Semantic Annotation and Reasoning for Sensor Data Streams

Lianli Gao

A thesis submitted for the degree of Doctor of Philosophy at
The University of Queensland in 2015
School of Information Technology and Electrical Engineering
Abstract

Sensor technologies and wireless sensor networks are enabling the capture and storage of large volumes of sensor data streams. However there are a number of characteristics associated with sensor data streams that hinder the sharing, analysis and re-use of such data on the Web. For example, because sensor data is both temporal and spatial in nature, its multi-dimensionality, combined with variations in granularity, makes it more difficult to analyse and interpret. These issues have created major challenges associated with the management, representation, analysis and indexing of large volumes of sensor data streams. Consequently, there is an urgent need to markup sensor data streams with well-defined semantics to drive the development of advanced applications such as situation awareness, predictive models and event detection. Given well-structured and semantically annotated sensor data streams, it is possible to reason across annotated sensor data streams to deduce new or implicit knowledge, discover significant (and erroneous) data and events and answer complex queries.

This thesis focuses on the application area of ecosystem monitoring. As such, it investigates novel solutions to the semantic annotation and reasoning challenges associated with sensor data streams acquired by ecosystem scientists who are monitoring: a) species behaviour and b) micro-climate changes within environmentally-sensitive regions. Within this context, this thesis focuses on the design, implementation and evaluation of innovative methods to tackle different challenges associated with the semantic annotation and reasoning of two classes of sensor data: a) animal accelerometry data streams (acquired via animal-attached tri-axial accelerometers); and b) environmental sensor data streams (acquired from wireless sensor networks). These two categories of sensor data are of particular interest because they are rapidly growing in volume, they present different but similar challenges and there is a need to correlate them in order to determine if changes in the environment are impacting on species behaviour.

The first component of the thesis investigates optimum methods of combining domain expert annotations and machine learning to improve the precision and efficiency of semantic annotations on 3D accelerometry data streams (to support animal behaviour recognition and analysis). The second component seeks to minimize the cost and effort involved in developing training corpuses for machine learning approaches, by evaluating an Optimal Graph Learning approach to automatic semantic annotation of 3D accelerometry data streams. The third component of the thesis tackles the problem of detecting, annotating and filtering errors and outliers in sensor data streams, from wireless sensor networks, employed for environmental monitoring. The fourth and final component investigates, implements and evaluates an approach for reasoning across multiple environmental sensor data streams to infer higher level knowledge (fire weather indices to predict bush fire risk).
In addition to introductory and literature reviews of the field, this thesis provides detailed descriptions and evaluations of the following four original contributions to the field:

- the SAAR (Semantic Annotation and Activity Recognition) approach, which is designed to assist biologists to automatically recognize animal activities from 3D accelerometry data streams, by combining an expert tagging service with machine learning algorithms. The experimental results show that SAAR enables ecologists with little knowledge of machine learning techniques to collaboratively build classification models with high levels of accuracy, sensitivity and specificity. The results also indicate that SAAR is able to use data from surrogate individuals to qualify and quantify the association between individual behavioural modes and tri-axial accelerometry data streams and apply the resulting model to similar species.

- the OGL (Optimal Graph Learning) approach, which is designed to enable semi-automatic annotation of animal accelerometry data streams by more accurately encoding similarities between data points. The OGL approach is compared with SAAR, and the experimental results show that OGL outperforms SAAR consistently, especially with a smaller number of annotated training samples. Moreover, additional experiments investigating the classification of images from three real world image datasets, demonstrate the superiority of OGL over existing graph construction methods, and demonstrate comparable performance with state-of-the-art learning methods, that rely on large manually annotated training corpuses.

- the SOUE-Detector (Segment Outliers and Unusual Events Detector) approach, which adopts ontologies and expert-defined, machine-processable rules (that define correlations between sensor properties) to detect and distinguish between erroneous segment outliers and genuine unusual events for wireless sensor networks. Experiments on real world sensor network datasets reveal that the proposed approach is able to efficiently and accurately detect both erroneous outliers and unusual events by making use of sensor data trend similarities and correlations between sensor properties.

