it into decremental learning and online learning for continuous detection and adaption to concept change from an unlabelled data stream. A variation of this technique was applied to explore the importance of context awareness to estimate road traffic (Nallaperuma et al., 2017), investigate the impact of driver behaviour change on the coordination between self-driven and human-driven vehicles (Nallaperuma et al., 2018), and as the core machine learning function of an expansive, intelligent traffic data integration and analysis platform (Nallaperuma et al., 2019). The proposed algorithm consists of three primary functions, 1) online learning, 2) incremental and decremental learning and 3) concept change detection (Figure 4.2). Each function is discussed below, alongside its algorithmic representation.

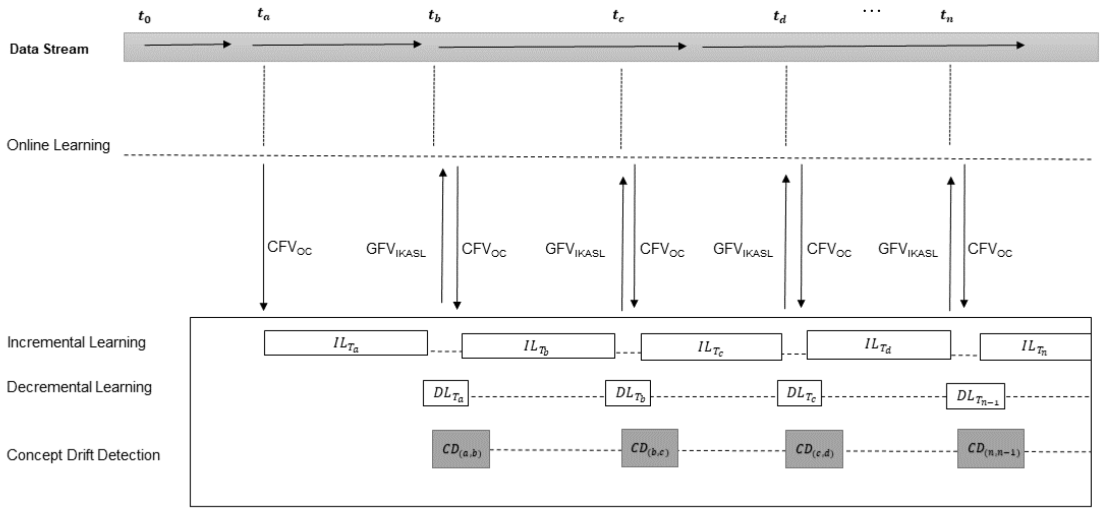



Figure 4.2 Proposed Self-Structuring AI algorithm for concept change detection

**Online learning:** Online k-means clustering is used for one-pass online learning for efficient one-pass processing of a stream of data rather than storing and processing in batches (Câmpan & Şerban, 2006). In the first iteration $k$ (number of cluster feature vectors) and $t_a$ (processing time of initial learning iteration) are user-defined for online k-means and the generated cluster feature vectors ($CFV_{OC}$) are input to the offline IKASL function. In subsequent iterations, $k$ is the number of cluster feature vectors ($\#GFV_{IKASL}$), $t_\theta$ (e.g. $t_b$-$t_a$) is the time taken by IKASL for the learning process, and cluster feature vectors for online k-means are the generalized nodes received from the IKASL function. These automated $k$ and $t_\theta$ implements the nonparametric nature of the algorithm.

**Initialization of the algorithm**

1: $CFV_{OC} \leftarrow$ random cluster feature vectors
2: Initialize starting node ($\aleph_i$) of the IKASL algorithm with random values for weight vector ($w_i$), zero for error value ($e_i$) and zero for hit count ($h_i$).
3: $k \leftarrow$ the initial number of cluster feature vectors, $t_a \leftarrow$ processing time of initial learning iteration

**Online learning – algorithmic representation**

4: **for** $d_i \in D$ **do**
5: $k^* \leftarrow \arg min_1^k (\| d_i - (CFV_{OC})_k \|)$
6: **if** $\| d_i - (CFV_{OC})_{k^*} \| \geq \emptyset$ **then**
7: $(CFV_{OC})_{k+1} \leftarrow d_i, k \leftarrow k + 1$

**else**
Update $(CFV_{OC})_{k^*}$
**end if**
**end for**

**Incremental and decremental learning:** IKASL learning occurs as per the original algorithm for incremental learning. Inputs are batches of $CFV_{OC}$ received periodically from the online learning function (Fig 2). We further extended the IKASL function to facilitate decremental learning by forgetting the generalized node that is not the winner of any of the inputs in the dataset of the subsequent learning phase. In this case, the generalized node is forgotten, indicating the concept has changed or evolved. Associations between nodes in the generalization layers will be persistent, leading to the creation of a memory-like structure based on the aggregated outcomes of the learning stages. Adaptation to a new concept is formalized with the incremental and decremental learning.

**Incremental Learning – algorithmic representation**

**while** $D \neq \emptyset$
**for** $x_i \in CFV_{OC}$ **do**
Select winner as $n^* \leftarrow \arg min_1^n (\| x_i - w_n \|)$
Calculate updated weights as

$$w_j(t+1) = \begin{cases} w_j(t)\ , j \notin N_{n^*}(t+1) \\ w_j(t) + LR(t)\left(x_k - w_j(t)\right), j \in N_{n^*}(t+1) \end{cases}$$

$e_{n^*} \leftarrow e_{n^*} + \| x_i - w_n \|$
$h_{n^*} \leftarrow h_{n^*} + 1$
**end for**
**if** $e_i \geq GT$ **then**
**if** *i* is a boundary node **then**
Grow the map by creating a new node
Initialize the new node vector $w_{(new)}$ to match the neighbouring node weights
**else**
Distribute weights to neighbours
**end if**
**end if**
$LR \leftarrow$ reduce $LR$
$N \leftarrow$ reduce neighbourhood size
**end while**
Select Hit Nodes $H_i \subseteq \aleph,\ s.t. h_i \geq hitT$

31: Calculate the proximity matrix, $S$, where $s_{km}$ contains the proximity of $n_k(H_i)_m$, the $m^{th}$ node of the $k^{th}$ neighbourhood to the corresponding hit node, $H_i$.

32: $S = (s_{km}) \in \mathbb{R}^{u \times v}$, $s_{km}$ is calculated as $s_{km} = |a_q - b_q| \, \forall q \in \mathbb{N}$, where $a, b$ are weight vectors of $n_k(H_i)_m$ and $H_i$ respectively and $q$ the position vector.

**Decremental Learning – algorithmic representation**

33: **if** $h_i < hitT$, $\forall i \in \mathbb{N}$ **do**
34: forget node $i$ from the map
35: **end if**
36: **if** $GFV_{IKASL}(t) \neq \emptyset$ **do**
37: $CFV_{OC}(t+1) \leftarrow GFV_{IKASL}(t)$
38: **end if**

**Concept change detection:** Concept change detection is carried out by calculating the distance between generalized nodes ($CFV_{IKASL}$) of consecutive iterations. The algorithm is sufficiently generic for any distance measure to be used, such as Euclidean distance (Yanhong Li et al., 2014; Tran, 2013), heterogeneous Euclidean overlap distance (Sobhani & Beigy, 2011; Tsymbal & Puuronen, 2000), Mahalanobis distance (Gonçalves Jr. et al., 2014; Toubakh & Sayed-Mouchaweh, 2015), Hellinger distance (Ditzler & Polikar, 2011). As a concept change occurs, there would be a significant distance change, followed by a reduced distance change in the following iteration. Concept changes detected are further classified by the algorithm as abrupt concept change and reoccurring concept change.

**Concept Change Detection – Algorithmic Representation**

39: Calculate $ED_t \leftarrow$ distance measure between $GFV_{IKASL}(t)$ and $GFV_{IKASL}(t-1)$
40: $List(CD) \leftarrow$ list of $ED_t$
41: **if** $ED_t > ED_{t-1}$ **and** $ED_t > ED_{t+1}$ **do**
42: Identify $(t)$ as an occurrence of Concept Change, $CD_t$
43: $List(CD) \mathrel{+}= ED_t$
44: **end if**

**Abrupt Concept Change Detection – Algorithmic Representation**

Calculate $mean(CD)$
**for** $CD_t \in List(CD)$ **do**
  identify neighbourhood
    $t' = t - 1$
    $t'' = t + 1$
    **if** $CD_t < CD_{t'}$ **do**
      Calculate $prominence \leftarrow |CD_{t'} - CD_t|$
      **if** $prominence > mean(CD)$ do
        Identify$(t)$ as Abrupt Concept Change, $CD_t^A$
      **end if**
    **else if** $CD_t < CD_{t''}$ **do**
      Calculate $prominence \leftarrow |CD_{t''} - CD_t|$
      **if** $prominence > mean(CD)$ **do**
        Identify$(t)$ as $CD_t^A$
      **end if**
    **else**
      expand the neighbourhood
      $t' = t' - 1$
      $t'' = t'' + 1$
    **end if**
**end for**

**Reoccurring Concept Change Detection – Algorithmic Representation**

$Fs = 1$
$Nf = 512$
$df = \frac{Fs}{Nf}$
$f = 0 : df : \frac{Fs}{2} - df$
$treSpots = fftshift(fft(CD_t - CD_t^A) - mean(CD_t), Nf)$
$dBspots = 20 \times log10 \left( abs \left( trSpots \left( \frac{Nf}{2} + 1 : Nf \right) \right) \right)$
$highestFreq \leftarrow Sort(dbSpots, DESC)$
$\#CD^R = 1/f(highestFreq)$
Identify $(t)$ as Reoccurring Concept Change, $CD_t^R$ every $\#CD^R$

$highestFreq \leftarrow Sort(dbSpots, DESC)$

# 4.5 Demonstration

SEA dataset (Street & Kim, 2001), a benchmark dataset widely used in supervised concept change detection, was used to demonstrate features of the proposed algorithm.

Figure 4.3 SEA dataset – Concept Change Detectionillustrates concept changes detected from the SEA dataset. The x-axis denotes timestamps of incremental learning, and distance measure (in this case Euclidean distance, $ED_n$) calculations from step 5 of the algorithm are denoted on

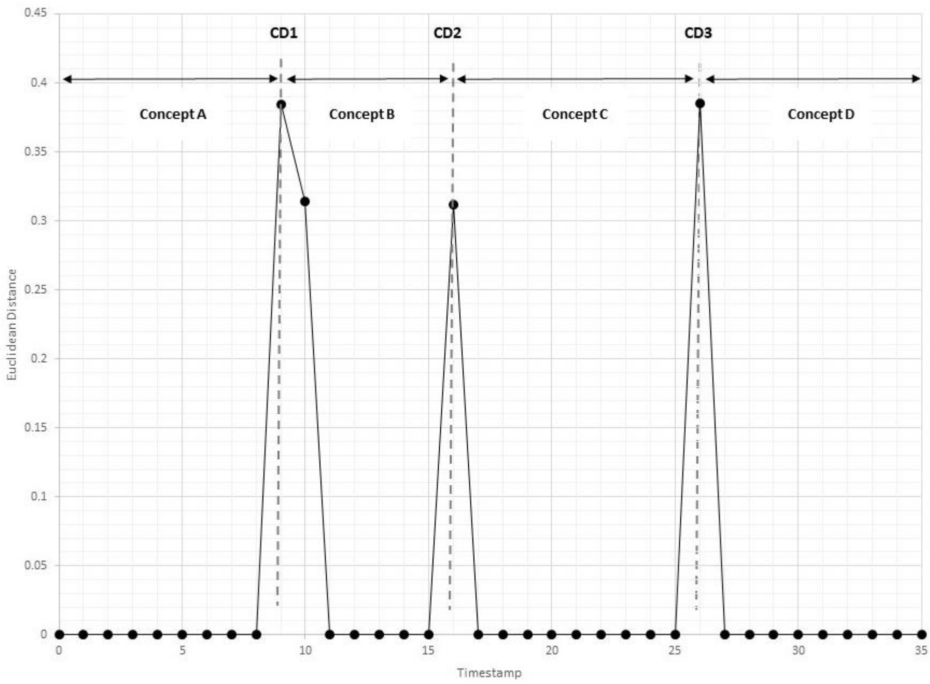



Figure 4.3 SEA dataset – Concept Change Detection

the y-axis. Abrupt concept changes were detected at timestamps 2, 9, 16 and 26 with $ED_n$ 0.43, 0.39, 0.31 and 0.39 respectively. Results were validated with concept changes detected in the same dataset in (Street & Kim, 2001) and (Bifet et al., 2010).

To demonstrate the importance of real-time concept change detection, the accuracy of a supervised predictive algorithm with and without concept change detection was compared (Figure 4.4). For the latter case, the algorithm was trained with first 1000 records, and the trained model was used to test the data in each subsequent batch of 1000 records. The accuracy of the algorithm reduces as the concepts evolve over time (Figure 4.4). For the former case, the algorithm was trained and retrained at each concept change detection with the most recent 1000 records. The accuracy of the algorithm improves as the algorithm was re-trained with the evolved concepts.

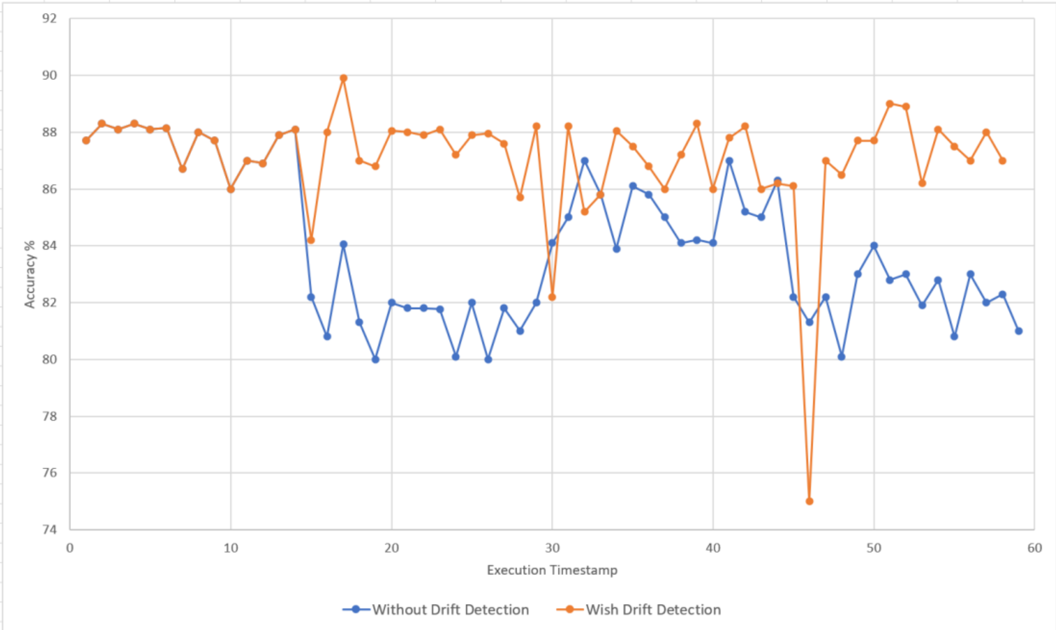



Figure 4.4 Support Vector Machine Accuracy with and without Concept Drift Detection

### 4.5.1 Demonstration with Modified SEA Dataset

An advantage of the SEA dataset generator is that it can be configured to generate data with the repetition of the same four concepts to evaluate the identification of reoccurring concepts. For this demonstration, we re-generated the SEA dataset with the four concepts repeated three times, creating twelve concept changes (Figure 4.5). This allows us to evaluate how the proposed algorithm performs in learning the concepts that have already been learnt and how that affects the concept change detection over time. The proposed unsupervised algorithm was analysed against the corresponding concept changes shown by MOA (Bifet et al., 2010) (Figure 4.5). All of the twelve (four concepts repeated three times) concept changes were identified by the proposed algorithm (Figure 4.6) and directly corresponded to the MOA output. Concept changes were identified at execution timestamps; [t2], [t4], [t7], [t9], [t12], [t14], [t16], [t18], [t20], [t23], [t5], [t28].

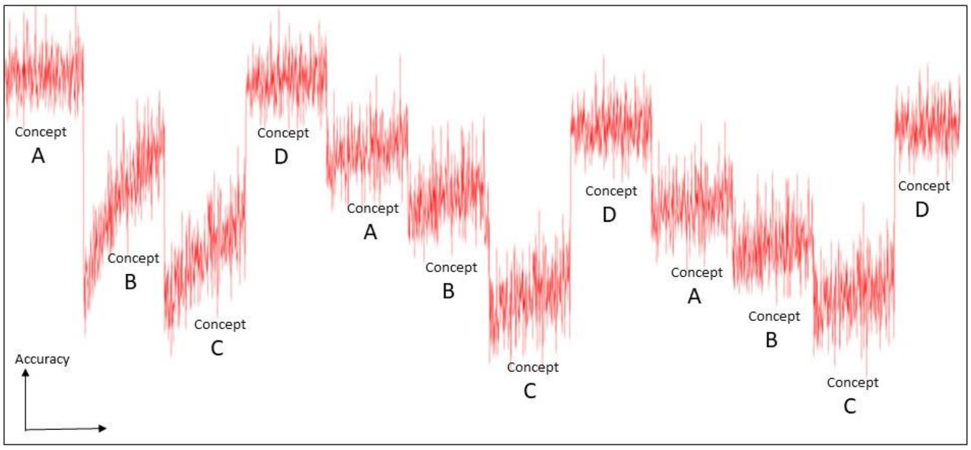



Figure 4.5 MOA (Bifet et al., 2010) output for modified SEA dataset

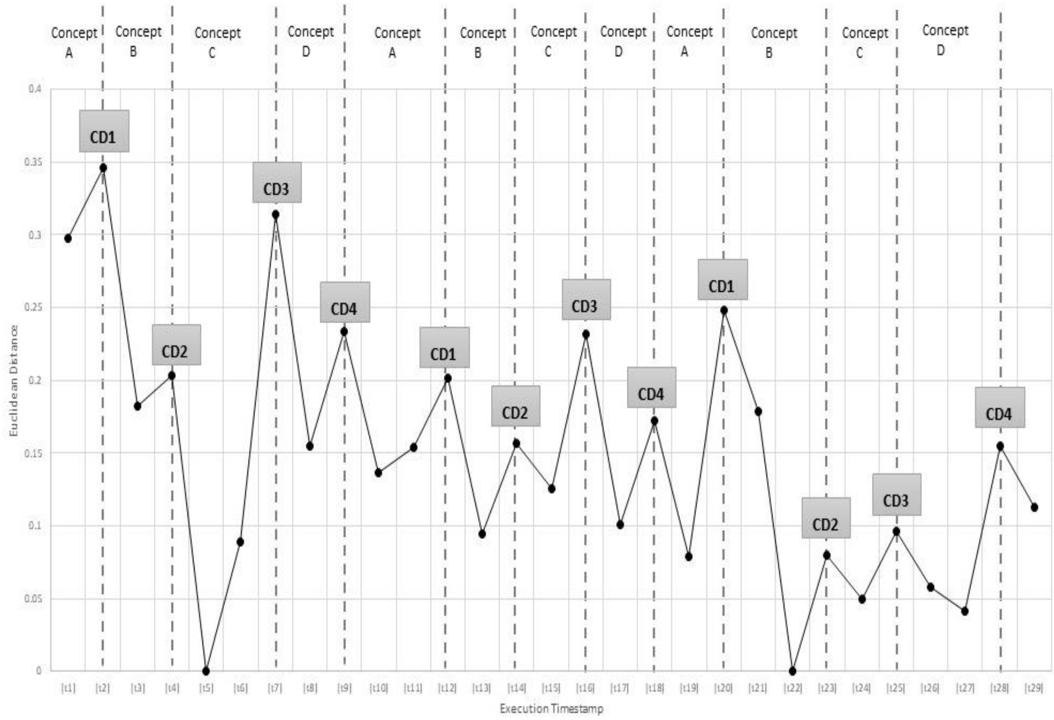



Figure 4. 6 Concept Change Detection for Modified SEA dataset

As shown in Table 4-1, time taken to detect a reoccurring concept change reduces over time, demonstrating the incremental nature of learning.

Table 4-1 Automated Time Window Analysis

| | **SEA dataset 1** | | **SEA dataset 2** | | **SEA dataset 3** | |
|---|---|---|---|---|---|---|
| | Concept Drift | Execution Time (ms) | Concept Drift | Execution Time (ms) | Concept Drift | Execution Time (ms) |
| **Concept A** | [t2] | 295 | [t12] | 285 | [t20] | 279 |
| **Concept B** | [t4] | 271 | [t14] | 267 | [t23] | 258 |
| **Concept C** | [t7] | 247 | [t16] | 239 | [t25] | 234 |
| **Concept D** | [t9] | 244 | [t18] | 234 | [t28] | 230 |

## 4.6 Chapter Summary

The chapter presented a novel Self-Structuring AI algorithm for change detection in data streams of an HCDE. The proposed algorithm uses three learning paradigms: online learning to handle high volume and velocity of data present in HCDE, incremental learning to provide the ability to learn new concepts, and decremental learning to forget the concepts that are no longer relevant. The proposed Self-Structuring AI algorithm was built upon the success of incremental learning of the IKASL algorithm by advancing it to support decremental learning and online learning for continuous detection and adaption to concept change from an unlabelled data stream in HCDE. The chapter provided a detailed algorithmic description of the proposed algorithm. The proposed Self-Structuring AI algorithm executes on automated time windows set by the algorithm, detects change based on the movement of feature space and determine the type of concept change (abrupt or reoccurring) based on the movement of time. The chapter demonstrated the proposed algorithm of SEA benchmark dataset.

# Chapter 5

# Empirical Evaluation of the Proposed Algorithm

This chapter presents the empirical evaluation of the Self-Structuring AI algorithm for change detection proposed in chapter 4. The algorithm was applied to six scenarios that are representatives of HCDE settings, including four real-world datasets in air traffic, smart energy, physical activity monitoring, smart city traffic, and two real-world case studies on the arterial road network of VicRoads and the first annotated driving recordings of self-driving cars.

## 5.1 Air Traffic

The dataset consists of 116 million records of flight arrival and departure details of all commercial flight details within the USA from October 1987 to April 2008. The dataset is represented by 13 input attributes: year, month, day of the month, day of the week. CSR departure time, CSR arrival time, unique carrier, flight number, actual elapsed time, origin, destination, distance and diverted and the target variable is arrival delay (as multiple target variables: diverted, carrier delay, weather delay, security delay) given in seconds.

The input data in the dataset was simulated as a continuous data stream, and the proposed algorithm learnt as the data are presented (as described in section 4.4). In each iteration of learning, distance measure (Euclidean Distance is used in this scenario) is calculated as explained in the algorithm (Step 38 - 43). A segment of the output is used in this section to demonstrate the occurrence of an abrupt concept change ($CD_t^A$) in the airline data stream. The algorithm detects an abrupt concept change at timestamp-33 (t=33), $CD_{33}^A$. There has been a significant distance variation between t=32 and t=33, followed by a reduced distance variation at t=34 (Figure 5.1). As there is an increased distance followed by a reduced distance, it confirms that a concept change has occurred, not an outlier.

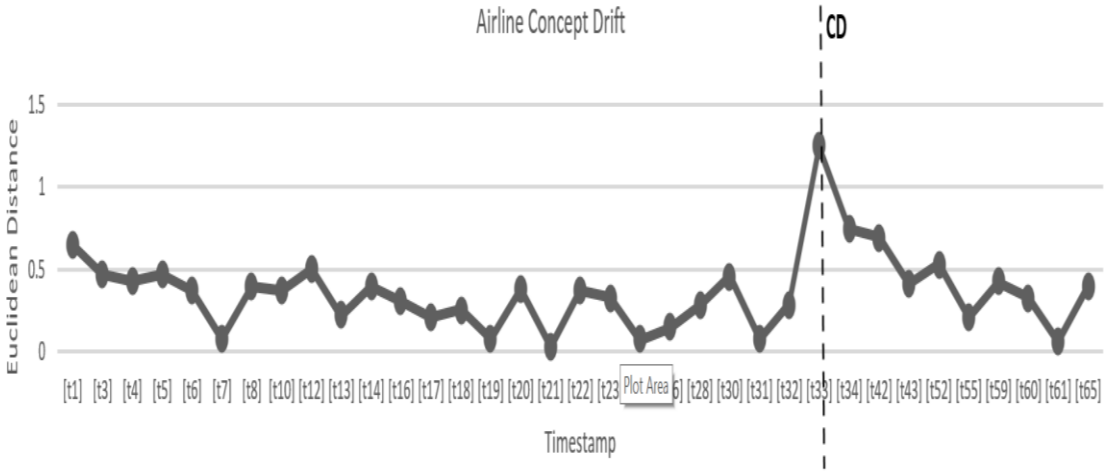



Figure 5.1 Airline dataset: Concept Change Detection

To analyse the causality for the concept change, the variance of the input variables and target variables at the time of the concept change occurrence was investigated. We could observe that the input variable, arrival time, has a change in the density denoting an increase in the number of flights (Figure 5.2). This has been denoted in target variables such as diverted carried delay, weather delay and security delay, which might have occurred the increased number of flights arriving at the airports at a particular time (Figure 5.3).

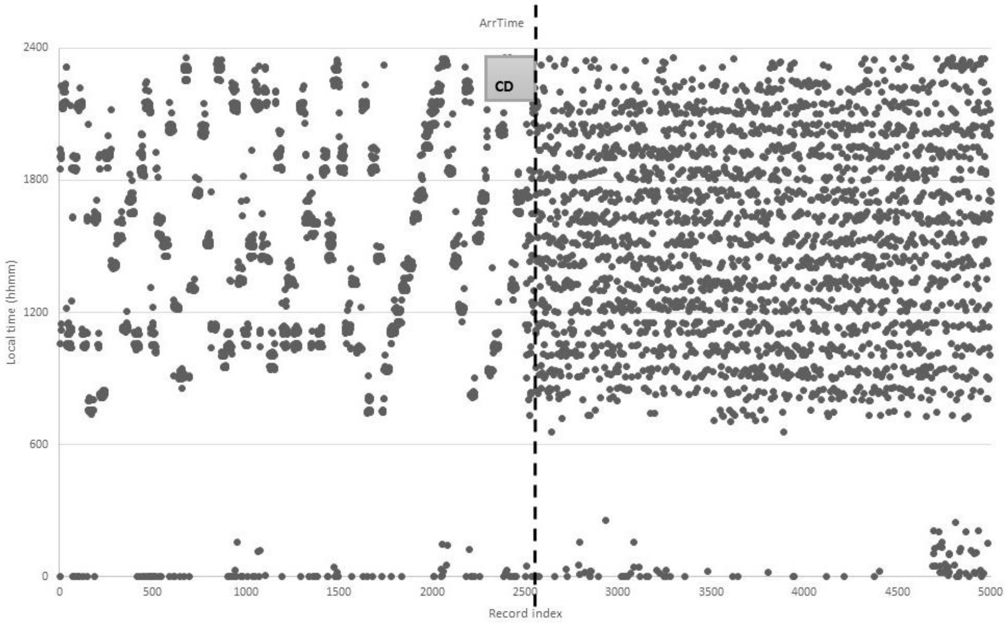



Figure 5.2 Airline dataset: Input Variable Analysis

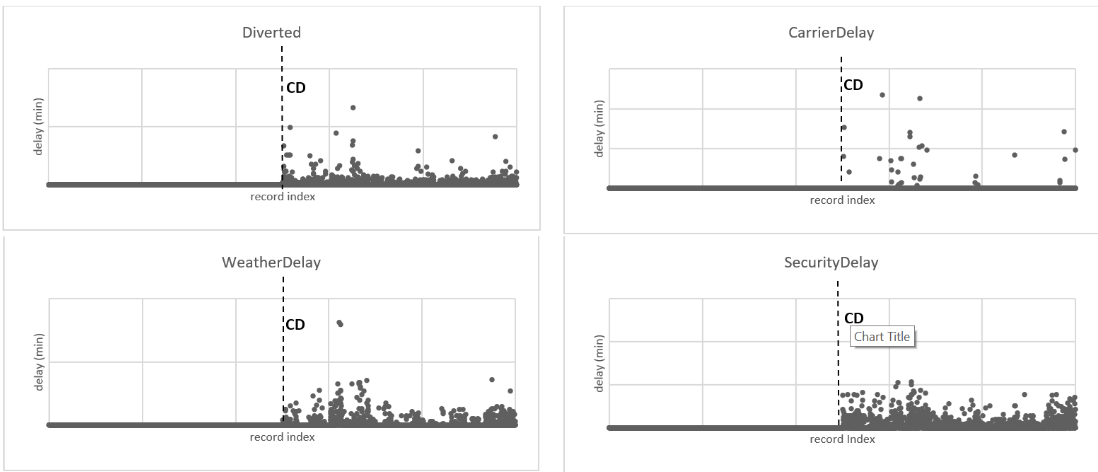



Figure 5.3 Airline dataset: Target Variable Analysis

This experiment demonstrates how the proposed concept change algorithm can be used to detect abnormal traffic build ups in near real-time that could otherwise have resulted in delays.

# 5.2 Smart Energy

The dataset comprises electricity power consumption of a household with a one-minute sampling rate, containing 2075259 records for 47 months (between December 2006 and November 2010). The electricity meter is sub-metered into three, submeter1: corresponds to the kitchen appliances such as a dishwasher, an oven and a microwave, submeter2: corresponds to laundry room appliances such as a washing machine, a tumble dryer, submeter3: corresponds to an electric water heater and an air-conditioner. With this unlabelled dataset, we aim to evaluate the features of the algorithm and detect reoccurring concept changes.

Different concept changes and patterns such as daily patterns (day/night) and seasonal patterns could be identified from the electricity dataset by differentiating the algorithmic parameters such as initial processing time. A daily pattern recognized with concept change detection is demonstrated in Figure 5.4 (top). An explanation of the concept change occurrences, we could see an increase in usage of submeter3 which has occurred a concept change (Figure 5.4 (bottom). Hence, on the extracted day (Sunday, 17$^{th}$ December 2006), events in Table 5-1 are anticipated.

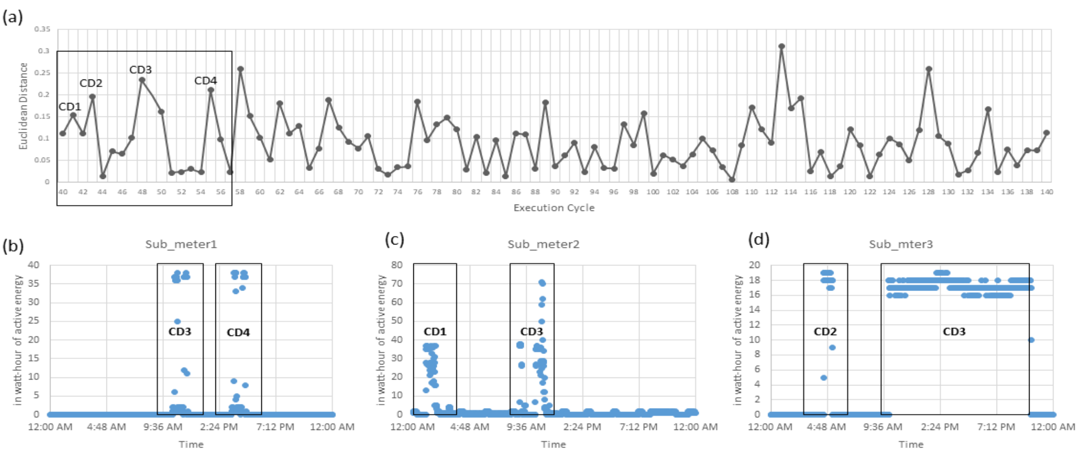



Figure 5.4 Demonstration of concept change detection in Electricity dataset. Top: Concept change detection, Bottom: Hourly electricity usage (segment marked in concept change detection diagram): Sub_meter_1 – kitchen appliances (dishwasher, oven, and microwave), Sub_meter_2 – laundry room (washing machine, tumble dryer, refrigerator and, a light), Sub_meter_3 – electric water heater and air-conditioner

Table 5-1 Electricity dataset: event mapping on concept change

| Concept Change | Event |
|---|---|
| CD [40] | Use of washing machine overnight. |
| CD [43] | Use of heating |
| CD [48] | Preparation of breakfast, use of the dryer and heating. |
| CD [55] | Preparation of lunch |

The application of proposed concept change algorithm in smart energy environment will allow detection of any usual activity such as malfunctioning of an electrical equipment as well as to profile regular power consumption patterns.

## 5.3 Physical Activity Monitoring Dataset

The PAMAP2 Physical Activity Monitoring dataset (Reiss & Stricker, 2012) consists of data from a heart rate monitor and three inertial measurement units (IMU) worn in hand, chest and ankle. The IMU measures the body part's specific force, angular rate, the magnetic field surrounding the body using a combination of accelerometers, gyroscopes and magnetometers. The data were collected from 9 subjects which performing 18 different physical activities; lying, sitting, standing, walking, running, cycling, Nordic walking, watching TV, computer work, car driving, ascending stairs, descending stairs, vacuum cleaning, ironing, folding laundry, house cleaning, playing soccer and rope jumping.

This multivariant time series dataset consists of more than 3.8 million data records with 54 columns; timestamp, 52 attributes of raw sensory data, and the activity label (the ground truth). Data pre-processing is conducted based on (Jayaratne et al., 2017), and new data are engineered to achieve better comprehension based on (Anguita et al., 2013) as follows.

**Data Pre-processing**

- Removal of transient activities which are coded with class '0'.

- Removal of records with missing values due to the loss of wireless network connectivity.
- Disregarding 3D-acceleration data with the scale of ±6g resolution as the readings get saturated with acceleration over 6g during high impact activities such as running.
- Removal of orientation data due to the errors in data collection.
- Interpolation of heart rate values to compensate for the low sampling frequency compared to IMUs. As shown in Table 5-2, the average heart rate increase for each activity is calculated based on the resting heart rate for each participant. Based on the activities, heart rate resembles low-intensity activities such as lying, sitting, standing; moderate-intensity activities such as walking, Nordic walking, cycling; and high-intensity activities such as running, rope jumping.

Table 5-2 Average heart rate increase (Jayaratne et al., 2017)

| **Activity** | **HR increase (bps)** | **Intensity** |
|---|---|---|
| Lying | 9.02 | |
| Sitting | 13.21 | |
| Standing | 22.42 | Low |
| Ironing | 24.34 | |
| Vacuum Cleaning | 37.69 | |
| Walking | 46.46 | |
| Nordic walking | 58.21 | |
| Cycling | 58.85 | Moderate |
| Descending Stairs | 62.94 | |
| Ascending Stairs | 63.01 | |

| | | |
|---|---|---|
| Running | 81.62 | High |
| Rope Jumping | 90.16 | |

**Data Engineering**

- New features are calculated from the triaxial signals of each IMU; triaxial acceleration, acceleration magnitude, triaxial acceleration jerk, acceleration jerk magnitude, triaxial angular speed, angular speed magnitude, triaxial angular acceleration, angular acceleration magnitude and triaxial magnetism.

After data pre-processing and data engineering, 2,872,533 records and 43 attributes are used in the experiments. The records from one subject are processed as a single data stream is used in a single instance of a proposed algorithm.

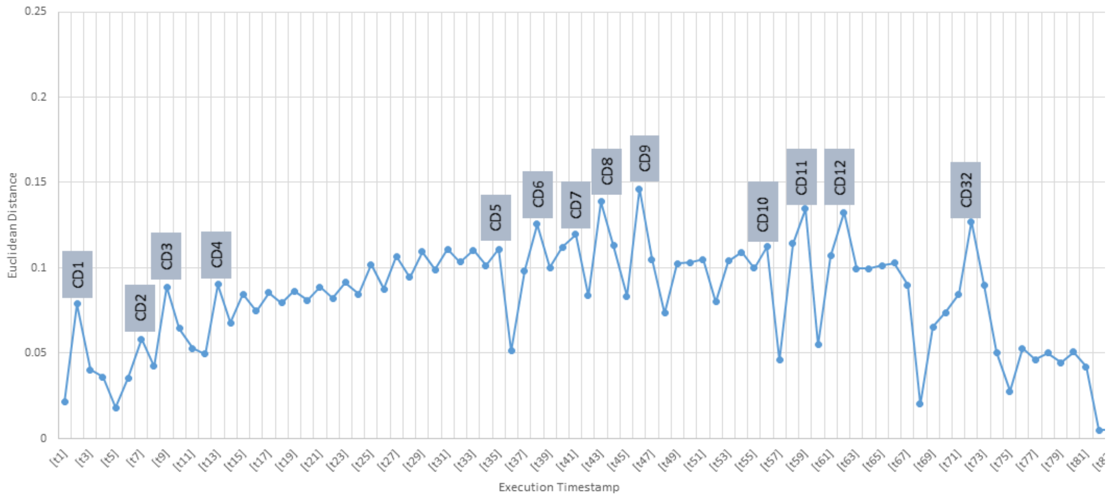



Figure 5.5 Activity Dataset: Unsupervised Concept Change Detection

Figure 5.5 illustrates the concept changes detection from User 2. Each detected concept change was mapped to a labelled activity which denotes high accuracy of the algorithm. Also, concept change was detected in the same time window the new data representing the new activity received by the algorithm which denotes the almost real-time detection. CD6 and CD8 were identified as a reoccurring concept change, which was confirmed by the labels ‘Ascending stairs → Descending stairs’.

Table 5-3 Activity mapping for Concept Change Detection

| Concept Change | Activity Change |
| --- | --- |
| **CD1** | Lying → Sitting |
| **CD2** | Sitting → Standing |
| **CD3** | Standing → Ironing |
| **CD4** | Ironing → Vacuum Cleaning |
| **CD5** | Vacuum Cleaning → Ascending Stairs |
| **CD6** | Ascending Stairs → Descending Stairs |
| **CD7** | Descending Stairs → Ascending Stairs |
| **CD8** | Ascending Stairs → Descending Stairs |
| **CD9** | Descending Stairs → Walking |
| **CD10** | Walking → Nordic Walking |
| **CD11** | Nordic Walking → Cycling |
| **CD12** | Cycling → Running |
| **CD13** | Running → Rope Jumping |

Further, the multi-dimensional generalization nodes (explained in chapter 4) is visualized using Sammon-mapping (Thrun, 2018) to understand the concept change detection (Figure 5.6). Each activity is learnt in several execution timestamps and is denoted by several generalization nodes. Generalization nodes mapping to an activity are clustered together, and low-intensity activities and high-intensity activities are separated in the feature space. This maps to the categorization done in Table 5-3. Hence, Sammon's mapping results confirm the learning of the concept change detection and adaptation is accurate.

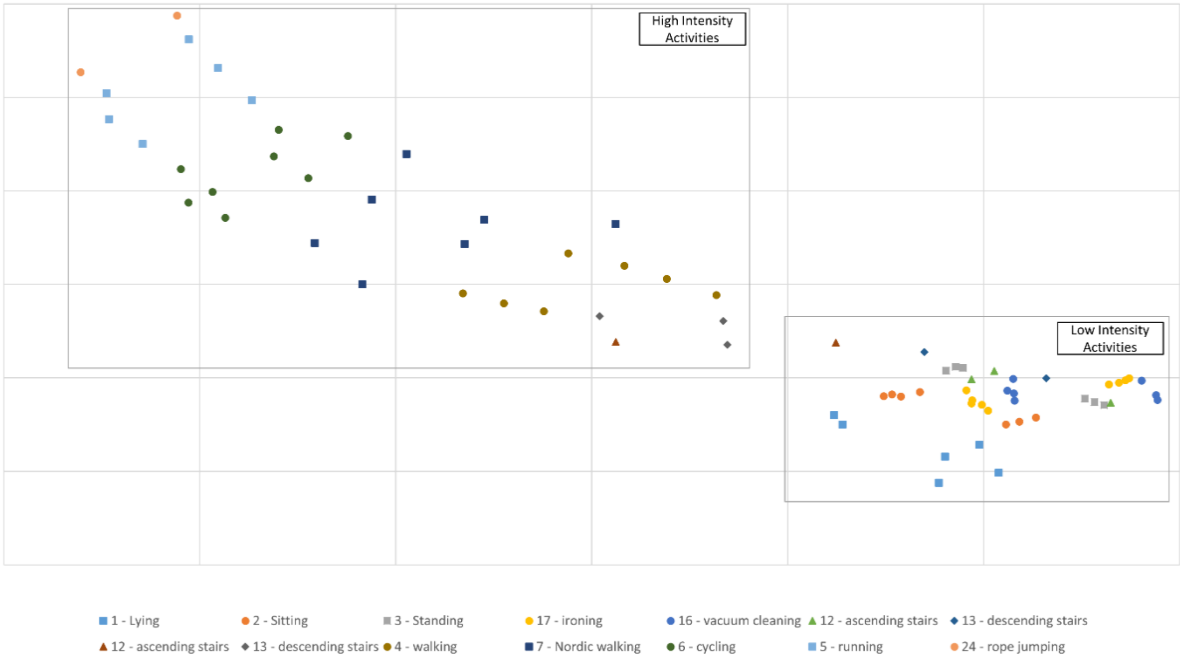



Figure 5.6 Sammon's Mapping of Generalization Nodes

This experiment demonstrates how concept change detection algorithm can be used in a human activity monitoring. An application area would be generating alerts in health care or aged care, where people can be monitored for any unusual activity.

# 5.4 Smart City Traffic

Aarhus city of Denmark smart city traffic publicly available dataset (Kolozali et al., 2014) is generated by the source and destination pairs of sensors placed on various traffic roads in different cities of City of Aarhus. The recorded traffic data estimate the traffic flow between two points in the road providing information regarding the geographical location, timestamp and traffic intensity such as average speed and vehicle count. This setup has generated more than 23 million unlabelled IoT data, recorded every 5 minutes from 449 observation points over a period of 6 months. Results of this experiment have been used to explain the cognitive data stream mining technique for context-aware IoT systems (Nallaperuma et al., 2017).

Data between two observation points are generated as a single data stream input to the proposed algorithm to experiment the use of the proposed algorithm in IoT traffic scenario. As explained in Chapter 4, concept changes were identified as either abrupt or reoccurring.

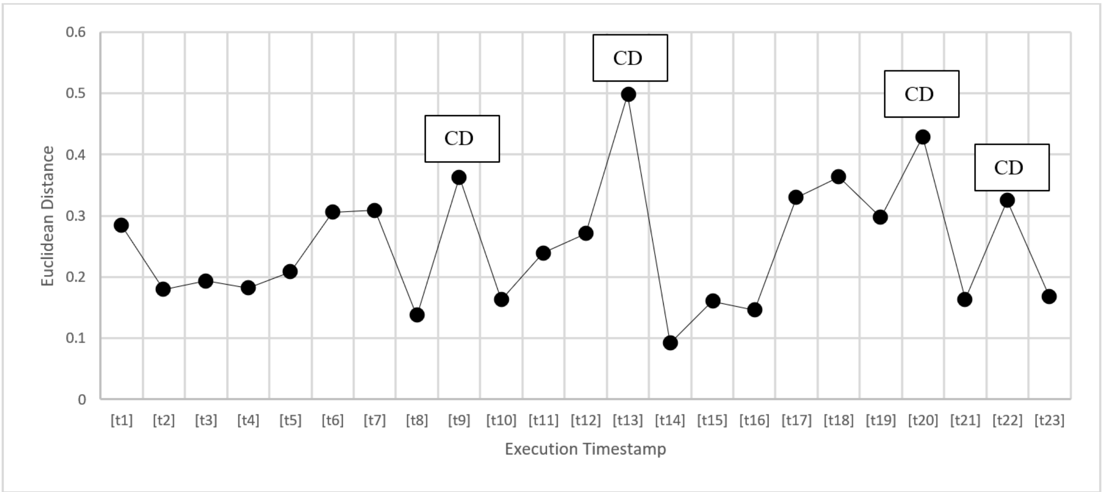



Figure 5.7 Reoccurring concept change detection (CD).