- the SFWI (Semantic Fire Weather Index) approach, which aims to estimate fire weather indices by reasoning across cleaned wireless sensor network data streams, represented in RDF. The experimental results demonstrate: comparable performance with the state-of-the-art detection methods; the ability of generating more precise, spatio-temporally finer-grained Fire Weather Indices than that are currently available; and greatly improved querying speed in terms of running repeated queries over an extended period.
Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.
Publications during candidature

- **Gao, L; Song, J; Nie, F; Yan, Y; Sebe, N & Shen H** 2015, ‘Optimal Graph Learning with Partial Tags and Multiple features for Image and Video Annotation’, Accepted, *2015 Computer Vision and Pattern Recognition Conference (CVPR)* - incorporated in Chapter 4.


- Bidder, O; Campbell, H; Gómez-Laich, A; Urgé, P; Walker, J; Cai, Y; **Gao, L; Quintana, F** 2014, ‘Love Thy Neighbour: Automatic Animal Behavioural Classification of Acceleration Data Using the K-Nearest Neighbour Algorithm’, *Journal of PLOS one*, vol.9, pp.e88609.


Publications included in this thesis


<table>
<thead>
<tr>
<th>Contributor</th>
<th>Statement of contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lianli Gao (Candidate)</td>
<td>Data collection (60%)</td>
</tr>
<tr>
<td></td>
<td>Designed the algorithms</td>
</tr>
<tr>
<td></td>
<td>Designed, implemented and evaluated the proposed approach</td>
</tr>
<tr>
<td></td>
<td>Wrote the paper</td>
</tr>
<tr>
<td>Hamish Campbell</td>
<td>Data collection (35%)</td>
</tr>
<tr>
<td></td>
<td>Applied for the ethics approval</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td></td>
<td>Contributed to the discussion</td>
</tr>
<tr>
<td>Owen R.Bidder</td>
<td>Data collection (5%)</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td>Jane Hunter</td>
<td>Contributed to the discussion</td>
</tr>
<tr>
<td></td>
<td>Provided feedback to the design</td>
</tr>
<tr>
<td></td>
<td>Edited the paper</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Statement of contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lianli Gao (Candidate)</td>
<td>Data collection (50%)</td>
</tr>
<tr>
<td></td>
<td>Designed the algorithms</td>
</tr>
<tr>
<td></td>
<td>Designed and implemented the experiments</td>
</tr>
<tr>
<td></td>
<td>Analysed the experimental data</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td></td>
<td>Wrote the paper (25%)</td>
</tr>
<tr>
<td>Hamish Campbell</td>
<td>Data collection (50%)</td>
</tr>
<tr>
<td></td>
<td>Wrote the paper (75%)</td>
</tr>
<tr>
<td>Owen Bidder</td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td>Jane Hunter</td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td>Craig Franklin</td>
<td>Reviewed the paper</td>
</tr>
</tbody>
</table>
- **Gao, L; Song, J; Nie, F; Yan, Y; Sebe, N & Shen H** 2015, ‘Optimal Graph Learning with Partial Tags and Multiple features for Image and Video Annotation’, Accepted, *2015 Computer Vision and Pattern Recognition Conference (CVPR)* - incorporated as Chapter 4.

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Statement of contribution</th>
</tr>
</thead>
</table>
| Lianli Gao (Candidate) | Data collection  
Designed the research approach  
Designed, implemented and evaluated the proposed approach  
Wrote the paper         |
| Jingkuan Song          | Contributed to the discussion  
Provided feedback to the design  
Contributed to the research idea  
Edited the paper  
Reviewed the paper |
| Feiping Nie            | Contributed to the discussion  
Provided feedback to the design  
Contributed to the research idea  
Reviewed the paper |
| Yan Yan                | Contributed to the discussion  
Provided feedback to the design  
Reviewed the paper |
| Nicu Sebe              | Contributed to the discussion  
Provided feedback to the design  
Reviewed the paper |
| Hengtao Shen           | Contributed to the discussion  
Provided feedback to the design  
Reviewed the paper |