To identify the causes of the concept changes, the number of vehicles - the important contributing attribute - was analysed corresponding to the times of the concept changes (

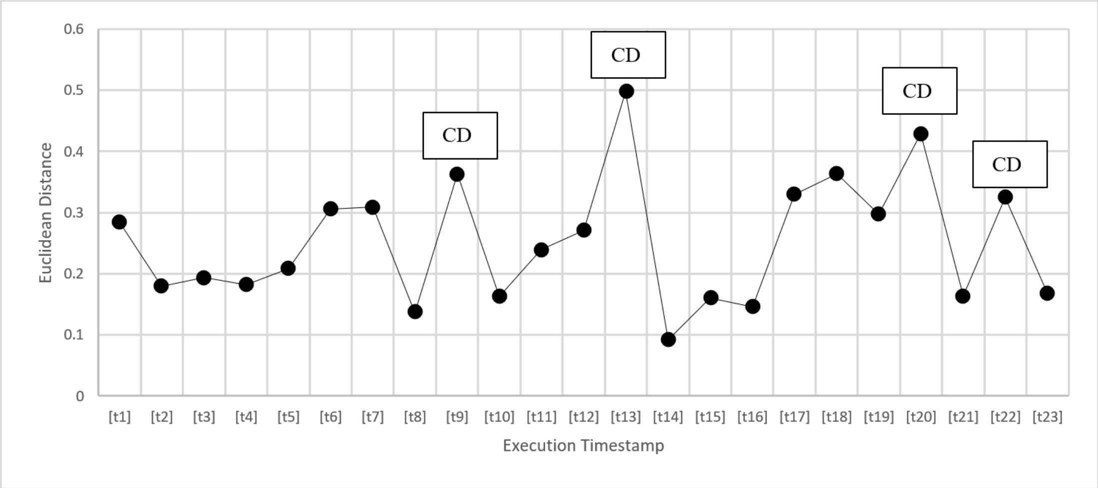



Figure 5.7). It can be observed that from 0000 hours on 7/10/2014 till approximately 0500 hours on 7/10/2014 there was a reduced number of vehicles on this street. The vehicle count has

increased from approximately 0500 hours to approximately 1900 hours on 7/10/2014, resulting in a concept change in timestamp-9 (t9 in

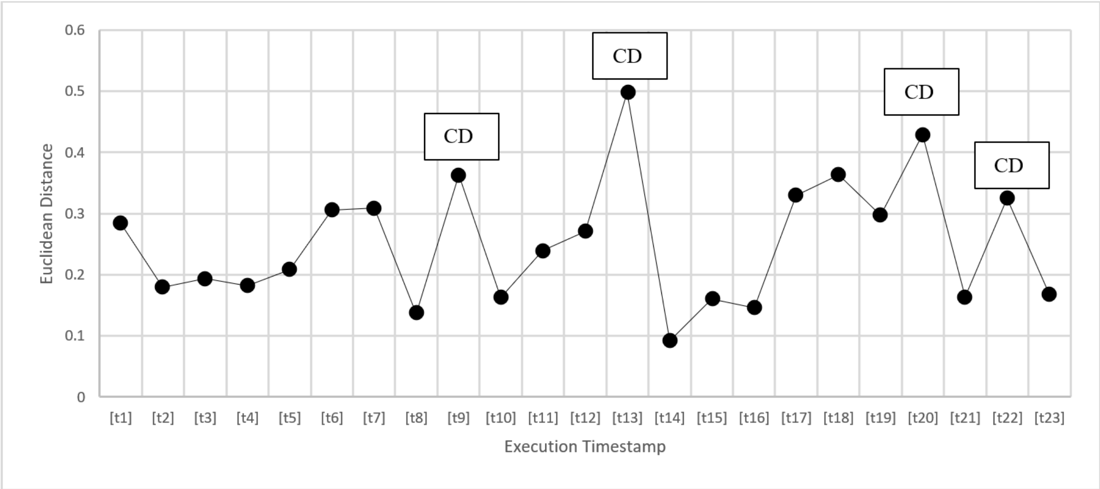



Figure 5.7). Similarly, from approximately 1900 hours on 7/10/2014 to approximately 0500 hours on 8/10/2014, the vehicle count has reduced again, which has resulted in another concept change depicted at timestamp-13 (t13 in

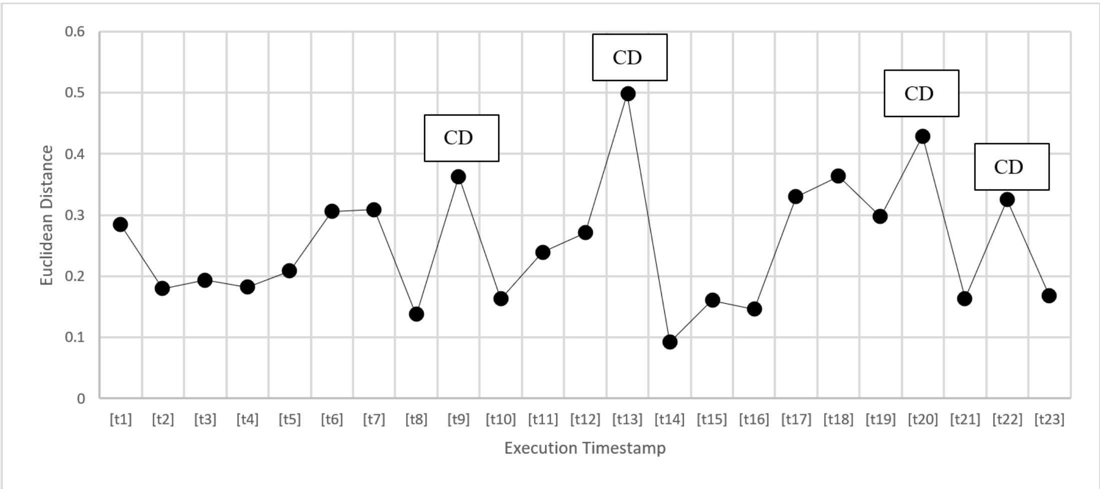



Figure 5.7). An increase in the vehicle count from approximately 0500 hours on 8/10/2014 to approximately 1900 hours has resulted in the concept change at timestamp-20 (t20 in

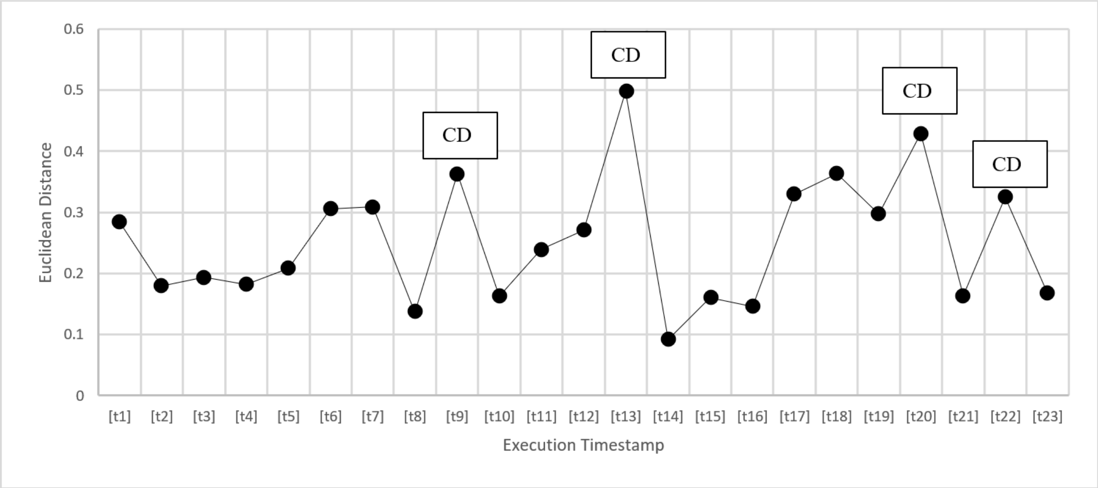



Figure 5.7). These results are summarized in Table 5-4.

Table 5-4 Activity mapping for Concept Change Detection

| **Time Duration** | **General Concept** | **Concept Change?** |
|---|---|---|
| 7/10.2014 00:00 – 7/10/2014 4:48 | A low number of vehicles | |
| 7/10/2014 4:48 – 7/10/2014 19:12 | A high number of vehicles | Concept change at execution timestamp-9 |
| 7/10/2014 19:12 – 8/10/2014 4.48 | A low number of vehicles | Concept change at execution timestamp-13 |
| 8/10/2014 4:48 – 8/10/2014 19:12 | A high number of vehicles | Concept change at execution timestamp-20 |

In a smart traffic scenario, detection of abrupt concept changes almost real-time is very important. For example: if an accident occurs, there would be an abrupt concept change in the traffic condition. Generation of an alert based on the detection of abrupt concept changes (Figure 4.1) can be used for effective decision making in real-time (e.g., reducing traffic congestion by a traffic diversion). An abrupt concept change has occurred in timestamp-14 (t14 in Figure 5.8).

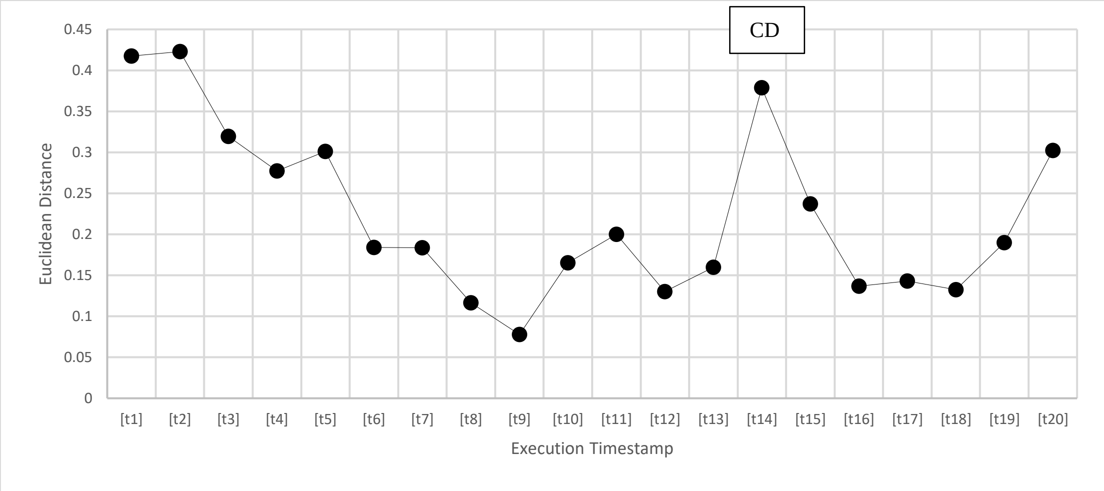



Figure 5.8 Abrupt concept change detection (CD)

To evaluate the accurate detection of concept change, data distribution over three days were analysed. Concept change has occurred due to a remarkable reduction in the number of vehicles on that road compared to the generally high number of vehicles. The concept change has continued, as the number of vehicles was quite less in the next day. This could be due to an event such as a holiday.

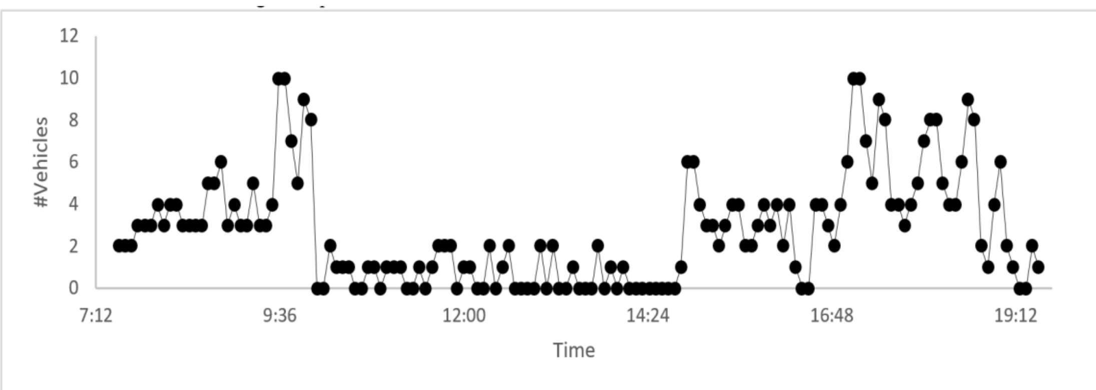



Figure 5.9 Data distribution of detected abrupt concept change

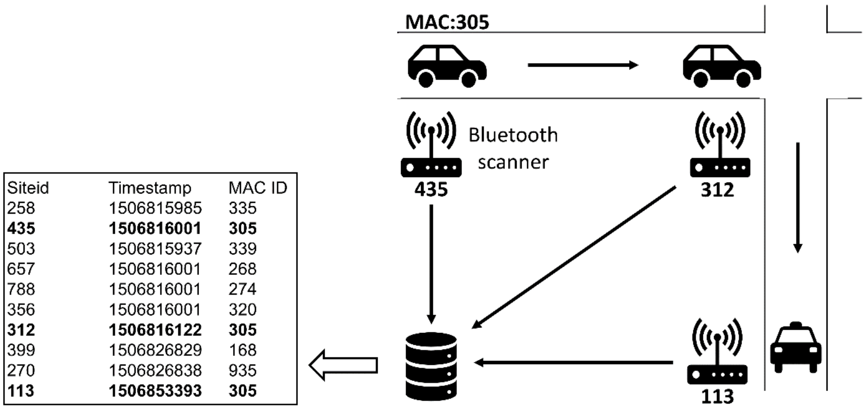
## 5.5 Case Study: Detecting Changes in Motor Traffic in the Arterial Road Network of Victoria, Australia

Road traffic conditions and flow management continue to be an important area of research with many practical implications. During the last decade, the technological landscape of transportation has gradually integrated disruptive technology paradigms into current transportation management systems, leading to Intelligent Transportation Systems (ITS) (Lana et al., 2018; Vlahogianni et al., 2014). The emergence of the Internet of Things (IoT) and sensor networks have surpassed traditional means of collecting data by creating voluminous and continuous streams of real-time data. Leveraging such big data environments is a formidable issue, due to the intense volume and velocity at which data is generated by transportation and mobility systems (Lana et al., 2018). Furthermore, the dynamic nature of these environments makes the data generation volatile, which impedes the effectiveness of decision-making in ITS.

The dynamicity of data generated by transportation systems consists of continuously changing patterns and concept changes. In a traffic context, concept changes are the changes to data distributions in a traffic data stream over time (Gama et al., 2014). These changes are further classified as abrupt and reoccurring concept changes based on the nature of fluctuations in data streams. For example, traffic congestion changes due to peak/ off-peak traffic is a reoccurring concept change, whereas the change in traffic congestion due to an accident or breakdown is an abrupt concept change. Special importance should be placed into identifying abrupt concept changes as it could have an unexpected influence on the entire road network.

This section experimentally evaluates the proposed algorithm using real traffic data from the arterial road network of the State of Victoria, Australia. This experiment has appeared in (Nallaperuma et al., 2019). The traffic information has been acquired from the Bluetooth traffic monitoring system (BTMS) used to monitor the road traffic of arterial roads in Victoria. BTMS is a type of automatic vehicle detector used to estimate travel times in a road network (Bhaskar & Chung, 2013; Mori et al., 2015).

As shown in Figure 5.10, BTMS consists of a network of Bluetooth traffic scanners placed in the junctions of arterial roads. These Bluetooth scanners capture the Bluetooth devices that transit the scanning zone, which is either Bluetooth enabled vehicle stereo systems or the mobile devices of the occupants. The scanners capture the unique electronic identifier (MAC address) of the Bluetooth devices that transit the scanning area and the transit timestamp (Bhaskar & Chung, 2013). Each Bluetooth scanner periodically transmits those records to a central database. Since the electronic identifier is unique to each Bluetooth devices, its travel path can be traced across the network of Bluetooth scanners placed in the road network.




| Siteid | Timestamp | MAC ID |
|---|---|---|
| 258 | 1506815985 | 335 |
| **435** | **1506816001** | **305** |
| 503 | 1506815937 | 339 |
| 657 | 1506816001 | 268 |
| 788 | 1506816001 | 274 |
| 356 | 1506816001 | 320 |
| **312** | **1506816122** | **305** |
| 399 | 1506826829 | 168 |
| 270 | 1506826838 | 935 |
| **113** | **1506853393** | **305** |

Figure 5.10 Bluetooth traffic monitoring system (BTMS)

For this study, the dataset was obtained from Victoria road authority (VicRoads) and comprised all vehicle records for October 2017. This dataset consists of approximately 190 million vehicle records obtained from 1,408 Bluetooth scanners placed at the junctions of arterial roads. It contains records from 545,851 unique MAC-IDs, which is assumed to be unique vehicles.

The capability of the proposed platform is demonstrated by analysing the traffic behaviour around a key Shopping Centre (SC), which is often a volatile traffic region in Victoria. It is one of the largest stand-alone shopping centres in Australia, with over 20 million annual visitor turnarounds. Thus, it accounts for a large traffic footprint in surrounding arterial roads. In addition, it is sandwiched between two large freeways (Princess Hwy and Monash Fwy) which are the key freeways that connect Melbourne Metropolitan Area to Southeast Victorian suburbs.

The combination of the abovementioned factors yields unique and highly volatile traffic patterns in the selected region. Moreover, the surrounding arterial roads are often operated near saturation; thus, any non-recurrent incident can result in significant congestion and its impact often propagated significantly across the arterial road network. Figure 5.11 presents a schematic of the selected SC and the surrounding arterial roads.

### 5.5.1 Traffic Flow Modelling

The raw traffic data are pre-processed, transformed and integrated into a computational format that can be effectively ingested by the proposed algorithm.

The traffic flow $T(t,\ s)$ at location $s$ and time $t$ is a directional measure determined separately for each direction of traffic at $s$, based on the number of vehicles that pass through location $s$ towards a particular direction at time interval $t$ to $t + \Delta t$ (Lighthill & Whitham, 1955). In legacy approaches, traffic flow is measured from several reference points and then extrapolated to determine the traffic flow of the road network based on the density flow theories of hydrodynamics (Aw et al., 2002; Daganzo, 1994; Lighthill & Whitham, 1955; Richards, 1956). However, as the arrival of more comprehensive traffic data collections systems, fully data-driven methods have been recently proposed to estimate the traffic flow (Y. Lv, Duan, Kang, Li, & Wang, 2015; Michau et al., 2017; Yu, Li, Shahabi, Demiryurek, & Liu, 2017).

In addition to point-based traffic flow, the availability of vehicle trajectory data enables the traffic flow to be estimated for road segments (between any two sensor locations). For example, the traffic flow of road segment AB can be determined by $T(t,\ A \rightarrow B)$ and $T(t,\ B \rightarrow A)$, which denotes directional traffic flow at time $t$ from $A$ to $B$ and $B$ to $A$ respectively.

In this dataset, each record $D = \{(v,\ s,\ t)\}$ can be denoted by $(v,\ s,\ t)$ where $v$ is the vehicle denoted by the MAC-ID, $s$ is the location denoted by the site id, and $t$ is the time denoted by the timestamp. The traffic flow of road segments was derived from these traffic records.

Based on the above definition, the traffic flow $T(t,\ A \to B)$ of the road segment, $AB$ can be defined as the number of vehicles that are first detected in $A$ at time $t$ and subsequently detected in $B$, which can be denoted as,

$$T(t, A \to B) = \int_t^{t+\Delta t} \sum_{\forall v} I(v, A \to B, t)dt \tag{5.1}$$

where $\Delta t$ is the sampling interval for the traffic flow, which can be adjusted to obtain the required granularity of the traffic flow. $I(v, A \to B, t)$ is an indicator function which is active if the vehicle $v$ is first detected at $A$ at time $t$ and subsequently detected at $B$ within a time threshold $\tau$. It can be defined as,

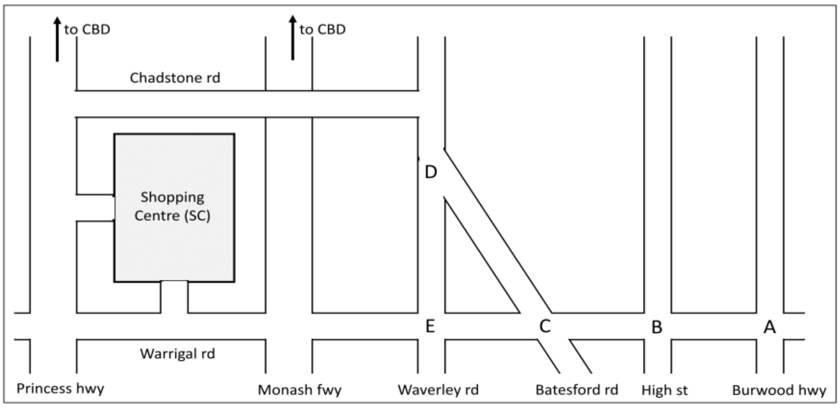



Figure 5.11 Schematic of the road network in the area of interest selected for traffic analysis.

$$I(v, A \to B, t) = \begin{cases} 1 & if\, \exists(v, A, t) \in D\ and\ \exists(v, B, t') \in D, where\ t < t' \leq t + \tau \\ 0 & otherwise \end{cases} \tag{5.2}$$

where $\tau$ is the trip threshold which is set enough for a single trip in the road segment. The idea of setting this threshold is to filter out noisy trips (Nantes et al., 2014) such as pedestrians as well as vehicles that make a stop inside the segment.

Transformed data were presented in the form of a data stream to the proposed algorithm. Figure 5.12 illustrates the concept changes detected in the selected traffic area Figure 5.12 demonstrated the recurrent and non-recurrent concept changes identified by the algorithm.

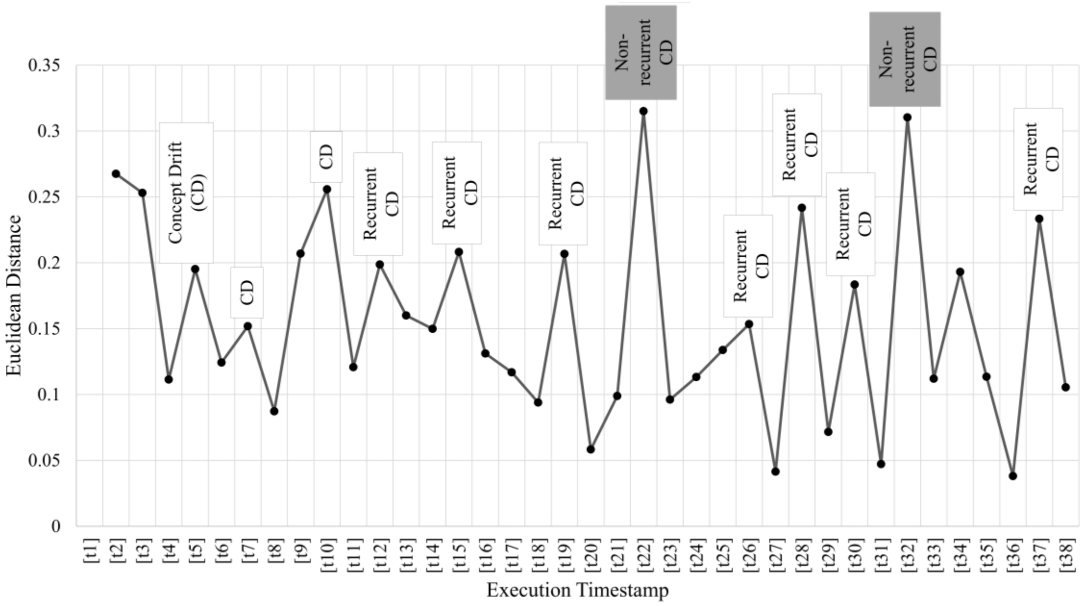



Figure 5.12 Recurrent and non-recurrent concept change detection

The algorithm identified three recurrent concept changes throughout the data stream. These recurrent concept changes relate to the traffic flow changes due to longer shopping hours, weekdays to weekend traffic flow and vice versa. A summary of the recurrent traffic flow changes is denoted in

Table 5-5.