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Statement of contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lianli Gao (Candidate)</td>
<td>Data collection&lt;br&gt;Designed the algorithms&lt;br&gt;Implemented and evaluated the proposed approach&lt;br&gt;Wrote the paper</td>
</tr>
<tr>
<td>Jane Hunter</td>
<td>Contributed to the discussion&lt;br&gt;Provided feedback to the design&lt;br&gt;Edited the paper&lt;br&gt;Reviewed the paper</td>
</tr>
<tr>
<td>Michael Brünig</td>
<td>Contributed to the discussion&lt;br&gt;Reviewed the paper</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Statement of contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lianli Gao (Candidate)</td>
<td>Data collection</td>
</tr>
<tr>
<td></td>
<td>Designed the algorithms</td>
</tr>
<tr>
<td></td>
<td>Implemented and evaluated the proposed approach</td>
</tr>
<tr>
<td></td>
<td>Wrote the paper</td>
</tr>
<tr>
<td>Jane Hunter</td>
<td>Contributed to the discussion</td>
</tr>
<tr>
<td></td>
<td>Edited the paper</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
<tr>
<td>Michael Brünig</td>
<td>Contributed to the discussion</td>
</tr>
<tr>
<td></td>
<td>Reviewed the paper</td>
</tr>
</tbody>
</table>
Contributions by others to the thesis
For all the published research work included in this thesis, Prof. Jane Hunter, as my principal supervisor, provided very helpful insights into the general research aims as well as technical approaches. She also assisted with the refinement of ideas and the pre-submission editing of the publications.

During this thesis research, Dr Michael Brünig, as my co-supervisor, provided me with access to sensor data streams (from the Springbroook wireless sensor network) as well as with ideas to improve the quality and analysis of data streams from wireless sensor networks.

Statement of parts of the thesis submitted to qualify for the award of another degree
None.
Acknowledgements

This thesis represents a major portion of the research undertaken during my PhD study at eResearch Lab, School of ITEE, The University of Queensland, Brisbane, Australia. I would like to take this opportunity to express my sincere gratitude to those who assisted me, in one way or another, during my PhD studies.

Firstly, I want to express my deepest gratitude to my principal supervisor Professor Jane Hunter, head of the eResearch Lab, The University of Queensland, for her valuable guidance on research direction and methodology, her supervision of key technical and theoretical problems, and her assistance with financial support. Her insightful and constructive criticism of numerous drafts and many other aspects of my research, that made this thesis possible.

Secondly, I would like to thank my co-supervisor Dr Michael Brünig, Director of the Digital Productivity Flagship in CSIRO, for giving me with priceless supervision with regard to wireless sensor networks. His valuable suggestions, encouragement, constructive comments, and evaluations contributed a great deal to the successful completion of my thesis as well as the improvement of my research skills.

I would like to extend my thanks to my earlier-stage co-supervisor Dr Yuan-Fang Li, lecturer at the School of Information Technology at the University of Monash, for providing me with the opportunity to study at the UQ eResearch Lab and for giving me never-ending encouragement and research guidance on Semantic Web technologies which enabled me to get through the first and the toughest year of my PhD.

I am deeply indebted to Ms Carol Owen, eResearch Executive Officer, for all her help with my PhD research and various aspects of daily life, in particular, the assistance she provided with various aspects of English language. I am sincerely grateful to her for her time and effort.

I gratefully acknowledge the previous writers whose publications informed and provided the groundwork and impetus for this thesis. I am also grateful to the external examiners who reviewed this thesis and provided critical comments and constructive feedback, which contributed to the clarification of many of the ideas presented in this thesis.

To my PhD committee members, Dr Mohamed Sharaf, Ms Anna Gerber and Dr Nigel Ward, thank you for your time and your careful examination of my research progress, and for your critical comments and professional recommendation on how to improve the quality of my PhD research during the PhD Confirmation, Mid-Candidature Review and Thesis Review processes.

To all my lab mates, thank you for your suggestions and help on all aspects of my research works. In particular, I thank all the PhD students in the eResearch Lab, especially Mr Suleiman Odat, Mr Razan
Paul, Mr Chih-Hao Yu, Ms Hasti Ziaimatin and Mr Hamed Hassanzadeh; your friendship and funny discussions gave me inspiration and helped me continue through the long and sometimes bumpy ride of my PhD study. It was my great pleasure to study with you all, side-by-side.

To my dear friends, Ms Dan Wu, Mr Jianghui Wu and Mr Lianye Zhang, who I met in Brisbane and with whom I had great fun, thank you for being my friends and for helping me learn how to relax my mind and maintain the elusive balance between work and life.

I wish to thank sincerely and deeply my parents who have raised me, helped me, taught me, supported me, and encouraged me; thank you for your endless love and for your continued faith in me.

I also wish to express my sincere and profound gratitude to my dearest sister, Xiaofang, who has been supporting me, encouraged me and always ‘has my back’ all the times. Also, I thank my brother-in-law, Qianjiang, who has been so supportive and kind during the past six years, as well as my beloved nephew, Shuzhan, who has been bringing hope and joy into the family, spreads laughter and fun whenever he goes.