Moreover, the algorithm identified two non-recurrent concept changes at execution timestamps [t22] and [t32]. These incidents occurred at the Warrigal Rd-Waverly Rd intersection (see Figure 5.12). The tweets relevant to this incident were collected using the technique delineated in (Nallaperuma et al., 2019). The collected tweets found that these non-recurrent traffic flow changes had occurred due to an accident (**Error! Reference source not found.**). Tweets in **Error! Reference source not found.** (right) show a communication gap of more than 45 minutes to gather information on the situation. Due to the algorithm's data-driven nature, concept changes on traffic flow can be detected almost real-time. Providing a real-time notification on non-recurrent traffic flow changes will allow better communications and effective optimizations.

Table 5-5 Summary of recurrent concept drift detection

| **Traffic flow change** | **Execution Timestamp** | **Explanation** |
|---|---|---|
| Wednesday → Thursday | [t5]: 4-5/10/2017<br>[t12]: 11-12/10/2017<br>[t19]: 18-19/10/2017<br>[t30]: 25-26/10/2017 | Traffic is affected by the longer shopping hours on Thursday. |
| Friday → Saturday | [t7], [t26], [t34] | The area is a central suburb where most of the traffic would cross while travelling from outer suburbs to city. Traffic will reduce on Saturday compared to Friday, as Saturday is a holiday. |
| Sunday → Monday | [t10], [t15], [t28], [t37] | The area is a central suburb where most of the traffic would cross while travelling from outer suburbs to city. Traffic will increase on Monday compared to Sunday as most people work in the city. |

## 5.6 Case Study: Detecting Change in Driving Behaviours of Autonomous and Human Driven Vehicles

The reality of mixed road traffic, composed of vehicles aided or controlled by autonomous systems and vehicles operated by human drivers, is fast approaching (Katrakazas et al., 2015). In this context, artificial intelligence algorithms in autonomous vehicles face advanced challenges as driving requires not only accurate perception and cognition of information pertaining to vehicle performance and traffic but also coordination and communication with human drivers. Human driving behaviour recognition is necessary to improve communication and coordination between autonomous vehicles and human drivers to increase the overall efficiency of traffic flow, hazard detection and collision avoidance (W. Wang et al., 2014). Current driving and collision avoidance systems are developed exclusively based on average driver performance without any consideration for unique individual behaviour in each situation (W. Wang et al., 2014).

General human behaviour recognition using biological data has been extensively researched in numerous studies on human physiology and psychology (Candamo et al., 2010; E. Kim et al., 2010; Maurer et al., 2006; Shinar, 2017; Yin et al., 2008). Similarly, driver behaviour recognition and change detection can be determined using endogenous factors such as experience, training, human personality and exogenous factors such as vehicle status, speed, road conditions and trajectory, environment, other road users (Shinar, 2017). These endogenous and exogenous factors representing driver behaviour are unique to the situation and the human driver. Some behaviour changes are abrupt, occurs due to a sudden change of environment, such as road hazards. In contrast, other behaviour changes are routine actions such as changing freeway and stopping for traffic lights. Driver Demands and Capabilities Model proposed by Fuller (2005) demonstrates the influence of exogenous and endogenous factors towards the control or loss of control of a motor vehicle. The model explains how the loss of control would lead to a collision or result in an escape by luck or by the complementary actions been taken by other drivers.

Given its dynamic and complex nature, it is pertinent to explore an artificial intelligence solution to address the problem of behaviour change recognition. We propose an extension to the Driver Demands and Capabilities Model, which is implemented as a Self-Structuring AI algorithm for human driver behaviour change detection, abrupt and repeating changes.

A majority of related work in autonomous vehicle navigation and communication focus on in-built intelligent algorithms to investigate and understand the surrounding environment. According to the environment, the autonomous car would transform its configurations such as vehicle position, orientation, linear or angular velocities etc. (Howard, 2009). Some of these in-built algorithms focus on searching the continuous coordinates using road boundaries and the position of the obstacles (Hardy & Campbell, 2010; Jeon et al., 2013; Wille & Form, 2008). Implementation of continuous coordinates is driving corridors which guide the vehicle in a collision-free space avoiding obstacles (Fletcher et al., 2008). Voronoi diagrams (Takahashi & Schilling, 1989) are another method used in inbuilt algorithms, which generates paths to maximize the distance between the vehicle surrounding and obstacles. Path planning of an autonomous vehicle travelling in a parking lot has been programmed using Voronoi diagrams and an obstacle avoidance system modified from a mobile robot system (Dolgov et al., 2010). Both Costmaps (Broggi et al., 2012; Murphy & Newman, 2011) and Occupancy grids (W. Xu et al., 2014) use the probability of a cell in the grid to be associated with an obstacle, road boundary or lane. The grid-based approach is more efficient and uses low computational power, but demonstrates shortcomings in dynamic environments.

Although the above techniques generate an efficient and safe trajectory for autonomous driving, they are inadequate to handle mixed traffic that consists of human-driven vehicles and autonomous vehicles. In mixed traffic scenarios, autonomous vehicles require a higher level of intelligence to understand environmental dynamics and coordinate effectively with other vehicles to ensure traffic flow optimization, collision avoidance, and hazard detection. Fuller (2005) has proposed Driver Demand and Capabilities Model (Figure 5.13) that represents the dynamic interface between the capability of the driver and demands of the driving task.

Capabilities of the driver refer to the endogenous factors such as experience, training, and personality. Task demands refer to exogenous factors such as vehicle status, speed, road conditions and trajectory, environment, other road users. This model can be extended to incorporate driver behaviour change detection using Self-Structuring AI.

The proposed extension to the Driver Demands and Capabilities Model (Fuller, 2005) successfully address diverse driver behaviour using driver behaviour change detection (2). This extension has been proposed in (Nallaperuma et al., 2018). As shown in Figure 5.14, the task is easily completed when capabilities (C) are higher than demand (D). When the demand becomes higher than capabilities, there is an inclination towards abrupt behaviour change. When the behaviour changes and tasks become more difficult than the capabilities, the driver fails at the task and loses control of the vehicles. In many instances, the increased demands are such that the driver is simply unable to maintain the desired trajectory, avoid an obstacle or stop

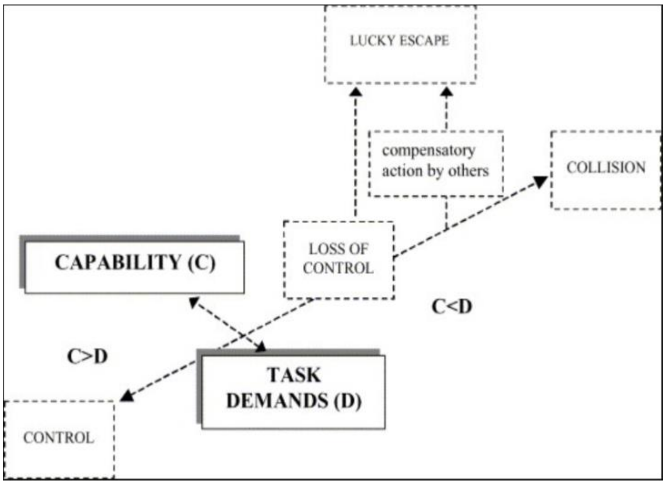



Figure 5.13 Fuller's Driver Demands and Capabilities Model (Fuller, 2005)

in time. Sometimes the actions of another user can be complementary and avoid the situation. Humans have situational awareness and take impulsive actions, resulting in a lucky escape from uncontrollable situations. Capabilities (C) and task demands (D) are unique to each situation, thereby necessitating the need for Self-Structuring AI for personalization and real-time detection of the behaviour change.

This extended Driver Demands and Capabilities model is implemented as Self-Structuring AI algorithm for concept change detection proposed in Chapter 4. Further, the proposed model was experimented using DDD17 (Binas et al., 2017), the first openly available dataset of annotated DAVIS driving recordings accompanying driving data. The dataset comprises 12 hours of driving data recording vehicle speed, GPS position, driver steering, throttle, and brake captured from the vehicle's onboard diagnostics interface. The data were collected while driving on

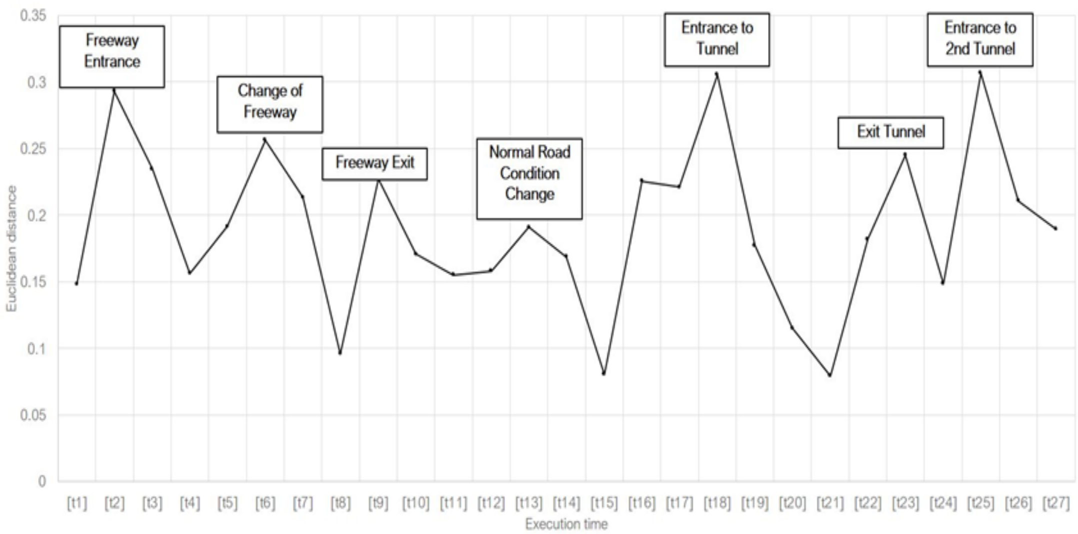



Figure 5.15 Intelligent Detection of Driver Behaviour Change

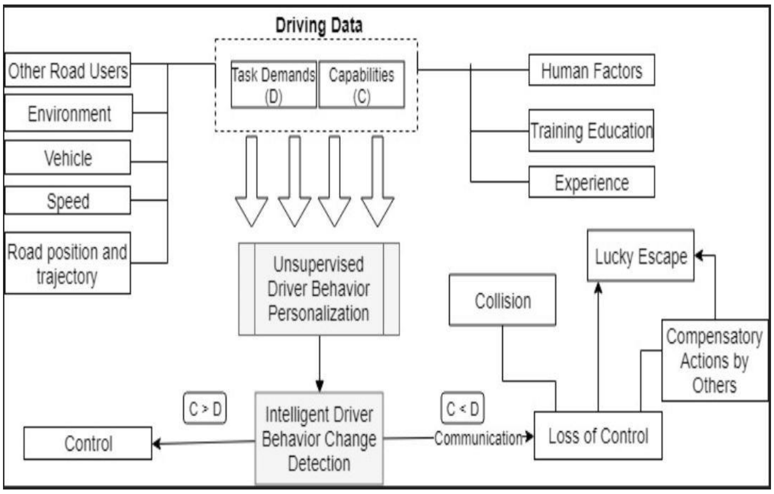



Figure 5.14 Extended Driver Capacity and Driving Demands Model for behavior recognition and change detection

highway and city in the daytime, evening, night, dry and wet weather conditions. The driving data from the onboard diagnostic interface generates a continuous stream of data.

The vehicle is being driven on a highway, and during the period of experimentation of 2720 seconds, several behavioural changes occur. The video feed was independently analysed to identify behaviour changes to be used as gold standard data to evaluate the algorithm. The gold standard data include eleven behaviour changes, including two repeating behaviour changes. Out of these eleven behaviour changes, the proposed algorithm identified five abrupt behaviour changes and one set of repeating behaviour changes (Figure 5.15). Table 5-6 presents the events that caused behavioural changes, time of the event and detection by the algorithm.

Table 5-6 Intelligent Detection of Driver Behaviour Change

| | **Event** | **Time** | **Description** | **Detection by algorithm** | **Abrupt/ Reoccurring** |
|---|---|---|---|---|---|
| **E1** | Lane closure | 190s | Sudden reduction of speed due to a road closure | No | Abrupt |
| **E2** | A sudden change of lanes | 670s | The vehicle moves to another lane to avoid a crash | Yes | Abrupt |
| **E3** | Uneven Road | 950s | Vehicle vibrates due to the uneven road segment | No | Abrupt |
| **E4** | Another car changing lanes | 1540s | A car entered the lane which affects the speed | No | Abrupt |
| **E5** | Change of highway | 2020s | Vehicle enters a highway with different restrictions | Yes | Abrupt |
| **E6** | Highway exit | 2375s | Vehicle exit the highway to a normal road | Yes | Abrupt |

| | | | | | |
|---|---|---|---|---|---|
| **E7** | Traffic congestion | 2410s | Built-up area | Yes | Abrupt |
| **E8** | Tunnel entrance | 2535s | Vehicle enters a tunnel | Yes | Reoccurring |
| **E9** | Tunnel exit | 2550s | Vehicle exit the tunnel | Yes | Reoccurring |
| **E10** | Tunnel entrance | 2630s | Vehicle enters a second tunnel | Yes | Reoccurring |
| **E11** | Tunnel exit | 2680s | Vehicle exit the tunnel to a normal road | No | Reoccurring |

The events comprise of different task demands and driver's capabilities (Table 5-6). Events E1, E3 and E7 occurred due to environmental factors. Events E2 and E4 have occurred due to the influence of another vehicle. Events E5 and E6 have occurred due to change of speed. Events E8-E11 have occurred due to road position and trajectory.

Given that the dataset is unlabelled, an effective means of evaluation of behaviour change detection is to reference the time stamp of behaviour change detection to the corresponding video stream. Due to space limitation, only one event is explained. This event is illustrated in Figure 5.16, a brief narrative as follows. "E2": The vehicle is travelling on a highway when a truck enters the highway via the ramp at time 668s. After 7 seconds, the driver suspects the truck will merge into the same lane, and he/she reacts by attempting to change lanes, as shown in Figure 5.16. This sudden change in driving behaviour would have affected the other drivers if they did not comprehend the situation, which could be the case for autonomous vehicles. To further explain, the vehicle immediately returns to the first lane at 684s as there is a vehicle already on that lane, which could have led to a collision. This sudden change of behaviour has occurred within about 16-second short time frame. Detection of this behaviour change in almost real-time successfully demonstrates the capabilities of the proposed algorithm.

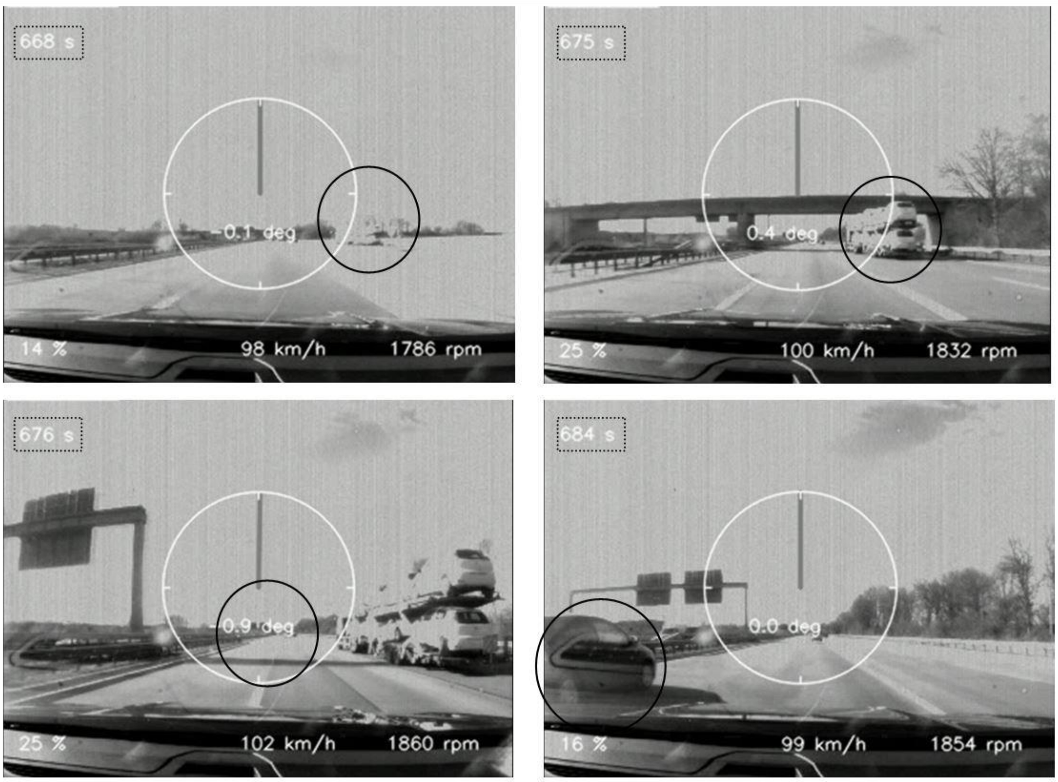



Figure 5.16 Explanation of the change of behaviour detected by the algorithm: Top left – truck entering the highway, Top right – the truck is close to entering the lane of the car, Bottom left – a sudden change of lane by the human driver, Bottom right – car returns to original lane to avoid likely collision

The accuracy of abrupt and repeating behaviour change detection was measured using precision, recall, and F1 as shown in Table 5- 7.

Table 5- 7 Precision, Recall and F1 of behaviour change detection

| | Precision | Recall | F1 |
|---|---|---|---|
| Abrupt behaviour change detection | 100% | 57.14% | 72.72% |
| Repeating behaviour change detection | 100% | 75% | 86% |
| Behaviour change Detection | 100% | 63.63% | 77.77% |

High precision is generally at the expense of recall. However, recall still lies at 57.14%, 75% and 63.63% in abrupt, repeating and overall behaviour change detections respectively. Using the gold standard evaluation, we have determined the following limitations for failing to detect the abrupt behaviour changes in E1, E3, and E4.

E1: Event occurred right at the start of the learning process.

E3: Event was not captured in any of the driving data.

E4: Speed change was insignificant.

## 5.7 Chapter Summary

The chapter presented the empirical evaluation of the proposed Self-Structuring AI algorithm for change detection. The algorithm was applied to six different real-world data streams representatives of various HCDE settings.

- An air traffic dataset comprises 116 million records of flight arrival and departure details of all commercial flight details within the USA.
- A smart electricity meter dataset on the power consumption of a household with a one-minute sampling rate, containing around 2 million records for a duration of 47 months.
- A physical activity monitoring dataset consisting of data from a heart rate monitor and three inertial measurement units (IMUs) worn in hand, chest and ankle. The data were collected from 9 subjects while performing 18 different physical activities.
- A smart city traffic dataset generated by sensors placed in the City of Aarhus. The dataset contains more than 23 million unlabelled IoT data, recorded every 5 minutes from 449 observation points over a period of 6 months.
- A case study carried out with a dataset obtained from Victorian road authority, VicRoads. This dataset consists of approximately 190 million vehicle records obtained from 1,408 Bluetooth scanners placed at the junctions of arterial roads.
- A case study carried out with the DAVIS dataset. The dataset comprises of 12 hours of driving data recording vehicle speed, GPS position, driver steering, throttle, and brake captured from the vehicle's onboard diagnostics interface.

# Chapter 6

# A Self-Structuring AI Algorithm for Concept Change Causality

This chapter proposes the Self-Structuring AI algorithm for concept change causality outlined in chapter 3. The proposed algorithm uses the output of the change detection algorithm (chapter 4) and comprises of (1) a new generalised data structure based on suffix trie to explore sequences of behaviours embedded in each data stream at each $\Delta t$, and (2) a behavioural tree-based method to detect the causal relationship between related multiple data streams at each $\Delta T$, where $\Delta T > \Delta t$. The proposed approach is successfully evaluated using two datasets; a publicly available video stream of a volleyball game and PAMAP2 physical activity monitoring dataset. The volleyball game results were evaluated with published annotations of player behaviour, and the results of PAMAP2 dataset were evaluated with corresponding activity labels.

The chapter is organised as follows. Section 6.1 provides an overview of data streams in an HCDE and highlights the importance of understanding multiple data streams and causal relationships. Section 6.2 defines the terms and provide background on sequence analysis and causal relationships in the context of multiple data streams. Section 6.3 proposes two Self-Structuring AI algorithms. Section 6.4 presents an empirical evaluation of the proposed methodology using two real datasets; PAMAP2 physical activity monitoring dataset and annotated video stream of a volleyball game.

# 6.1 Identifying Causal Relationships between Multiple Data Streams

Data streamed in HCDE contain a large volume of useful information. Understanding causality for concept change in these data streams and taking appropriate actions in a timely manner is crucial for many applications such as healthcare, transportation, manufacturing, and sports. As explained in chapter 1, most of these applications in HCDE generate multiple data streams, rather than single, isolated data stream. A set of related data streams at a given moment will have components that reference each other, which together represent a single event (Krempl et al., 2014). Hence, to comprehend such an event, we need to understand the multiple data streams and the relationships among them.

As explained in chapters 2 and 4, one major challenge in working with data streams is their evolving nature, i.e., the concept at a given time, $t$, will evolve to another concept at time, $t + \alpha$ (Lam & Mostafa, 2001). Further, most concepts reoccur over time (Gama et al., 2014). However, real data streams are not necessarily characterised by a single sequence of concepts that reoccur (Han et al., 2011). Instead, data streams behave as a mix of multiple sequences, each having different conditional probabilities attached to them. We define such sequences and their respective probabilities as the *behaviour* of the data stream. The first part of this chapter focuses on understanding these intricate behaviours of data streams.

Further, these behaviours are affected by related data streams. There is no simple or a linear causal relationship between data streams, and the causal assertions depend on temporal constraints between the set of data streams (Jalali & Jain, 2015; W. Liu et al., 2011). The second part of the chapter emphasises on detection of causal relationships over time on a set of data streams.

## 6.2 Related Work in Change Detection Causality

The essence of causality is the generation and determination of one phenomenon by another (Spirkin, 1975). It is imperative to identify the causal relationships across multiple data streams that are generated in an environment. To the author's best knowledge, literature does not have a comprehensive solution that addresses all the challenges mentioned above. Related work on detecting evolving concepts, exploration of sequences of behaviour, and identifying causal relationships are discussed in the sections below.