Finally and most importantly, to my beloved husband, Jingkuan Song, your understanding, encouragement, support and ceaseless love during the past years of this research has sustained me through the challenges and joys of my life.
Keywords

Australian and New Zealand Standard Research Classifications (ANZSRC)
ANZSRC code: 080109 Pattern Recognition and Data Mining (20%)
ANZSRC code: 080309 Software Engineering (30%)
ANZSRC code: 080605 Decision Support and Group Support Systems (50%)

Fields of Research (FoR) Classification
FoR code: 0801 Artificial Intelligence and Image Processing (20%)
FoR code: 0803 Computer Software (30%)
FoR code: 0806 Information Systems (50%)
# Table of Contents

Contents

Table of Contents ................................................................. xvi
List of Figures ........................................................................... xx
List of Tables ........................................................................... xxii

Chapter 1: Introduction ................................................................. 1
  1.1 Background ........................................................................ 1
  1.2 Two Case Studies ................................................................. 3
  1.2.1 Case Study #1 - Behaviour Recognition from Animal Attached Accelerometry Data Streams ......................................................... 3
  1.2.2 Case Study #2 - Reasoning over Environmental Sensor Network Data Streams ................................................................. 5
  1.3 Aims, Objectives and Research Questions ........................................ 7
  1.3.1 Semantic Annotation and Activity Recognition ............................... 7
  1.3.2 Semantic Annotation Using Graph-based Learning ....................... 8
  1.3.3 Semantic Outlier and Unusual Events Detection .......................... 10
  1.3.4 Semantic Fire Weather Index .................................................... 11
  1.4 Main Contributions ............................................................... 12
  1.4.1 Semantic Annotation and Activity Recognition (SAAR) ............... 12
  1.4.2 Optimal Graph Learning for Automatic Annotation (OGL) ........ 13
  1.4.3 Semantic Outlier and Unusual Events Detector for Wireless Sensor Network Data Streams (SOUE-Detector) .......................... 14
  1.4.4 Estimating Semantic Fire Weather Indices via Semantic Reasoning over Wireless Sensor Network Data Streams (SFWI) ............... 15
  1.5 A Common Technical Framework ............................................. 15
  1.5.1 Underlying Semantic Annotation Model .................................... 19
  1.6 Thesis Outline ................................................................. 20

Chapter 2: Literature Review ......................................................... 22
  2.1 Introduction ........................................................................ 22
  2.2 Supervised-based Automatic Semantic Annotation of Animal Sensor Data Streams ................................................................. 22
  2.3 Optimal Graph-based Learning for Automatic Annotation of Animal Sensor Data Streams ................................................................. 25
  2.4 Outlier Detection for Wireless Sensor Network Data Streams ............... 27
  2.5 Reasoning across Multiple Wireless Sensor Network Data Streams .......................... 30
  2.6 Summary ........................................................................... 32

Chapter 3: Semantic Annotation and Activity Recognition for Animal Sensor Data Streams .... 34
  3.1 Overview ........................................................................... 34

Data Streams ................................................................................................................. 97

6.1 Overview .................................................................................................................. 97

6.2 Methodology .......................................................................................................... 98
  6.2.1 Case Study ......................................................................................................... 98
  6.2.2 Methodological Steps ...................................................................................... 100

6.3 Data Processing and Storage ................................................................................... 101
  6.3.1 Data Pre-processing ....................................................................................... 101
  6.3.2 Conversion of Cleaned Sensor Observations to RDF Triples ..................... 101
  6.3.3 Fire Weather Index Ontology ......................................................................... 102
  6.3.4 Storage Techniques ......................................................................................... 103

6.4 Semantic Inferencing .............................................................................................. 104
  6.4.1 Defining SPARQL Inference Rules ................................................................. 104
  6.4.2 Inferencing Algorithm ..................................................................................... 107
  6.4.3 IDW-Based Neighbourhood Region Prediction ........................................... 108

6.5 Implementation ....................................................................................................... 109
  6.5.1 System Architecture ......................................................................................... 109
  6.5.2 User Interface .................................................................................................. 111

6.6 Evaluation ............................................................................................................... 111
  6.6.1 Evaluation of the Accuracy and Precision of Fire Weather Index Calculations ...... 111
  6.6.2 Evaluation of the RDF Triple Stores and SPARQL Query Performance .......... 113
  6.6.3 Evaluation of System Usability ........................................................................ 114