In related literature, model-based approaches for recognition, tracking, and segmentation of behaviours are widely used (Fleet et al., 2000; Oliver et al., 2002). A model-based approach for behaviour segmentation carried out by Arikan, Forsyth, O'Brien, & O'Brien (2003) uses hand-annotated training data. Recent years have seen a number of algorithms for behaviour sequencing from annotated data. These algorithms (Arikan & Forsyth, 2002; Lee et al., 2002; Yan Li et al., 2002; Pullen & Bregler, 2002; Tanco & Hilton, 2000) can identify in a video of real person, sequences that follow a path, go to a particular position, perform a particular activity at a specific time. However, these approaches can only capture the limited direction of behaviour; for example, there are many ways to follow a specified path. Kovar, Gleicher, & Pighin (2008) have addressed this limitation by using motion graphs which confines their search to subgraphs induced by the desired action. Nevertheless, this method is ill-suited if the desired actions have short temporal span, such as "jumping" or "catching" or if the actions are to be composed: "jump and catch while running". Hence, this work focuses on capturing the behaviour that will denote the sequence and the temporal span of the behaviour.

Availability of large volume of diverse data streams can now be used to make efficient analysis on different applications. A data stream is often available as a part of a set of related data streams. Hence, understanding causal relationships across multiple data streams are essential to maximizing the utility of these data (Jalali & Jain, 2015).

## 6.3 A Self-Structuring AI Algorithm for Concept Change Causality

This section introduces the proposed Self-Structuring AI algorithm for concept change causality that uses a generalised data structure to capture sequences of behaviours and a similarity-based method to detect the causal relationship between data streams.

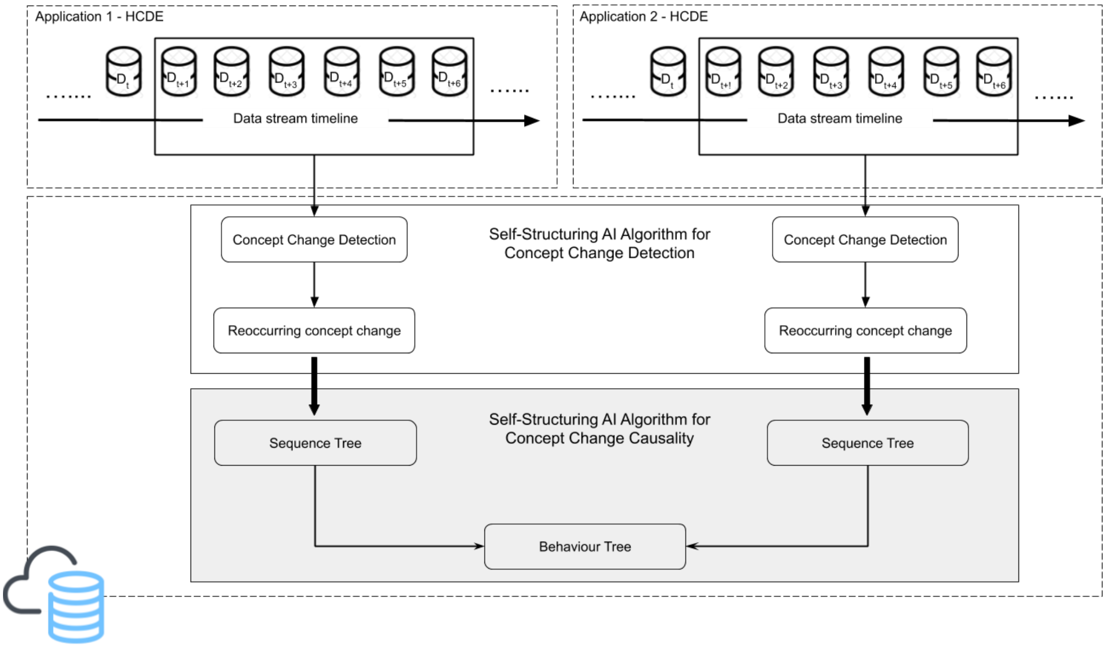



Figure 6.1 Overview of the proposed approach for capturing the causal relationship between related data streams as a sequence of evolving concepts

As shown in Figure 6.1, each data stream of HCDE is processed by a unique instance of the algorithm, in which the first two research challenges are addressed. (1) Concept change detection algorithm: The Self-Structuring AI algorithm described in detail in chapter 4, detects concept changes in any type of data stream. This spatio-temporal technique uses online learning to handle the velocity and volume of the data streams, incremental learning to learn from incoming data representing new concepts, and decremental learning to forget the memories that are no longer relevant. The method will also calculate the usual number of reoccurring concept changes available in the data stream. (2) Generalised suffix trie: A novel generalised algorithm based on suffix trie captures frequent reoccurring concept changes. The algorithm learns online as new concept changes are presented. The resulting generalised suffix trie is used to explore the dynamic behaviours embedded in the data stream represented by continuous behaviours from multidimensional data. (3) Behaviour Tree: An algorithm based on phylogenetic tree theories to identify causal relationships among interrelated data streams. The behaviour algorithm calculates the similarities based on the distance on multiple iterations and executes based on temporality.

Each contributing algorithm is explained in detail in the next sections.

### 6.3.1 Concept Change Detection Algorithm for Detection of Evolving Concepts

Real-time detection of evolving concepts is implemented based on Self-Structuring AI algorithm proposed in chapter 4. As shown in Figure 6. 2, the proposed algorithm consists of one pass online clustering and offline learning. Online clustering addresses the volume and velocity challenges of the data stream and presents aggregated data to offline learning. Offline learning consists of two learning features: (1) incremental learning to learn from new incoming data. With incremental learning, the algorithm can determine the difference between the previously learned concepts and new concepts represented by incoming data. (2) Decremental learning to forget the previous concepts that are no longer useful and relevant. Data-driven triggers define the processing time window and the level of abstraction.

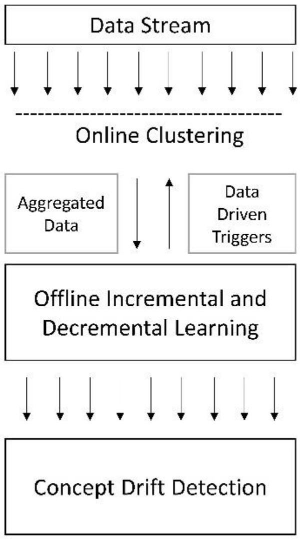



Figure 6. 2 Proposed concept change detection algorithm for detection of evolving concepts using online, incremental and decremental learning

**Initialisation:**

In the initialisation phase, starting cluster centroids for the online clustering algorithm $CFV_{OC}$, number of clusters $n$, initial processing time window $t$ and cluster creation threshold $\emptyset$ is initialised.

**Online Clustering:**

In the first iteration, for each incoming data point in the time window $t$, find the nearest cluster centroid $CFV_{OC}^{q'}$ using Euclidean geometry such that $|v - w_{q'}| \leq |v - w_q| \; \forall q \in N$, where $v$, $w$ are the input and centroid weight vectors, respectively, and $q$ is the index of the cluster centroid. Calculate the new centroid for updated $CFV_{OC}^{q'}$. If $|v - w_{q'}| \geq \emptyset$, a new centroid $CFV_{OC}^{n+1}$ is created to present the data point. In the subsequent iterations, the time window $t$ is the time taken by the offline algorithm.

**Offline Incremental and Decremental Learning:**

The list of $CFV_{OC}$ is the aggregated data that is sent to the offline incremental and decremental learning. Incremental learning is carried out as per the Incremental Knowledge Acquisition and Self Learning (IKASL) algorithm (D. De Silva & Alahakoon, 2010). Inputs to the algorithm are batches of $CFV_{OC}$ received by the online function processed during the time window $t$. Decremental learning is facilitated by forgetting the IKASL learning nodes that are not winners in the subsequent iteration. This denotes that the concept represented by the IKASL learning node has changed or evolved. Using decremental learning allows the algorithm to learn from the current concept and differentiate between the current and old concept. Associations between IKASL learning nodes will be persistent, leading to the creation of memory like structure based on the aggregated outcomes of the learning stages. Adaptation to a new concept is formalised with the incremental and decremental learning.

**Detection of evolving concepts:**

As explained in chapter 4, to detect an evolving concept, the spatial distance between the IKASL learning nodes, $CFV_{offline}$ on consecutive iterations are calculated. If a concept has evolved, there would be a significant distance change $D_t$ at $t$ followed by a reduced distance change $D_{t+1}$ in the following iteration.

**Algorithm 1: Concept Change Detection Algorithm**

1. Calculate the distance measure change ($D_t$) between offline learning nodes $CFV^{t}_{offline}$ and $CFV^{t-1}_{offline}$
2. **if** $D_t > D_{t-1}$ **and** $D_t > D_{t+1}$
   Detect $t$ as an occurrence of the evolution of concept, $\boldsymbol{EC_t}$
   **end if**
3. **if** $Prominence\ (EC_t) > mean\ (EC)$
   Detect $EC_t$ as an occurrence of abrupt concept change
   **else**
   Detect $EC_t$ as an occurrence of reoccurring concept change
   Use IKASL learning nodes of $EC_t$ in the generalised suffix trie
   **end if**

Algorithm 1 is explained in detail in chapter 4. Evolution of concepts are categorised into two; abrupt and reoccurring. This categorisation depends on how often the concepts appear in the data stream. Abrupt concept changes result in relatively higher knowledge acquisition in

IKASL learning as that concept has not been learned by the algorithm before. Therefore, the spatial movement of the IKASL learning nodes between the two learning iterations containing an occurrence of abrupt concept change will be higher. This would be resembled by the prominence of the spatial movement. In contrast, reoccurring concept changes result in lower knowledge acquisition as they have been learned before. As the reoccurring concept change is identified, the IKASL learning nodes are sent to the generalised suffix trie algorithm and will be processed as explained below.

### 6.3.2 Generalised Suffix Trie for Exploration of Sequences of Behaviour

In this section, an adaptation of (Gunasinghe & Alahakoon, 2010, 2013) is introduced for capturing frequent variable-length sequences and their substructures by enhancing the suffix trie data structure. The proposed algorithm is an online algorithm as it can learn and incorporate new concepts as they are presented. As shown in Figure 6.3, the proposed algorithm could capture continuous patterns embedded in multidimensional data. The algorithm is a continuation of the proposed concept change detection algorithm above and consists of three layers; multidimensional IKASL learning nodes (output) from the offline learning layer, discretisation/clustering layer which consists of a pretrained GSOM cluster and pattern arrangement layer which results in the generalised suffix tree.

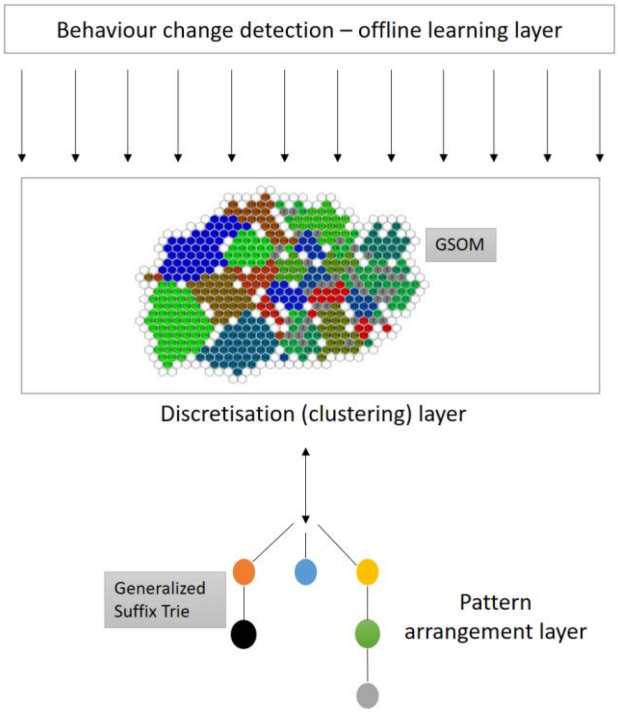



Figure 6.3 Exploration of dynamic behaviours using generalised suffix trie.

The first layer is the output from the offline learning layer of concept change detection algorithm which consist of IKASL learning nodes, $CFV_{offline}$. The algorithm will trigger every time an IKASL learning node is available.

The second layer is the discretisation layer which consists of a GSOM (Alahakoon et al., 2000) map. GSOM has the ability to grow the map dynamically allowing the size of the map to be defined based on the underlying concepts. The GSOM layer is used for labelling the reoccurring concept changes detected by the algorithm 1.

The output of the GSOM is a two-dimensional map of nodes which gives a discretised representation of the input to the algorithm. The nodes are positioned depending on their similarity to each other's weight vectors. To build the generalised suffix trie, the GSOM nodes in the discretisation layer are added as child nodes to the root of the suffix trie. These will then correspond to the initial elements of the sequences captured by the algorithm. Nodes of this layer will have a set of attributes corresponding to the GSOM nodes including a weight vector, positions in the output layer (which is usually a pair of (x, y) coordinates) and the accumulated error value.

The final layer is the sequencing layer in which the generalised suffix trie is constructed. In the generalised suffix trie, the path from the root to each node represents a sequence learned by the algorithm. The generalised suffix trie stabilises as the learning continues and can be used for predictions. Each node in the tree has a weight and a parameter resembling the maturity. The learning rate and threshold in the algorithm define when a node should be considered as a matured node (weight adaptation of the algorithm). Modifying the values of the learning rate and threshold can be used to capture sequences with different characteristics. Both these parameters can take a value between 1 and 0 and 0 for the calculated weight is considered as the most matured. Child nodes can be added only to the mature nodes. The weights of the non-mature nodes are adapted towards the weight of their parent nodes, using a weight adaptation rule based on Hebbian learning until the weight difference between the parent node and child node reaches a predefined threshold. When the weight different reaches the threshold value, the node is marked as mature, and child nodes can be added to it.

The proposed algorithm consists of two phases: initialisation phased and training phase, which is described below.

**The Initialisation Phase:**

During the initialisation phase, the root node is created as a matured node with a weight of 0. As the root node is defined with maximum maturity, children nodes can be added without further learning. Further, the learning parameters; learning rate and threshold are initialised such that the sequence with the required characteristics can be captured by the algorithm. The GSOM is trained with the past data based on (Alahakoon et al., 2000).

**The Training Phase:**

In the training phase, labelled IKASL learning nodes are processed to build an implicit generalised suffix trie which holds frequent subsequence. The algorithm retains the root and a list of mature nodes corresponding to previous elements and the root in memory. The algorithm checks whether the sequence has a child corresponding to each IKASL learning node presented by the concept change detection algorithm. If a child does not exist, a new node is created with a weight of 1 and its maturity property set to false. If a non-mature node exists, the weight is adapted towards the weight of the root. The weight adaptation of a node depends on the learning rate and is used to stop the growth of the tree after a certain depth has been reached. At any given time, the generalised suffix trie that is created represents the frequent sequences present at that time.

**Algorithm 2: Generalised suffix trie algorithm**

---

**I. Initialisation**

```
rootNode ← create a new node
rootNode.maturity ← true
rootNode.weight ← 0
initialise the LEARNING_RATE and THRESHOLD
trainGSOM() for labelling
```

**II. Train generalised suffix trie**

```
for each S ∈ List of sequences
    nodeList ← add root to the list
    for each s ∈ S
        newNodeList ← add root to a temporary list
        for each n ∈ nodeList
            childNode ← find whether the node has a child s
            newNodeList ← add childNode to the temporary list
            if childNode equals to s
                childNode ← update weight of the node
            else
                If childNode = null
                    childNode ← create a childNode s for node
                else if childNode is not mature
                    childNode ← update weight of the node
                end if
            end if
            if childNode is mature
                newNodeList ← add the childNode
            end if
        end for
        nodeList ← newNodeList
```

27. **end for**
28. **end for**
29. Prune non-mature nodes

**III. Weight adaptation**

30. node.weight ← (1-LEARNING_RATE) × nodeCurrentWeight
31. **If** node.weight ≤ THRESHOLD
32. node ← set as a mature node

---

### 6.3.3 Behaviour Tree Method for Identification of Causal Relationships in Multiple Data Streams

To understand the causal relationships between the related data streams, a computational phylogenetic algorithm is proposed in this section. A phylogenetic tree is a diagram showing inferred evolutionary relationships (Meneely et al., 2017). The branching pattern in the phylogenetic tree reflects the process of descent with modification which can be used to denote the causal relationships. Sequences of behaviour with shorter evolutionary distance are expected to be more similar to one another than the ones that are separated over longer evolutionary distance. Further, sequences of behaviour in data streams vary as the concepts evolve; hence the causal relationships between the data streams change. Therefore, the behaviour tree generation needs to be carried out incrementally.

In the proposed algorithm, a behaviour tree is built after each time window, $T$. The algorithm considers the label generated in each iteration by the GSOM algorithm, as explained in algorithm 2: generalised suffix trie. The behavioural labels at time $T$ of multiple data streams are compared together to distinguish the distance between their behaviour and build the behaviour tree. The results of the tree can be used to determine the causal relationship between the data streams at that point. In the next time window $T$, the behaviour tree is modified, which will resemble the new and current causal relationships between the data streams.

**Algorithm 3: Behaviour Tree Algorithm**

L ← List of sequences
**while** size (L) > 1
    **for each** $l_i, l_j \in L, i \neq j$
        $d_{ij}$ ← count number of position differences
    **end for**
    $m$ ← min ($d_{ij}$)
    Draw a phylogenetic tree with $l_i, l_j$ in the $m$ grouping
    $L \leftarrow L.remove(l_i, l_j)$
    $L \leftarrow L.add(avg(l_i, l_j))$
**end while**

# 6.4 Experiments and Results

The proposed approach is evaluated using two publicly available datasets; an annotated video of a volleyball game and PAMAP2 physical activity monitoring dataset.

## 6.4.1 Video Analytics for Sports

Publicly available video streams of volleyball games are used to evaluate the proposed Self Structuring AI algorithm for Concept Change Causality. The output of the concept change detection algorithm is evaluated against the annotations provided by Ibrahim, Muralidharan, Deng, Vahdat, & Mori (2015). This large-scale annotated dataset contains labels for player locations and their corresponding actions. It consists of 55 volleyball games where each player is annotated with one of the nine individual actions resulting in 4830 labelled frames altogether. We would like to point out that the players in the game tend to present more in static behaviours such as standing, waiting as compared to dynamic actions such as blocking, spiking, setting etc. For the demonstration of the proposed approach, 2012 Olympic women's volleyball quarterfinals match between Brazil and Russia and the corresponding annotations are used.

To utilize the video stream in the proposed approach, the video was processed as a raw image sequence and features were extracted for each individual player.

### 6.4.2 Feature Extraction from the Video

It is important to extract features from each individual player for behaviour analysis; otherwise, the features will be overpowered by the features of the volleyball court. Therefore, a boundary box of each player was captured manually in each frame (Figure 6.4 - A) as the region of interest (ROI). A new frame is created with the black background in the ROI (Figure 6.4 - B) for focused features on the player.

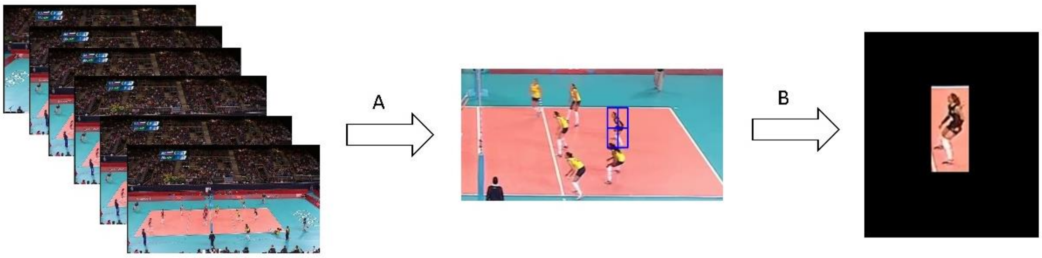



Figure 6.4 Selection of a player for feature extraction: A – capturing the ROI of the player, B – conversion of the black background of the ROI for focused feature extraction.

To extract features and trajectory information, Improved Trajectory (IT) features proposed in (H. Wang & Schmid, 2013) were used. IT is based on the trajectory of local features and have shown impressive performance for many human activity recognition benchmark datasets (Gaur et al., 2011; H. Wang & Schmid, 2013). In this approach, points are densely sampled at several spatial scales. Points located in homogeneous areas are suppressed as it will not be practical to track them reliably. The tracking point in the current frame is achieved by median filtering of a dense optical flow field (Farnebäck, 2003). To avoid drifting, tracking is carried out to 15 frames, followed by new frames to sample them. Also, the approach removes static trajectories before feature extraction as those do not contain motion information. As the next step, the IT approach computes several descriptors such as Histograms of Oriented Gradients (HOG), Histograms of Optical Flows (HOF) and Motion Boundary Histograms (MBH) for each trajectory (Heng Wang et al., 2013). Final trajectory descriptors are a concatenation of normalised vectors of HOG, HOF, and MBH, forming a 204-dimensional feature vector.

### 6.4.3 Evaluation of Proposed Approach

An overview of our approach for identifying causal relationships between data streams as a sequence of evolving concepts is described in Figure 6.5.

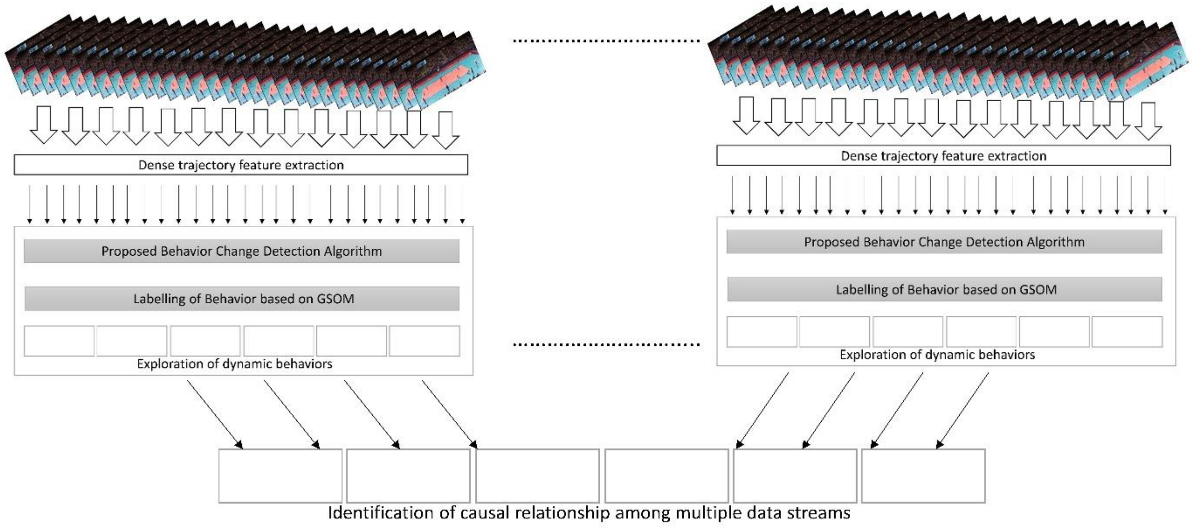



Figure 6.5 Overview of the proposed algorithm applied in the Volleyball dataset

As shown in Figure 6.5, each player is processed as a separate data stream. First, dense trajectory feature extraction is carried out for each individual player. Then the extracted features processed as a data stream are fed into an instance of proposed concept change detection algorithm. After each learning iteration of the concept change detection algorithm, learning nodes in the current layer are labelled using the pretrained GSOM cluster. These labelled nodes are used to explore dynamic behaviours using the proposed suffix trie based generalised data structure. Finally, results from the multiple data streams are used to identify the causal relationship between data streams using the proposed behaviour tree.