6.7 Summary ................................................................................................................. 115
Chapter 7: Future Work and Conclusion ................................................................. 116

7.1 Overview ............................................................................................................. 116

7.2 Major Contributions ........................................................................................... 116

7.2.1 Semantic Annotation and Activity Recognition ........................................... 117

7.2.2 Optimal Graph-based Learning (OGL) for Automatic Annotation ............... 118

7.2.3 Semantic Outlier and Unusual Events Detector ............................................ 119

7.2.4 Semantic Fire Weather Index (SWFI) System ................................................ 120

7.3 Lessons Learnt and Future Research ................................................................. 120

7.4 Concluding Summary ......................................................................................... 123

References ................................................................................................................ 125
List of Figures

Figure 1.1 High-level architectural view of the proposed framework ........................................... 16
Figure 1.2: Extending the OAC model to describe an annotation on a segment of a 3D accelerometer data stream ................................................................. 20
Figure 3.1: FFT-based feature extraction algorithm ............................................................... 37
Figure 3.2: High-level architectural view of the SAAR system .............................................. 42
Figure 3.3: Screenshot of SAAR Plot-Video visualization interface and the annotation interface ... 44
Figure 3.4: User interface showing retrieval of specific annotations to train a C-SVC activity classifier .............................................. 45
Figure 3.5: Screenshot of the SAAR Interface with dog activity identification results .......... 45
Figure 3.6: The experimental results from applying a classifier trained using dog data to accelerometer datasets for different species .................................................. 52
Figure 3.7: Mean classification scores versus SL:SH (Spine Length: Spine Height) ............ 53
Figure 4.1: An overview of OGL .................................................................................. 58
Figure 4.2: Effect of K ......................................................................................... 71
Figure 5.1: The CESP ontology consists of an object property cesp:Correlation with a set of sub-object properties (LHS) and a sensor property class ssn:Property with a set of sub-classes (RHS). 76
Figure 5.2: An example of an expert defining a strong correlation (cesp:hasNegativeCorrelation) 84
Figure 5.3: Architectural overview of the SOUE-Detector system .......................................... 87
Figure 5.4: Screenshots showing User Interfaces for the SOUE-Detector .......................... 90
Figure 5.5: 36 sensor nodes deployed in the Springbrook National Park .......................... 92
Figure 5.6: Precision and recall for detecting erroneous segment outliers ..................... 93
Figure 5.7: Precision and recall for detecting unusual events with cesp:hasStrongCorrelation .... 93
Figure 5.8: Precision and recall for detecting unusual events with cesp:hasMediumCorrelation .... 94
Figure 5.9: Precision and recall for detecting unusual events with cesp:hasPositiveCorrelation ...... 94
Figure 5.10: Precision and recall for detecting unusual events with cesp:hasLinearCorrelation ...... 94
Figure 5.11: Precision and recall for detecting unusual events with cesp:hasCurvilinearCorrelation .... 95
Figure 5.12: Precision and recall for detecting unusual events with cesp:hasSimpleCorrelation .... 95
Figure 5.13: Precision and recall for detecting unusual events with cesp:hasComplexCorrelation .. 95
Figure 6.1: Study area - The Springbrook Wireless Sensor Network deployed in the Springbrook National Park ................................................................. 99
Figure 6.2: Instance of a temperature observation represented using the SSN, AWS, DUL, CF and Unit ontologies ........................................................................ 102
Figure 6.3: Applying the FWI ontology to describe the calculation of a high fire weather index from three sensors (WindSpeed WS_2, RelativeHumidity RH_2, AirTemperature AT_2) on SN-Node2 ......................................................... 103
Figure 6.4: Algorithm for saving sensor observations to the Multiple Repository RDF triple store ................................................................. 104
Figure 6.5: An example of applying SPARQL reasoning to an RDF dataset to infer a High fire weather index ................................................................. 106
Figure 6.6: Inference algorithm for inferring new FWIs ...................................................... 107
Figure 6.7: High-level architecture view of the SFWI system ........................................... 109
Figure 6.8: User interface for displaying SFWI Search and Browse results .................... 110
Figure 6.9: Similarities between FFDI estimates and SFWI estimates for period Jan- Jun 2012 ... 112
Figure 6.10: Comparison of BoM-FWIs with SFWIs for the period 09/01/2012 00:00:00 - 14/01/2012 00:00:00 ................................................................................................................................................................. 112
Figure 6.11: Execution time for running SPARQL queries over different time periods for new queries and repeat queries over single and multiple repositories respectively (NQ-1R, RQ-1R, NQ-MR, RQ-MR) .............................................................................................................................................................................. 114
List of Tables