### 6.4.4 Detection of Concept Changes

In Figure 6.6, there are two concept changes detected by the algorithm at learning iteration [t5] and [t8]. The corresponding annotation of the player is presented in Table 6-1.

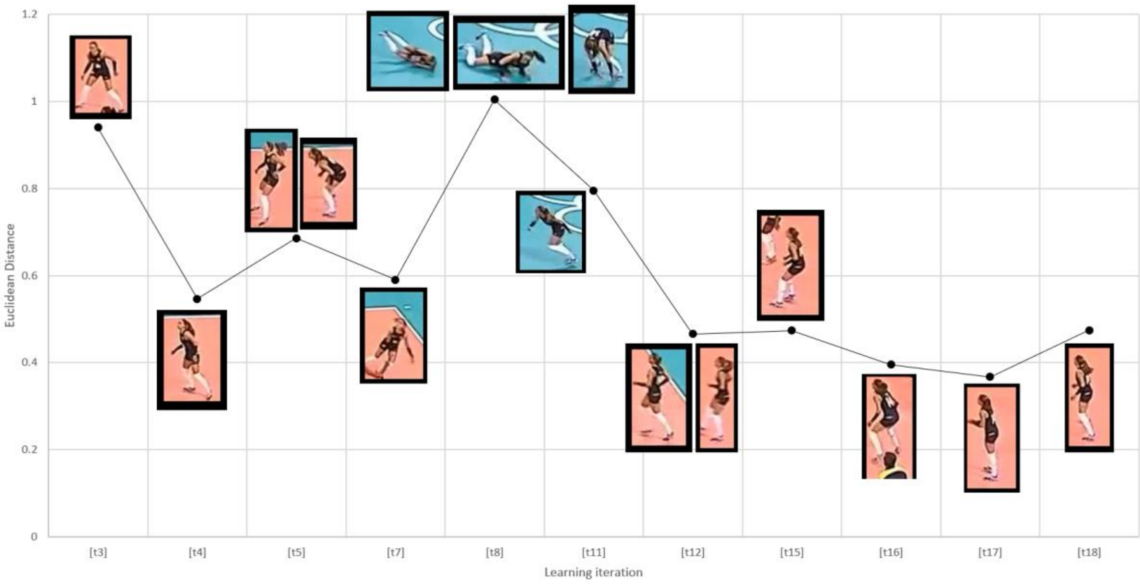



Figure 6.6 Concept change detection in the Volleyball dataset

Table 6-1 Explanation of the concept in each learning iteration

| **Learning Iteration** | **Behaviour activity (as per the annotations from the dataset)** | **Concept Change Detection?** |
|---|---|---|
| **[t3]** | Standing | No |
| **[t4]** | Standing | No |
| **[t5]** | Moving | Yes |
| **[t7]** | Standing | No |
| **[t8]** | Falling | Yes |
| **[t11]** | Standing | No |
| **[t12]** | Standing | No |
| **[t15]** | Standing | No |
| **[t16]** | Standing | No |

| | | |
|---|---|---|
| **[t17]** | Standing | No |
| **[t18]** | Standing | No |
| **[t3]** | Standing | No |

In Figure 6.6, there are three different activities taking place in the video segment. From learning iteration [t3] – [t4] the player is standing, at learning iteration [t5] the player is moving/jumping which has caused a concept change, from learning iteration [t6] - [t7] the player was standing and at the end of [t7] player started to fall which continues to [t8] causing another concept change, from [t12] – [t17] the player continues to stand which is a learned activity by the algorithm.

### Exploration of Sequences of Behaviour using Generalised Suffix Trie

Understanding each player's usual behaviour lets the algorithm understand the learning and use it for causality analysis and implement the predictive capability. Therefore, suffix trie based generalised data structure described in section 6.3.2 provides an overview of the dynamic behaviours that generate concepts. Below sections explain the different sequences of behaviour of each player in the Brazil team. As explained in the dataset, players tend to present more in static behaviours such as standing and waiting than dynamic behaviours. As this could make a biased outcome when determining the frequent sequence of behaviour, sequences starting with static behaviours such as standing is not considered. When the arrangement is frequently reoccurring, the nodes get mature (weight $w$ gets closer to 0).

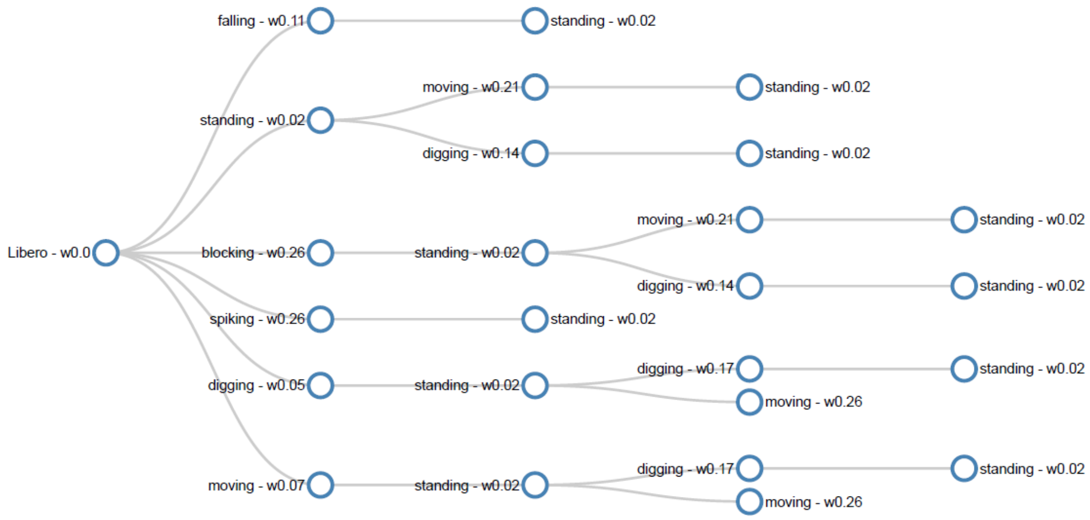



Figure 6.7 Sequence of behaviour generated by Brazilian Player – Libero. $w$ represent the maturity of each activity, $w = 0$ is the most mature.

Libero's most frequent sequence of behaviour (Figure 6.7) is digging ($w$ 0.05) → standing ($w$ 0.02) → digging ($w$ 0.14) → standing ($w$ 0.02) or digging ($w$ 0.05) → standing ($w$ 0.02) → moving ($w$ 0.21). Libero is the player in the second line and is more prompt to be digging the ball and play a defence role.

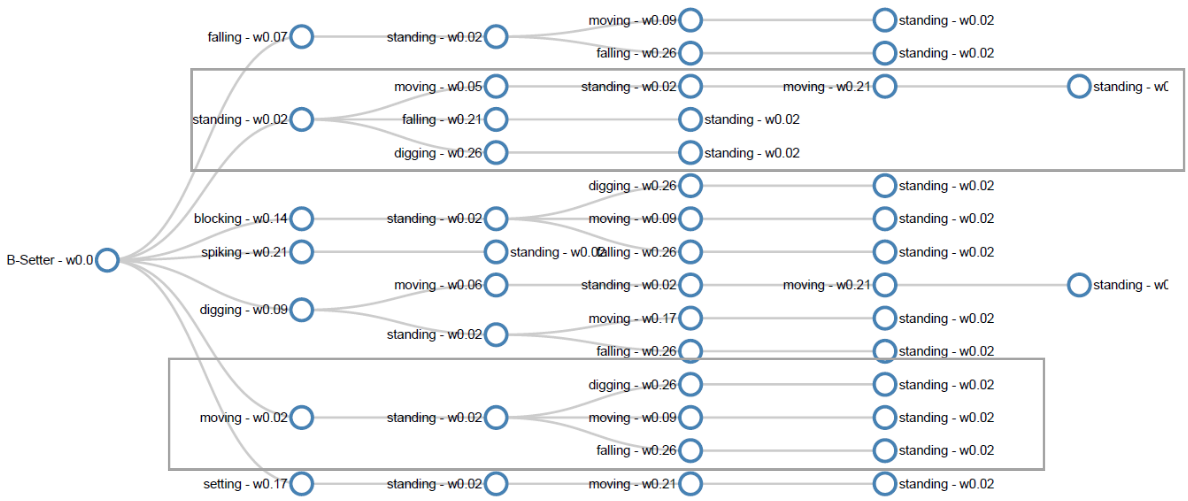


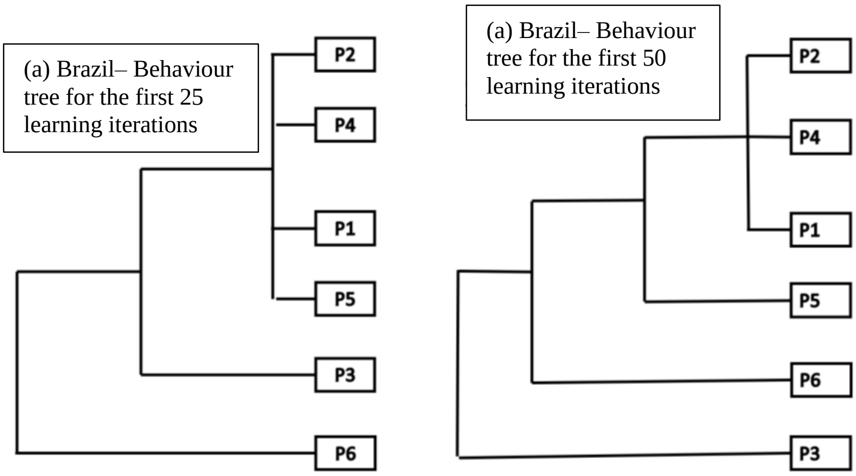

Figure 6.8 Sequence of behaviour generated by Brazilian Player – Setter. $w$ represents the maturity of each activity, $w = 0$ is the most mature.

Setter of the Brazil team (Figure 6.8) plays a more active role where her most matured activity arrangement is moving ($w$ 0.02) → standing ($w$ 0.02) → moving ($w$ 0.09) → standing ($w$ 0.02) or moving ($w$ 0.02) → standing ($w$ 0.02) → digging ($w$ 0.26) → standing ($w$ 0.02) or moving ($w$ 0.02) → standing ($w$ 0.02) → falling ($w$ 0.26) → standing ($w$ 0.02).

### Identification of the Causal Relationship between Data Streams using Behaviour Tree

In a game, the player and neighbouring players handling the ball are more likely to act in synchrony as they are working towards the same local objective. In the next time interval, the outer neighbours will get a concept change when the ball is released. This behaviour will be denoted by data streams and the relationship between them. As explained in Section 0, concepts of Phylogenetic trees are used to identify a similar neighbourhood. Figure 6.9 (a) shows the similar neighbourhood based on the first 25 sequences of behaviours of all the players and Figure 6.9 (b) shows the similar neighbourhood based on the first 50 sequences of behaviours.

As shown in Figure 6.9, during the first 25 behaviours Outside Hitter (P1), Opposite Hitter (P4), Libero (P2) and Middle Blocker (P5) have similar behaviour. In the next 25 behaviours, Middle Blocker (P5) behaves differently to the others in the earlier group. The number of differences in the first and second instance is calculated as follows.




| | P1 | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|---|
| P1 | | 12 | 17 | 14 | 12 | 16 |
| P2 | | | 13 | 8 | 13 | 17 |
| P3 | | | | 15 | 16 | 17 |
| P4 | | | | | 11 | 13 |
| P5 | | | | | | 16 |
| P6 | | | | | | |

| | P1 | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|---|
| P1 | | 28 | 37 | 35 | 30 | 34 |
| P2 | | | 33 | 28 | 31 | 35 |
| P3 | | | | 30 | 35 | 38 |
| P4 | | | | | 31 | 32 |
| P5 | | | | | | 32 |
| P6 | | | | | | |

Figure 6.9 Causal relationship between the Brazil team players

### 6.4.5 PAMAP2 Physical Activity Monitoring Dataset

The proposed algorithm was also evaluated based on the PAMAP2 Physical Activity Monitoring dataset (Reiss & Stricker, 2012). The PAMAP2 dataset consists of data from a heart rate monitor and three inertial measurement units (IMU) worn in hand, chest, and ankle. The IMU measures the body part's specific force, angular rate, the magnetic field surrounding the body using a combination of accelerometers, gyroscopes, and magnetometers. The data were collected from 9 subjects who performed 18 different physical activities such as lying, walking, running, cycling. This multivariate time series dataset consists of more than 3.8 million data records timestamp, 52 attributes of raw sensory data, and the activity label (the ground truth).

Datastream from each subject was processed by a single instance of a proposed algorithm. Unsupervised detection of evolving concepts and exploration of sequences of behaviour is demonstrated using a single subject. Unfortunately, causal relationships between the subjects cannot be demonstrated in this dataset as the subjects perform the activities independently and do not have an impact on causality.

### Detection of Evolving Concepts using Concept Change Detection Algorithm

The results from the concept change detection algorithm are discussed in detailed in section 5.3. In this section, the results are summarised for the purpose of explaining the behaviour. As shown in Figure 6.10, the algorithm detects 13 behaviour changes which correspond to the 13 activity changes (Table 6-2) marked with a ♦ (14 activities).

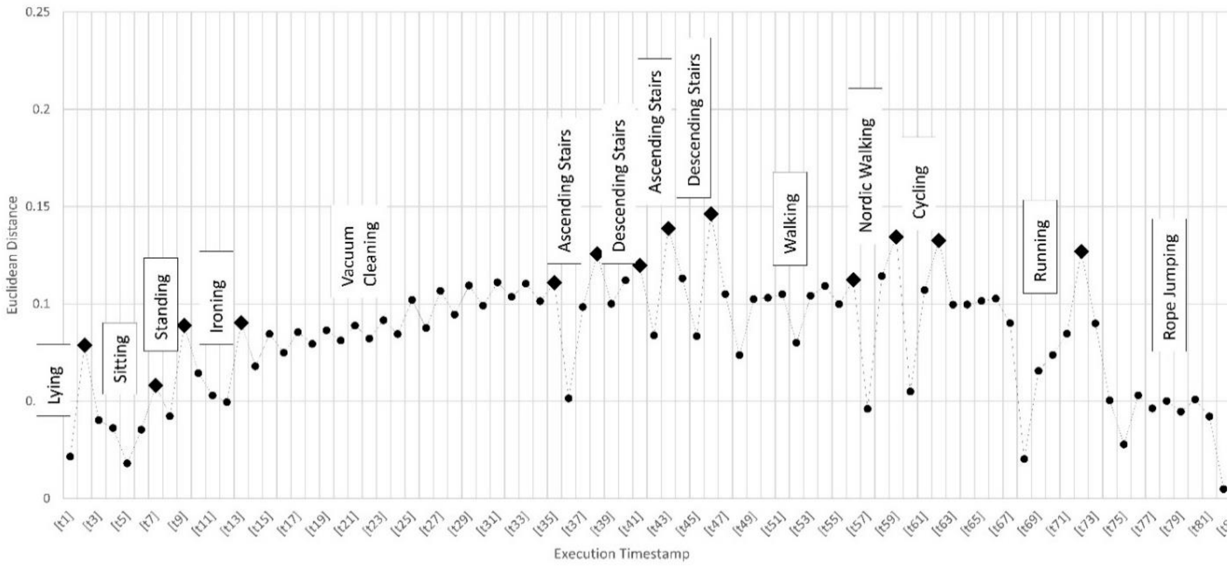



Figure 6.10 Concept change detection in the PAMAP2 Physical Activity Monitoring dataset

Table 6-2 Explanation of captured concept changes

| **Concept Changes** | **Activity Change** |
|---|---|
| CD1 | Lying → Sitting |
| CD2 | Sitting → Standing |
| CD3 | Standing → Ironing |
| CD4 | Ironing → Vacuum Cleaning |
| CD5 | Vacuum Cleaning → Ascending Stairs |
| CD6 | Ascending Stairs → Descending Stairs |
| CD7 | Descending Stairs → Ascending Stairs |
| CD8 | Ascending Stairs → Descending Stairs |
| CD9 | Descending Stairs → Walking |
| CD10 | Walking → Nordic Walking |
| CD11 | Nordic Walking → Cycling |
| CD12 | Cycling → Running |
| CD13 | Running → Rope Jumping |

### Exploration of Sequences of Behaviour using Generalised Suffix Tries

As shown in Figure 6.11, vacuum cleaning is identified as the activity that has been learned by the algorithm longest number of iterations, achieving a high maturity (weight close to 0). Vacuum cleaning has a variation within the activity due to gradual increase of heart rate as the activity is being carried out. In the sequence of behaviour, this is identified by a higher maturity in the nodes; vacuum cleaning – 0.02, walking – 0.07, rope jumping – 0.07, running – 0.07. Also, the algorithm takes less number of iterations to learn the activities that has a small variance such as sitting (w – 0.26), standing (w - 0.41), ironing (w – 0.26). Ascending stairs (w – 0.21) activity has a slightly less variance compared to descending stairs (w – 0.17) due to recordings from IMUs on hand and ankle. Further, cycling and nordic walking achieved a less maturity than deserved as the activities were carried out only a short period of time. Therefore, it is evident that the behaviour of the activities are demonstrated by the behaviour tree.

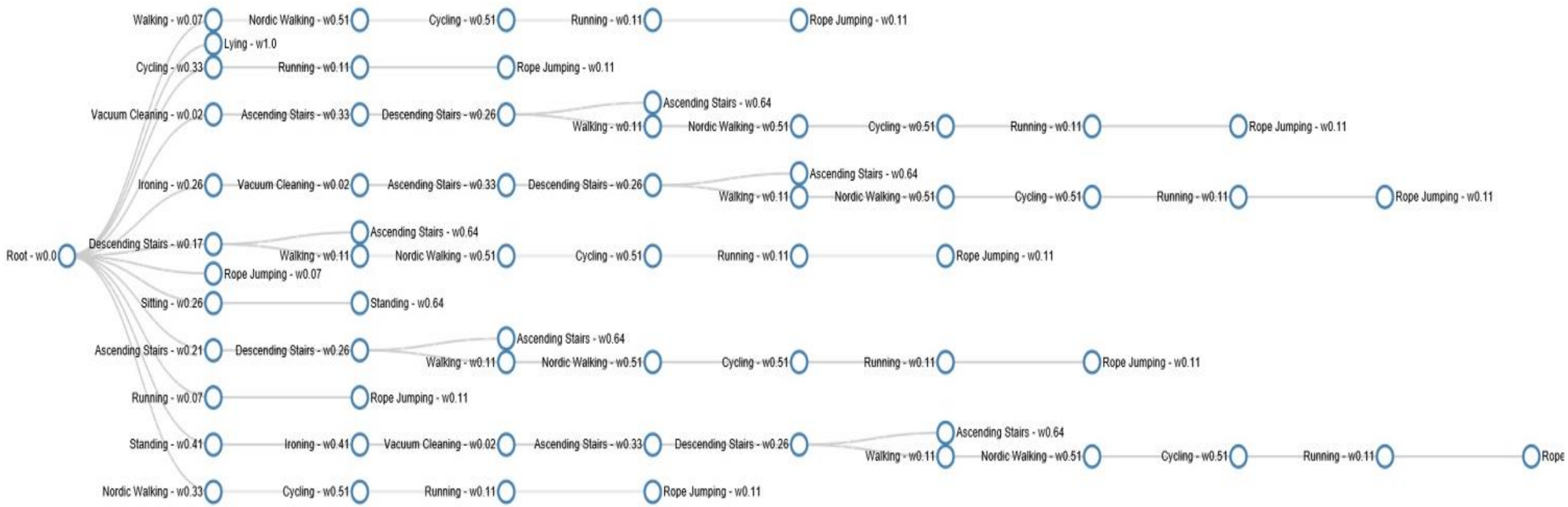



Figure 6.11 Sequence of behaviour in PAMAP2 Physical Activity Monitoring Dataset

## 6.5 Chapter Summary

This chapter proposed two novel Self-Structuring AI algorithms for concept change causality. First, the chapter provided an overview of the proposed algorithm and highlighted the importance of understanding multiple data streams and the causal relationships among them. Next, the chapter provided a background study on sequence analysis and causal relationships and described the proposed algorithms in detail. The first algorithm proposed, based on a generalized suffix tree, is to identify the common sequences of behaviour which facilitate predicting the next behaviour. The second proposed algorithm generates a behavioural tree to identify the causal relationship between multiple data streams in HCDE and outlines an overview of the environment. The chapter presented the empirical evaluations of these algorithms on real-world data streams from HCDE settings, including physical activity monitoring and sports video analytics. In conclusion, the proposed algorithm captures the sequences of behaviours in a data stream. This enables the predictive capability of data stream mining as well as provide evidence on past learnings in causality.

# Chapter 7
# Conclusion

The complexities and plurality in HCDE need a novel artificial intelligence approach as conventional artificial intelligence has not been effective. Therefore, in this thesis, we presented a novel approach motivated by general equilibrium and system dynamics. We introduced and formalised ‘digital equilibrium’ as an extension of the general equilibrium and proposed a conceptual model for detecting concept change and understanding the causality of concept change. The conceptual model was materialised with Self-Structuring AI by designing and developing novel algorithms for concept change detection and concept change causality. These proposed algorithms are facilitated by online, incremental, and decremental learning, hierarchical depiction of influence as well as similarity and gradual causality. This novel approach is empirically evaluated in different HCDE settings using real-world datasets of air traffic, smart motor traffic, smart energy, health and sports analytics.

This chapter concludes this thesis by summarizing the research contributions in section 7.1, addressing the research questions in section 7.2 and finally providing directions for future work in section 7.3.

## 7.1 Summary of Research Contributions

This section provides a summary of each research contribution delineated in chapter 1 of this thesis.

1. A comprehensive literature survey on HCDE, system dynamics, and state of the art machine learning techniques for concept change detection and concept change causality was carried out and presented in chapter 2. We studied the theories of complex environments and system dynamics to characterise an HCDE. The system dynamics theories delineated the challenges that make a system extremely complex and properties that keep a system's stability. We further looked into how a natural system is sensed through events and how the events accumulate into system behaviour. Understanding the system behaviour will give us information related to the underlying system structure. The literature study on machine learning techniques summarized the research problems and challenges in an HCDE. Finally, we carried out a detailed discussion of concept change theories and existing work on change detection in line with data management, forgetting mechanisms, detection, adaption and learning methods.
2. A conceptual model for detecting concept change and causality of change was designed to understand an HCDE and presented in chapter 3. We studied information flows and feedback relationships of natural equilibrium with examples from eco-systems, homeostasis, and macroeconomics. We explored the need for equilibrium in HCDE and theories of equilibrium and complex environment. The proposed conceptual model uses Self-Structuring AI to detect digital representation of natural events and behaviours.
3. Based on the above conceptual model, a novel Self-Structuring AI algorithm was designed and developed to detect concept change in data streams of an HCDE. The proposed algorithm was presented in chapter 4 of this thesis. This Self-Structuring AI algorithm uses three learning paradigms: online learning to handle high volume and velocity of data present in HCDE, incremental learning to provide the ability to learn new concepts, and decremental learning to forget the concepts no longer relevant. The algorithm was built

upon the success of incremental learning of the IKASL algorithm by advancing it to support decremental learning and online learning for continuous detection and adaption to concept change from an unlabelled data stream in HCDE. Key characteristics of the above-proposed algorithm are: 1) unsupervised to learn from unlabelled HCDE, 2) automated time windows as the time of concept change is unpredictable, 3) detection of concept change based on the movement in feature space, and 4) determination of the type of concept change (abrupt or reoccurring) based on the movement of time.

4. Two novel Self-Structuring AI algorithms were designed and developed for understanding concept change causality in HCDE. These algorithms were presented in chapter 6 of the thesis. We explored the data streams in HCDE and highlighted the importance of understanding multiple data streams and the causal relationships among them. As a result, two novel algorithms, a generalized suffix trie and a behaviour tree were proposed. Based on generalized suffix tree, the first algorithm proposed is to identify the common sequences of behaviour which facilitate predicting the next behaviour. The second proposed algorithm generates a behavioural tree to identify the causal relationship between the multiple data streams in HCDE and outlines an overview of the environment.
5. The proposed Self-Structuring AI algorithm for concept change detection was demonstrated in chapter 4 on the SEA dataset, a widely used benchmark dataset for concept change detection. Further demonstrations of the proposed algorithm were carried out on the SEA dataset modified to 12 concepts evaluated against MOA dataset (Bifet et al., 2010).
6. The proposed Self-Structuring AI algorithm for concept change detection was empirically evaluated on six different real-world data streams representatives of various HCDE settings and presented in chapter 5. These include four real-world datasets in air traffic, smart energy, physical activity monitoring, smart city traffic.
7. The proposed Self-Structuring AI algorithm for concept change detection was applied on two real-world case studies and presented in chapter 5. (1) detecting change in motor traffic in the arterial road network of Victoria, Australia (2) detecting change in driving behaviours of autonomous and human-driven vehicles.

8. The proposed Self-Structuring AI algorithm for concept change causality was evaluated on a physical activity monitoring dataset and publicly available video streams of volleyball games and presented in chapter 6.

## 7.2 Addressing the Research Questions

This section describes how the above contributions addressed the research questions delineated in chapter 1. The main research question was composed of four sub-questions.

1. *What factors of natural equilibrium stabilise a natural environment, and how are these factors represented in an HCDE?*

This research question was formulated to investigate the established theories on natural equilibrium applied in complex environments and how these theories can be applied to HCDE. System dynamics, a study of natural and human-made, complex and dynamic systems, was explored in chapter 2 to identify the properties and challenges HCDE.

With this research question, it was identified that any complex environment, including an HCDE, consists of an interconnected set of elements, interconnections and functions coherently organized to achieve a function or purpose. As any of the elements, interconnections or functions can change in these environments, the environments' stability is maintained through resilience, self-organization and hierarchy amid challenges such as non-linear relationships, non-existent boundaries, ubiquitous delays and layers of limit. As explained in chapter 2, changes in a natural or human-made environment are sensed by humans through events, events accumulate into a dynamic pattern of behaviour, behaviour relates information on the underlying system structure and structure is the key to understanding not just what is happening, but why.

This research question was further addressed in chapter 3, creating a conceptual model to introduce and formalize digital equilibrium in HCDE as an extension to general equilibrium. The system structure represented by the natural behaviours and natural events holds the same underlying structure for the digital environment. Therefore, the abstraction of the complexity of an HCDE is denoted by digital representations of the events behaviours. An artificial being or system observes the environment with digital data. The digital equilibrium is materialized with Self-Structuring AI, focusing on concept change detection and understanding the causality of concept change.

2. *How can these factors of equilibrium be used to design an artificial intelligence model capable of detecting changes that lead to disequilibrium and detecting the causality of such change in an HCDE?*

This research question investigates how factors of equilibrium explained above can be used to design an artificial intelligence model to understand the causality for a concept change. The conceptualization of an HCDE includes digital representations of natural events and natural behaviours that hold the system structure. An HCDE consist of challenges such as non-linear relationships, non-existent boundaries, ubiquitous delays and layers of limit, and properties such as self-organization, resilience and hierarchy.

In an HCDE, digital data have high volume, velocity and variety. Chapter 3 justified that artificial intelligence facilitates detection of digital representation of natural events and understanding system structure using the digital representation of natural. With the above-mentioned challenges and properties, chapter 3 proposed a Self-Structuring AI algorithm, where digital representations of natural events are captured through concept change, and digital representations of natural behaviours are captured through sequences and causality.

3. *How can unsupervised machine learning be advanced to develop new algorithms based on the artificial intelligence model designed in Question 2?*

This research question is aimed at designing and developing a novel Self-Structuring AI algorithm. As justified in chapter 3, unsupervised, Self-Structuring AI algorithms are suitable for concept change detection and concept change causality in HCDE settings.

Majority of existing work on concept change detection is based on supervised learning where the data are infrequent, small, isolated, sparse, and labelled. Supervised learning is not feasible in an HCDE where data are connected, dense, unlabelled, and processed in high volume and velocity. Therefore, the proposed concept change detection algorithm developed an unsupervised method for change detection. The proposed algorithm is based on three learning features: 1) incremental learning, 2) decremental learning, and 3) online learning as elaborated in chapter 4. With these learning features, the proposed algorithm facilitates unsupervised, self-adaptive learning in unlabelled data streams to detect concept changes and distinguish between abrupt and reoccurring changes.

Digital representation of natural behaviour reveal information about the system structure that can be predicted using sequences and causality. Perceiving the sequence of events or concepts provides an understanding of the behaviour of an individual component in the environment. Proposed Self-Structuring AI algorithm creates a stabilized sequence tree that learns from reoccurring concept changes in individual data streams and explains ‘why’ a concept change occurred. Causalities describe the non-linear dynamic relationships that exist within the environment. The proposed algorithm creates a behavioural tree that indicates the similarity or distance between different data streams and explains ‘how’ the concept change influences the overall environment. Sequences of dynamic behaviours in HCDE was explored by designing and developing a generalized data structure based on suffix trie. Behavioural tree-based method explored the causal relationships between data streams in HCDE.

4. *How can the algorithms developed in Question 3 be applied to address the practical challenges and complexities of real-world HCDE, demonstrated in use cases of smart cities, smart homes, digital health and sports analytics*?

This research question was formulated to evaluate the Self-Structuring AI algorithms proposed in research question 3 above. These experiments were carried out on a number of real-world HCDE use cases such as air traffic, smart city traffic, smart home, digital health and sports. The datasets include,

- An air traffic dataset comprising of 116 million records of flight arrival and departure details.
- A smart electricity meter dataset on the power consumption of a household with a one-minute sampling rate, containing around 2 million records for a duration of 47 months.
- A physical activity monitoring dataset containing data from a heart rate monitor and three inertial measurement units (IMUs) worn in hand, chest and ankle. The data were collected from 9 subjects while performing 18 different physical activities.
- A smart city traffic dataset consisting of source and destination pairs generated by sensors placed on various road segments in different parts of the City of Aarhus. The dataset contains more than 23 million unlabelled IoT data, recorded every 5 minutes from 449 observation points over a period of 6 months.
- A case study was carried out with a dataset obtained from Victorian road authority, VicRoads, which comprised all vehicle records for October 2017. This dataset consists of approximately 190 million vehicle records obtained from 1,408 Bluetooth scanners placed at the junctions of arterial roads.
- A case study carried out with the DAVIS dataset and annotated driving recordings accompanying driving data. The dataset comprises 12 hours of driving data recording vehicle speed, GPS position, driver steering, throttle, and brake captured from the vehicle's onboard diagnostics interface. The data were collected while driving on highway and city in the daytime, evening, night, dry and wet weather conditions. The driving data from the onboard diagnostic interface generates a continuous stream of data.
- A complete case study on concept change detection and causality was carried out using a video stream of a publicly available volleyball game consisting of 55 games. Each player is annotated with 9 individual actions resulting in 4830 labelled frames.

## 7.3 Future Directions

The Self-Structuring AI algorithms proposed, designed and developed in this thesis have addressed numerous challenges in an HCDE, including volatility and causal relationships in data streams. However, considering the complexities in behaviours and relationships of an HCDE, data stream analytics is still in its infancy. Therefore, we would like to outline several future directions.

The concept change detection algorithm presented in this thesis was developed targeting a processing environment based on a centralized server. However, there is an increased discussion on how the processing can be accommodated in IoT devices themselves. This would facilitate quick and accurate alerts in time-critical IoT applications such as patient monitoring. However, these IoT devices are low resource environments, both in processing power and memory. Hence, a future direction would be to revisit the proposed algorithms to optimize them for low resource environments.

A challenge faced by the online learning layer of the concept change detection algorithm is handling high volumes of data arriving simultaneously. As a future direction, combining scalability of distributed computing to process the high volumes of streaming data with the efficiency of online learning can address this challenge.

Currently, the concept change causality algorithm uses fixed time windows to assess the causality relationship among multiple data streams. A fixed time window might lose the opportunity to provide real-time analysis on causality. This limitation could be further improved to use data-driven, automated time windows similar to how the concept change algorithm operates.

Another category of future improvements is the application of proposed algorithms in settings other than streaming data. For example, the proposed algorithms can be applied on time series datasets and sequential datasets by removing the online learning, which is used to learn from streams of data processed in high velocity.

Another research application of the proposed algorithms would be textual data streams such as Twitter streams. An embedding layer that derives numerical representation of textual content needs to be included to facilitate textual data. Such a setting will be able to detect changes in the discussion topic over time.

In a potential scenario, where the traffic environment consists of fully autonomous vehicles, this environment will be managed by an ensemble of AI algorithms that receive multiple streams of IoT data to manage and control the behaviour of the entire system. The proposed self-structuring AI algorithm for concept change detection will detect the changes in traffic flow that occur due to natural events such as changes in weather or traffic incidents. These natural events would be captured through the proposed algorithm using their digital representations. Detection of these natural events in real-time will optimize the traffic flow and perform efficiently without human intervention.

# References

Aggarwal, C. C. (2007). *Data Streams: Models and Algorithms*. Springer Science & Business Media.

Alahakoon, D., Halgamuge, S., & Srinivasan, B. (2000). Dynamic Self-organizing Maps with Controlled Growth for Knowledge Discovery. *IEEE Transactions on Neural Networks*, *11*(3), 601–614. https://doi.org/10.1109/72.846732

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition using Smartphones. *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 437–442. https://upcommons.upc.edu/handle/2117/20897

Arikan, O., & Forsyth, D. A. (2002). Interactive Motion Generation from Examples. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 483–490. https://doi.org/10.1145/566570.566606

Arikan, O., Forsyth, D. A., O'Brien, J. F., & O'Brien, J. F. (2003). Motion Synthesis from Annotations. *ACM SIGGRAPH 2003 Papers*, 402–408. https://doi.org/10.1145/1201775.882284

Armah, F. A., Yawson, D. O., & Pappoe, A. A. N. M. (2010). A Systems Dynamics Approach to Explore Traffic Congestion and Air Pollution Link in the City of Accra, Ghana. *Sustainability*, *2*(1), 252–265. https://doi.org/10.3390/su2010252

Aw,  a., Klar, A., Rascle, M., & Materne, T. (2002). Derivation of Continuum Traffic Flow Models from Microscopic Follow-the-Leader Models. *SIAM Journal on Applied Mathematics*, *63*(1), 259–278. https://doi.org/10.1137/S0036139900380955

Axelrod, R., & Cohen, M. D. (1999). *Harnessing Complexity: Organizational Implications of a Scientific Frontier*. Free Press.

Bailey, K. D. (1984). Equilibrium, Entropy and Homeostasis: A Multidisciplinary Legacy. *Systems Research*, *1*(1), 25–43. https://doi.org/10.1002/sres.3850010104

Ball, P. (2012). *Why Society is a Complex Matter: Meeting Twenty-first Century Challenges with a New Kind of Science*. Springer.

Bandaragoda, T., De Silva, D., & Alahakoon, D. (2017). Automatic Event Detection in Microblogs using Incremental Machine Learning. *Journal of the Association for Information Science and Technology*, *68*(10), 2394–2411. https://doi.org/10.1002/asi.23896

Barbará, D. (2002). Requirements for Clustering Data Streams. *ACM SIGKDD Explorations Newsletter*, *3*(2), 23–27. https://doi.org/10.1145/507515.507519

Beliakov, G., Pradera, A., & Calvo, T. (2007). *Aggregation Functions: A Guide for Practitioners*. Springer. https://doi.org/10.1007/978-3-540-73721-6

Bhargava, R., Kargupta, H., & Powers, M. (2003). Energy Consumption in Data Analysis for On-board and Distributed Applications. *Proceedings of the ICML'03 Workshop on Machine Learning Technologies for Autonomous Space Applications*.

Bhaskar, A., & Chung, E. (2013). Fundamental Understanding on the Use of Bluetooth Scanner as a Complementary Transport Data. *Transportation Research Part C: Emerging Technologies*, *37*, 42–72. https://doi.org/10.1016/j.trc.2013.09.013

Bifet, A., & Gavaldà, R. (2007). Learning from Time-Changing Data with Adaptive Windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (pp. 443–448). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611972771.42

Bifet, A., & Gavaldà, R. (2009). Adaptive Learning from Evolving Data Streams. In N. M. Adams, C. Robardet, A. Siebes, & J.-F. Boulicaut (Eds.), *Advances in Intelligent Data Analysis VIII* (pp. 249–260). Springer. https://doi.org/10.1007/978-3-642-03915-7_22

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive Online Analysis. *Journal of Machine Learning Research*, *11*, 1601–1604.

Bifet, A., Holmes, G., Pfahringer, B., & Gavaldà, R. (2009). Improving Adaptive Bagging Methods for Evolving Data Streams. In Z.-H. Zhou & T. Washio (Eds.), *Advances in Machine Learning* (pp. 23–37). Springer. https://doi.org/10.1007/978-3-642-05224-8_4

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009). New Ensemble Methods for Evolving Data Streams. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 139–148. https://doi.org/10.1145/1557019.1557041

Binas, J., Neil, D., Liu, S.-C., & Delbruck, T. (2017). DDD17: End-To-End DAVIS Driving Dataset. *ArXiv:1711.01458 [Cs]*. http://arxiv.org/abs/1711.01458

Bouchachia, A. (2011). Fuzzy Classification in Dynamic Environments. *Soft Computing*, *15*(5), 1009–1022. https://doi.org/10.1007/s00500-010-0657-0

Bouchachia, A., & Vanaret, C. (2014). GT2FC: An Online Growing Interval Type-2 Self-Learning Fuzzy Classifier. *IEEE Transactions on Fuzzy Systems*, *22*(4), 999–1018. https://doi.org/10.1109/TFUZZ.2013.2279554

Brain, D., & Webb, G. I. (2002). The Need for Low Bias Algorithms in Classification Learning from Large Data Sets. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *Principles of Data Mining and Knowledge Discovery* (pp. 62–73). Springer.

Broggi, A., Medici, P., Zani, P., Coati, A., & Panciroli, M. (2012). Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. *Annual Reviews in Control, 36*(1), 161–171. https://doi.org/10.1016/j.arcontrol.2012.03.012

Cai, Y. D., Clutter, D., Pape, G., Han, J., Welge, M., & Auvil, L. (2004). MAIDS: Mining Alarming Incidents from Data Streams. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, 919–920. https://doi.org/10.1145/1007568.1007695

Câmpan, A., & Şerban, G. (2006). Adaptive Clustering Algorithms. In L. Lamontagne & M. Marchand (Eds.), *Advances in Artificial Intelligence* (pp. 407–418). Springer. https://doi.org/10.1007/11766247_35

Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., & Kasturi, R. (2010). Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *IEEE Transactions on Intelligent Transportation Systems*, *11*(1), 206–224. https://doi.org/10.1109/TITS.2009.2030963

Carvalho, V. R., & Cohen, W. W. (2006). Single-pass Online Learning: Performance, Voting Schemes and Online Feature Selection. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 548–553. https://doi.org/10.1145/1150402.1150466

Cervero, R., & Golub, A. (2007). Informal Transport: A Global Perspective. *Transport Policy*, *14*(6), 445–457. https://doi.org/10.1016/j.tranpol.2007.04.011

Cortada, J. W. (2020). *Living with Computers*. Springer. https://doi.org/10.1007/978-3-030-34362-0

Daganzo, C. F. (1994). The Cell Transmission Model. Part I: A Simple Dynamic Representation of Highway Traffic. *Transportation Research Part B: Methodological*, *28*(4), 269–287.

Dasu, T., Krishnan, S., Venkatasubramanian, S., & Yi, K. (2006). An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams. *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*.

De Silva, D. (2010). *A Cognitive Approach to Autonomous Incremental Learning* [PhD Thesis]. Monash University.

De Silva, D., & Alahakoon, D. (2010). Incremental Knowledge Acquisition and Self Learning from Text. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2010.5596612

De Silva, D., Sierla, S., Alahakoon, D., Osipov, E., Yu, X., & Vyatkin, V. (2020). Toward Intelligent Industrial Informatics: A Review of Current Developments and Future Directions of Artificial Intelligence in Industrial Applications. *IEEE Industrial Electronics Magazine*, *14*(2), 57–72. https://doi.org/10.1109/MIE.2019.2952165

De Silva, D., Yu, X., Alahakoon, D., & Holmes, G. (2011a). A Data Mining Framework for Electricity Consumption Analysis From Meter Data. *IEEE Transactions on Industrial Informatics*, *7*(3), 399–407. https://doi.org/10.1109/TII.2011.2158844

De Silva, D., Yu, X., Alahakoon, D., & Holmes, G. (2011b). Incremental Pattern Characterization Learning and Forecasting for Electricity Consumption using Smart Meters. *2011 IEEE International Symposium on Industrial Electronics*, 807–812. https://doi.org/10.1109/ISIE.2011.5984262

De Silva, D., Yu, X., Alahakoon, D., & Holmes, G. (2011c). Semi-Supervised Classification of Characterized Patterns for Demand Forecasting using Smart Electricity Meters. *2011 International Conference on Electrical Machines and Systems*, 1–6. https://doi.org/10.1109/ICEMS.2011.6073434

Devi, Y. U., & Rukmini, M. S. S. (2016). Iot In Connected Vehicles: Challenges and Issues — A Review. *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 1864–1867. https://doi.org/10.1109/SCOPES.2016.7955769

Ditzler, G., & Polikar, R. (2011). Hellinger Distance Based Drift Detection for Nonstationary Environments. *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, 41–48. https://doi.org/10.1109/CIDUE.2011.5948491

Dixon, H. D. (2001). *Surfing Economics*. Palgrave.

Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments. *The International Journal of Robotics Research*, *29*(5), 485–501. https://doi.org/10.1177/0278364909359210

Domingos, P., & Hulten, G. (2001). A General Method for Scaling Up Machine Learning Algorithms and Its Application to Clustering. *Proceedings of the Eighteenth International Conference on Machine Learning*, 106–113.

Domingos, P., & Hulten, G. (2000). Mining High-Speed Data Streams. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 71–80. https://doi.org/10.1145/347090.347107

Downs, A. (1999). Some Realities about Sprawl and Urban Decline. *Housing Policy Debate*, *10*(4), 955–974. https://doi.org/10.1080/10511482.1999.9521356

Eco Sapien. (2015). *Eco Sapien—An infographic exploring Yellowstone National Park..* https://ecosapienshow.tumblr.com/post/109408298159/an-infographic-exploring-yellowstone-national-park

Faiz, A., Gautam, S., & Burki, E. (1995). Air Pollution From Motor Vehicles: Issues and Options for Latin American Countries. *Science of The Total Environment*, *169*(1), 303–310. https://doi.org/10.1016/0048-9697(95)04662-K

Farnebäck, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In J. Bigun & T. Gustavsson (Eds.), *Image Analysis* (pp. 363–370). Springer.

Fleet, D. J., Black, M. J., Yacoob, Y., & Jepson, A. D. (2000). Design and Use of Linear Models for Image Motion Analysis. *International Journal of Computer Vision*, *36*(3), 171–193. https://doi.org/10.1023/A:1008156202475

Fletcher, L., Teller, S., Olson, E., Moore, D., Kuwata, Y., How, J., Leonard, J., Miller, I., Campbell, M., Huttenlocher, D., Nathan, A., & Kline, F.-R. (2008). The MIT–Cornell Collision and Why it Happened. *Journal of Field Robotics*, *25*(10), 775–807. https://doi.org/10.1002/rob.20266

Forrester, J. W. (1971). Counterintuitive Behavior of Social Systems. *Simulation*, *16*(2), 61–76. https://doi.org/10.1177/003754977101600202

Fortin, D., Beyer, H. L., Boyce, M. S., Smith, D. W., Duchesne, T., & Mao, J. S. (2005). Wolves Influence Elk Movements: Behavior Shapes a Trophic Cascade in Yellowstone National Park. *Ecology*, *86*(5), 1320–1330. https://doi.org/10.1890/04-0953

French, R. M. (1999). Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, *3*(4), 128–135. https://doi.org/10.1016/S1364-6613(99)01294-2

Frías-Blanco, I., Campo-Ávila, J. del, Ramos-Jiménez, G., Morales-Bueno, R., Ortiz-Díaz, A., & Caballero-Mota, Y. (2015). Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 810–823. https://doi.org/10.1109/TKDE.2014.2345382

Fuller, R. (2005). Towards a General Theory of Driver Behaviour. *Accident Analysis & Prevention*, *37*(3), 461–472. https://doi.org/10.1016/j.aap.2004.11.003

Furao, S., & Hasegawa, O. (2006). An Incremental Network for On-Line Unsupervised Classification and Topology Learning. *Neural Networks: The Official Journal of the International Neural Network Society*, *19*(1), 90–106. https://doi.org/10.1016/j.neunet.2005.04.006

Gaber, M. M., Krishnaswamy, S., & Zaslavsky, A. (2005). On-board Mining of Data Streams in Sensor Networks. In S. Bandyopadhyay, U. Maulik, L. B. Holder, & D. J. Cook (Eds.), *Advanced Methods for Knowledge Discovery from Complex Data* (pp. 307–335). Springer. https://doi.org/10.1007/1-84628-284-5_12

Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining Data Streams: A Review. *SIGMOD Record*, *34*(2), 18–26. https://doi.org/10.1145/1083784.1083789

Gama, J. (2010). *Knowledge Discovery from Data Streams* (1st ed.). Chapman & Hall/CRC.

Gama, J., & Castillo, G. (2006). Learning with Local Drift Detection. In X. Li, O. R. Zaïane, & Z. Li (Eds.), *Advanced Data Mining and Applications* (pp. 42–55). Springer. https://doi.org/10.1007/11811305_4

Gama, J., & Gaber, M. M. (Eds.). (2007). *Learning from Data Streams*. Springer. https://doi.org/10.1007/3-540-73679-4

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with Drift Detection. In A. L. C. Bazzan & S. Labidi (Eds.), *Advances in Artificial Intelligence – SBIA 2004* (pp. 286–295). Springer. https://doi.org/10.1007/978-3-540-28645-5_29

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, *46*(4), 44:1-44:37. https://doi.org/10.1145/2523813

Gaur, U., Zhu, Y., Song, B., & Roy-Chowdhury, A. (2011). A “String of Feature Graphs” Model for Recognition of Complex Activities in Natural Videos. *2011 International Conference on Computer Vision*, 2595–2602. https://doi.org/10.1109/ICCV.2011.6126548

Golab, L., & Özsu, M. T. (2003). Issues in Data Stream Management. *SIGMOD Record*, *32*(2), 5–14. https://doi.org/10.1145/776985.776986

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., & Abdessalem, T. (2017). Adaptive Random Forests for Evolving Data Stream Classification. *Machine Learning*, *106*(9), 1469–1495. https://doi.org/10.1007/s10994-017-5642-8

Gonçalves Jr., P. M., de Carvalho Santos, S. G. T., Barros, R. S. M., & Vieira, D. C. L. (2014). A Comparative Study on Concept Drift Detectors. *Expert Systems with Applications*, *41*(18), 8144–8156. https://doi.org/10.1016/j.eswa.2014.07.019

Green, W. (2019). *ESS Topic 1.3: Energy and Equilibria*. Amazing World of Science with Mr. Green. https://www.mrgscience.com/ess-topic-13-energy-and-equilibria.html

Gunasinghe, U., & Alahakoon, D. (2010). A Biologically Inspired Neural Clustering Model for Capturing Patterns from Incomplete Data. *2010 Fifth International Conference on Information and Automation for Sustainability*, 126–131. https://doi.org/10.1109/ICIAFS.2010.5715647

Gunasinghe, U., & Alahakoon, D. (2013). The Adaptive Suffix Tree: A Space Efficient Sequence Learning Algorithm. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2013.6707052

Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Elsevier.

Hardy, J., & Campbell, M. (2010). Contingency Planning over Probabilistic Hybrid Obstacle Predictions for Autonomous Road Vehicles. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2237–2242. https://doi.org/10.1109/IROS.2010.5652763

Howard, T. (2009). *Adaptive Model-Predictive Motion Planning for Navigation in Complex Environments* [Master's thesis, Carnegie Mellon University]. https://apps.dtic.mil/sti/citations/ADA507004

Ibrahim, M., Muralidharan, S., Deng, Z., Vahdat, A., & Mori, G. (2015). A Hierarchical Deep Temporal Model for Group Activity Recognition. *ArXiv:1511.06040 [Cs]*. http://arxiv.org/abs/1511.06040

Jalali, L., & Jain, R. (2015). Bringing Deep Causality to Multimedia Data Streams. *Proceedings of the 23rd ACM International Conference on Multimedia*, 221–230. https://doi.org/10.1145/2733373.2806278

Jayaratne, M., Alahakoon, D., Silva, D. D., & Yu, X. (2017). Apache Spark based Distributed Self-Organizing Map Algorithm for Sensor Data Analysis. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 8343–8349. https://doi.org/10.1109/IECON.2017.8217465

Jeon, J. hwan, Cowlagi, R. V., Peters, S. C., Karaman, S., Frazzoli, E., Tsiotras, P., & Iagnemma, K. (2013). Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving. *2013 American Control Conference*, 188–193. https://doi.org/10.1109/ACC.2013.6579835

Johnson, N. (2009). *Simply Complexity: A Clear Guide to Complexity Theory*. Oneworld Publications.

Joskow, P. L. (2001). California's Electricity Crisis. *Oxford Review of Economic Policy*, *17*(3), 365–388. https://doi.org/10.1093/oxrep/17.3.365

Kargupta, H., Park, B.-H., Pittie, S., Liu, L., Kushraj, D., & Sarkar, K. (2002). MobiMine: Monitoring the Stock Market from a PDA. *SIGKDD Explorations*, *3*, 37–46. https://doi.org/10.1145/507515.507521

Katrakazas, C., Quddus, M., Chen, W.-H., & Deka, L. (2015). Real-Time Motion Planning Methods for Autonomous On-Road Driving: State-Of-The-Art and Future Research Directions. *Transportation Research Part C: Emerging Technologies*, *60*, 416–442. https://doi.org/10.1016/j.trc.2015.09.011

Kifer, D., Ben-David, S., & Gehrke, J. (2004). Detecting Change in Data Streams. *Proceedings of the 30th VLDB Conference*, 180–192.

Kim, E., Helal, S., & Cook, D. (2010). Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Computing*, *9*(1), 48–53. https://doi.org/10.1109/MPRV.2010.7

Kim, S. K., & Kim, S. (2020). Brain-inspired Method for Hyper-connected and Distributed Intelligence. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 657–660. https://doi.org/10.1109/ICAIIC48513.2020.9065226

Klinkenberg, R. (2004). Learning Drifting Concepts: Example Selection vs. Example Weighting. *Intelligent Data Analysis*, *8*(3), 281–300. https://doi.org/10.3233/IDA-2004-8305

Klinkenberg, R., & Joachims, T. (2000). Detecting Concept Drift with Support Vector Machines. *In Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 487–494.

Klinkenberg, R., & Renz, I. (1998). Adaptive Information Filtering: Learning in the Presence of Concept Drifts. *Learning for Text Categorization*, 33–40.

Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., & Barnaghi, P. (2014). A Knowledge-Based Approach for Real-Time IoT Data Stream Annotation and Processing. *2014 IEEE International Conference on Internet of Things (IThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 215–222. https://doi.org/10.1109/iThings.2014.39

Kolter, J. Zico, & Maloof, M. A. (2007). Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts. *The Journal of Machine Learning Research*, *8*(91), 2755–2790.

Kolter, Jeremy Z., & Maloof, M. A. (2003). Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift. *Proceedings of the Third IEEE International Conference on Data Mining*, 123–130. https://doi.org/10.1109/ICDM.2003.1250911

Kovar, L., Gleicher, M., & Pighin, F. (2008). Motion Graphs. *ACM SIGGRAPH 2008 Classes*, 51:1-51:10. https://doi.org/10.1145/1401132.1401202

Koychev, I. (2000). Gradual Forgetting for Adaptation to Concept Drift. *Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning*, 101–107.

Koychev, I. (2002). Tracking Changing User Interests through Prior-Learning of Context. In P. De Bra, P. Brusilovsky, & R. Conejo (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 223–232). Springer. https://doi.org/10.1007/3-540-47952-X_24

Krempl, G., Žliobaite, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., & Stefanowski, J. (2014). Open Challenges for Data Stream Mining Research. *ACM SIGKDD Explorations Newsletter*, *16*(1), 1–10. https://doi.org/10.1145/2674026.2674028

Kutzbach, M. (2009). Motorization in Developing Countries: Causes, Consequences, and Effectiveness of Policy Options. *Journal of Urban Economics*, *65*(2), 154–166. https://doi.org/10.1016/j.jue.2008.10.002

Lam, W., & Mostafa, J. (2001). Modeling User Interest Shift Using a Bayesian Approach. *Journal of the American Society for Information Science and Technology*, *52*(5), 416–429.

Lana, I., Ser, J. Del, Velez, M., & Vlahogianni, E. I. (2018). Road Traffic Forecasting: Recent Advances and New Challenges. *IEEE Intelligent Transportation Systems Magazine*, *10*(2), 93–109. https://doi.org/10.1109/MITS.2018.2806634

Lazarescu, M. M., Venkatesh, S., & Bui, H. H. (2004). Using Multiple Windows to Track Concept Drift. *Intelligent Data Analysis*, *8*(1), 29–59. https://doi.org/10.3233/IDA-2004-8103

Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K., & Pollard, N. S. (2002). Interactive Control of Avatars Animated with Human Motion Data. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 491–500. https://doi.org/10.1145/566570.566607

Li, J., Chen, G., Chi, Z., & Lu, C. (2004). Image Coding Quality Assessment using Fuzzy Integrals with a Three-Component Image Model. *IEEE Transactions on Fuzzy Systems*, *12*(1), 99–106. https://doi.org/10.1109/TFUZZ.2003.822682

Li, Yan, Wang, T., & Shum, H.-Y. (2002). Motion Texture: A Two-level Statistical Model for Character Motion Synthesis. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 465–472. https://doi.org/10.1145/566570.566604

Li, Yanhong, Li, D., Wang, S., & Zhai, Y. (2014). Incremental Entropy-Based Clustering on Categorical Data Streams with Concept Drift. *Knowledge-Based Systems*, *59*, 33–47. https://doi.org/10.1016/j.knosys.2014.02.004

Lighthill, M. J., & Whitham, G. B. (1955). On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *229*(1178), 317–345. https://doi.org/10.1098/rspa.1955.0089

Littlestone, N. (1988). Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning*, *2*(4), 285–318. https://doi.org/10.1023/A:1022869011914

Liu, A. (2018). *Concept Drift Adaptation for Learning with Streaming Data* [PhD Thesis, University of Technology Sydney]. http://hdl.handle.net/10453/125627

Liu, A., Zhang, G., & Lu, J. (2017). Fuzzy Time Windowing for Gradual Concept Drift Adaptation. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–6. https://doi.org/10.1109/FUZZ-IEEE.2017.8015596

Liu, W., Zheng, Y., Chawla, S., Yuan, J., & Xing, X. (2011). Discovering Spatio-temporal Causal Interactions in Traffic Data Streams. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1010–1018. https://doi.org/10.1145/2020408.2020571

Lu, N., Zhang, G., & Lu, J. (2014). Concept Drift Detection via Competence Models. *Artificial Intelligence*, *209*, 11–28. https://doi.org/10.1016/j.artint.2014.01.001

Lühr, S., & Lazarescu, M. (2009). Incremental Clustering of Dynamic Data Streams Using Connectivity Based Representative Points. *Data & Knowledge Engineering*, *68*(1), 1–27. https://doi.org/10.1016/j.datak.2008.08.006

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. (2015). Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873. https://doi.org/10.1109/TITS.2014.2345663

Mahendra, A. (2008). Vehicle Restrictions in Four Latin American Cities: Is Congestion Pricing Possible? *Transport Reviews*, *28*(1), 105–133. https://doi.org/10.1080/01441640701458265

Maloof, M. A., & Michalski, R. S. (1995). A Method for Partial-Memory Incremental Learning and its Application to Computer Intrusion Detection. *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, 392–397. https://doi.org/10.1109/TAI.1995.479784

Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. (2006). Activity Recognition and Monitoring using Multiple Sensors on Different Body Positions. *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, 116–119. https://doi.org/10.1109/BSN.2006.6

Meadows, D. H. (2008). *Thinking in Systems: A Primer* (D. Wright, Ed.). Chelsea Green Publishing.

Meadows, D. H. (2015). *Thinking in Systems: A Primer* (1 edition). Chelsea Green Publishing Co.

Meadows, D. H., Meadows, D. L., & Randers, J. (1992). *Beyond the Limits: Global Collapse or a Sustainable Future.* Earthscan Ltd.

Meneely, P., Hoang, R. D., Okeke, I. N., & Heston, K. (2017). *Genetics: Genes, Genomes, and Evolution*. Oxford University Press.

Michau, G., Nantes, A., Bhaskar, A., Chung, E., Abry, P., & Borgnat, P. (2017). Bluetooth Data in an Urban Context: Retrieving Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2377–2386. https://doi.org/10.1109/TITS.2016.2642304

Mitchell, M. (2011). *Complexity: A Guided Tour* (1st Edition). Oxford University Press.

Mori, U., Mendiburu, A., Álvarez, M., & Lozano, J. A. (2015). A Review of Travel Time Estimation and Forecasting for Advanced Traveller Information Systems. *Transportmetrica A: Transport Science*, *11*(2), 119–157. https://doi.org/10.1080/23249935.2014.932469

Mouchaweh, M. S., Devillez, A., Lecolier, G. V., & Billaudel, P. (2002). Incremental learning in Fuzzy Pattern Matching. *Fuzzy Sets and Systems*, *132*(1), 49–62. https://doi.org/10.1016/S0165-0114(02)00060-X

Murphy, L., & Newman, P. (2011). Risky Planning: Path Planning Over Costmaps with a Probabilistically Bounded Speed-Accuracy Tradeoff. *2011 IEEE International Conference on Robotics and Automation*, 3727–3732. https://doi.org/10.1109/ICRA.2011.5980124

Muthukrishnan, S. (2005). Data Streams: Algorithms and Applications. *Foundations and Trends® in Theoretical Computer Science*, *1*(2), 117–236. https://doi.org/10.1561/0400000002

Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., Silva, D. D., Alahakoon, D., & Pothuhera, D. (2019). Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 1–12. https://doi.org/10.1109/TITS.2019.2924883

Nallaperuma, D., Silva, D. D., Alahakoon, D., & Yu, X. (2017). A Cognitive Data Stream Mining Technique for Context-Aware IoT Systems. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 4777–4782. https://doi.org/10.1109/IECON.2017.8216824

Nallaperuma, D., Silva, D. D., Alahakoon, D., & Yu, X. (2018). Intelligent Detection of Driver Behavior Changes for Effective Coordination Between Autonomous and Human Driven Vehicles. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3120–3125. https://doi.org/10.1109/IECON.2018.8591357

Nantes, A., Miska, M. P., Bhaskar, A., & Chung, E. (2014). Noisy Bluetooth traffic data? *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, *23*(1), 33–43.

Nash, J. F. (1950). Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences*, *36*(1), 48–49. https://doi.org/10.1073/pnas.36.1.48

Navarro-Gonzalez, J. L., Lopez-Juarez, I., Ordaz-Hernandez, K., & Rios-Cabrera, R. (2015). On-line Incremental Learning for Unknown Conditions During Assembly Operations With Industrial Robots. *Evolving Systems*, *6*(2), 101–114. https://doi.org/10.1007/s12530-014-9125-x

Nishida, K., & Yamauchi, K. (2007). Detecting Concept Drift Using Statistical Testing. In V. Corruble, M. Takeda, & E. Suzuki (Eds.), *Discovery Science* (pp. 264–269). Springer. https://doi.org/10.1007/978-3-540-75488-6_27

Obeng-Odoom, F. (2009). The Future of Our Cities. *Cities*, *26*(1), 49–53. https://doi.org/10.1016/j.cities.2008.11.001

Oliver, N., Horvitz, E., & Garg, A. (2002). Layered Representations for Human Activity Recognition. *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, 3–8. https://doi.org/10.1109/ICMI.2002.1166960

Pullen, K., & Bregler, C. (2002). Motion Capture Assisted Animation: Texturing and Synthesis. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, 501–508. https://doi.org/10.1145/566570.566608

Recordati, G., & Bellini, T. G. (2004). A Definition of Internal Constancy and Homeostasis in the Context of Non-Equilibrium Thermodynamics. *Experimental Physiology*, *89*(1), 27–38. https://doi.org/10.1113/expphysiol.2003.002633

Reiss, A., & Stricker, D. (2012). Introducing a New Benchmarked Dataset for Activity Monitoring. *Proceedings of the 2012 16th Annual International Symposium on Wearable Computers (ISWC)*, 108–109. https://doi.org/10.1109/ISWC.2012.13

Richards, P. I. (1956). Shock Waves on the Highway. *Operations Research*, *4*(1), 42–51. https://doi.org/10.1287/opre.4.1.42

Sayed-Mouchaweh, M. (2016). *Learning from Data Streams in Dynamic Environments*. Springer International Publishing. https://doi.org/10.1007/978-3-319-25667-2

Schlimmer, J. C., & Granger, R. H. (1986). Incremental Learning from Noisy Data. *Machine Learning*, *1*(3), 317–354. https://doi.org/10.1007/BF00116895

Shao, J., Ahmadi, Z., & Kramer, S. (2014). Prototype-based Learning on Concept-Drifting Data Streams. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 412–421. https://doi.org/10.1145/2623330.2623609

Shinar, D. (2017). *Traffic Safety and Human Behavior* (Second Edition). Emerald Group Publishing.

Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). Smart Factories in Industry 4.0: A Review of the Concept and of Energy Management Approached in Production Based on the Internet of Things Paradigm. *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 697–701. https://doi.org/10.1109/IEEM.2014.7058728

Sobhani, P., & Beigy, H. (2011). New Drift Detection Method for Data Streams. In *Adaptive and Intelligent Systems* (pp. 88–97). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23857-4_12

Spirkin, A. (1975). Dialectical Materialism. In T. J. Blakeley (Ed.), *Themes in Soviet Marxist Philosophy: Selected Articles from the 'Filosofskaja Enciklopedija'* (pp. 5–47). Springer. https://doi.org/10.1007/978-94-010-1873-9_2

Springael, J., & Kunsch, P. L. (2002). A Multicriteria-based System Dynamics Modelling of Traffic Congestion Caused by Urban Commuters. *Central European Journal of Operational Research*, *10*(1), 81–97.

Sterman, J. (2001). System Dynamics Modeling: Tools for Learning in a Complex World. *California Management Review*, *43*(4), 8–25. https://doi.org/10.2307/41166098

Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches and Open

Issues. *IEEE Communications Surveys Tutorials*, *22*(2), 1–38. https://doi.org/10.1109/COMST.2019.2962586

Street, W. N., & Kim, Y. (2001). A Streaming Ensemble Algorithm (SEA) for Large-scale Classification. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 377–382. https://doi.org/10.1145/502512.502568

Syed, N. A., Liu, H., & Sung, K. K. (1999). Handling Concept Drifts in Incremental Learning with Support Vector Machines. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 317–321. https://doi.org/10.1145/312129.312267

Takahashi, O., & Schilling, R. J. (1989). Motion Planning in a Plane Using Generalized Voronoi Diagrams. *IEEE Transactions on Robotics and Automation*, *5*(2), 143–150. https://doi.org/10.1109/70.88035

Tanco, L. M., & Hilton, A. (2000). Realistic Synthesis of Novel Human Movements from a Database of Motion Capture Examples. *Proceedings Workshop on Human Motion*, 137–142. https://doi.org/10.1109/HUMO.2000.897383

Thrun, M. C. (2018). *Projection-Based Clustering through Self-Organization and Swarm Intelligence*. Springer. https://doi.org/10.1007/978-3-658-20540-9

Toubakh, H., & Sayed-Mouchaweh, M. (2015). Hybrid Dynamic Data-Driven Approach for Drift-Like Fault Detection in Wind Turbines. *Evolving Systems*, *6*(2), 115–129. https://doi.org/10.1007/s12530-014-9119-8

Tran, D.-H. (2013). Automated Change Detection and Reactive Clustering in Multivariate Streaming Data. *ArXiv:1311.0505 [Cs]*. http://arxiv.org/abs/1311.0505

Tsymbal, A., & Puuronen, S. (2000). Bagging and Boosting with Dynamic Integration of Classifiers. In D. A. Zighed, J. Komorowski, & J. Żytkow (Eds.), *Principles of Data Mining and Knowledge Discovery* (pp. 116–125). Springer. https://doi.org/10.1007/3-540-45372-5_12

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-Term Traffic Forecasting: Where We Are and Where We're Going. *Transportation Research Part C: Emerging Technologies*, *43*, 3–19. https://doi.org/10.1016/j.trc.2014.01.005

Wang, H., & Schmid, C. (2013). Action Recognition with Improved Trajectories. *2013 IEEE International Conference on Computer Vision*, 3551–3558. https://doi.org/10.1109/ICCV.2013.441

Wang, Heng, Kläser, A., Schmid, C., & Liu, C.-L. (2013). Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision*, *103*(1), 60–79. https://doi.org/10.1007/s11263-012-0594-8

Wang, W., Xi, J., & Chen, H. (2014). Modeling and Recognizing Driver Behavior Based on Driving Data: A Survey. *Mathematical Problems in Engineering*, *2014*, 1–20. https://doi.org/10.1155/2014/245641

Widmer, G., & Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, *23*(1), 69–101. https://doi.org/10.1007/BF00116900

Wille, J. M., & Form, T. (2008). Realizing Complex Autonomous Driving Maneuvers the Approach taken by Team Carolo At The DARPA Urban Challenge. *2008 IEEE International Conference on Vehicular Electronics and Safety*, 232–236. https://doi.org/10.1109/ICVES.2008.4640889

Wu, J., & Loucks, O. L. (1995). From Balance of Nature to Hierarchical Patch Dynamics: A Paradigm Shift in Ecology. *The Quarterly Review of Biology*, *70*(4), 439–466. https://doi.org/10.1086/419172

Xu, S., & Wang, J. (2017). Dynamic Extreme Learning Machine for Data Stream Classification. *Neurocomputing*, *238*(C), 433–449. https://doi.org/10.1016/j.neucom.2016.12.078

Xu, W., Pan, J., Wei, J., & Dolan, J. M. (2014). Motion Planning Under Uncertainty for On-Road Autonomous Driving. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2507–2512. https://doi.org/10.1109/ICRA.2014.6907209

Yager, R. R. (1993). Element selection from a fuzzy subset using the fuzzy integral. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(2), 467–477. https://doi.org/10.1109/21.229459

Yin, J., Yang, Q., & Pan, J. J. (2008). Sensor-Based Abnormal Human-Activity Detection. *IEEE Transactions on Knowledge and Data Engineering*, *20*(8), 1082–1090. https://doi.org/10.1109/TKDE.2007.1042

Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017). Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. *Proceedings of the 2017 SIAM International Conference on Data Mining*, 777–785. https://doi.org/10.1137/1.9781611974973.87

Zeira, G., Maimon, O., Last, M., & Rokach, L. (2004). Change Detection in Classification Models Induced from Time Series Data. In M. Last, A. Kandel, & H. Bunke (Eds.), *Data Mining in Time Series Databases* (Vol. 57, pp. 101–125). World Scientific. https://doi.org/10.1142/9789812565402_0005

Zou, N., Liang, S., & He, D. (2020). Issues and Challenges of User and Data Interaction in Healthcare-related IoT: A Systematic Review. *Library Hi Tech*, *38*(4), 769–782. https://doi.org/10.1108/LHT-09-2019-0177