Table 3.1: The subject animals with attached accelerometers .......................................................... 39
Table 3.2: The characteristics for each animal used in the study (BS: Body Mass; SL: Spine Length; SH: Spine Height above the ground) ............................................................................................................ 39
Table 3.3: Effect of NA (NA-number of annotated training data for each behaviour) on AP- average precision, AR-average recall. ST- Shortest Time for preparing Training Data Set.............................. 47
Table 3.4: Accuracy achieved from applying ANN, HMM and SAAR to the well-trained dog data set ........................................................................................................................................... 47
Table 3.5: Precision achieved from applying ANN, HMM and SAAR to the well-trained dog data set ........................................................................................................................................... 47
Table 3.6: Summary of time taken by user group to perform requested tasks .................................. 54
Table 4.1: Results on the dog dataset (NA - number of annotated training data for each behaviour; AP – average precision; AR - average recall)................................................................................................. 67
Table 4.2: Result on Corel5k .................................................................................................................. 69
Table 4.3: Results from using the ESP Game and IAPR TC-12 Data Sets ........................................... 70
Table 5.1 : Comparison of different outlier detection techniques....................................................... 96
Chapter 1: Introduction

In this chapter, the research topics and motivations that underpin this thesis are described. Specific sub-sections in this chapter include: the background; two case studies; aims, objectives and research questions; main contributions; the common technical framework; and the overall organization of the thesis.

1.1 Background

With the rapid development of micro sensor technology, sensors are becoming widely adopted to monitor everything from human health to air pollution [1-6]. However, a major challenge associated with sensor data streams is that they yield massive volumes of disparate, dynamic and geographically-distributed and heterogeneous sensor data streams that are difficult to store, index, search across, analyse, re-use and manage [7, 8].

Semantic annotation and reasoning of sensor data streams is a fundamental requirement to enable knowledge extraction, information retrieval and data mining of sensor data captured across a wide range of application areas. Below, three application areas are described that illustrate the significant role that semantic annotation and reasoning services play when applied to sensor data streams:

• *Environmental applications.* Sensor networks are increasingly deployed to: track the movements and behavior of animals and insects [1-4]; the impact of environmental conditions on livestock [6]; and changing rainforest ecosystems [9, 10]. Within these scenarios, semantic annotation can be applied to improve semantic interoperability and integration of multiple sensor data streams, as well as to facilitate reasoning, classification and automatic processing. Semantic annotation also facilitates the discovery of and access to sensor data on the Web [11]. In addition, semantic reasoning enables the formalization of interpretations of sensor data and the extraction of contextual knowledge [12]. The outcome for scientists is an improved understanding of the behavior of different species and interactions within complex eco-systems under changing environmental conditions.

• *Military applications.* Because sensor networks possess the following characteristics: rapid development, data-centric and application-oriented; they provide a very promising sensing technique for military applications, including monitoring of forces, battle damage assessment and biological attack detection [10]. Within these military applications, data fusion plays a major role in assisting decision makers by providing them with improved situation awareness.
Semantic annotation of sensor data in military applications improves the efficiency and accuracy of sensor data fusion and provides a more precise, fine-grained picture of the current situation (spatio-temporally, semantically and within the larger context) [13].

- **Health monitoring.** Health monitoring involves collecting sensor data (e.g., heart rate, body temperature, blood pressure) from patients to assist with medical diagnoses, health risk evaluation and medical treatment, as well as to underpin the development of automated and personalized approaches. Such applications require rapid integration of data from multiple sources to support individual situation awareness, particularly for the elderly or disabled [5]. Within this scenario, ontologies can be used to semantically annotate incoming sensor data streams with semantic descriptions to enable the rapid analysis and integration of multiple sensor observations and to assist medical experts in the decision making process [11].

Within this thesis, the primary focus is on sensor data streams associated with monitoring and understanding ecosystems, in particular species behavior and changing micro-climates. For example, recent years have witnessed tremendous advances in sensor hardware technology, such as the development of GPS-based devices, pedometers and accelerometers, which can be used to collect data that monitors spatial movement and location of a subject (human or animal) over time, to analyse and predict the subject’s behaviour. Consequently, animal biologists are taking advantage of low cost micro-sensor technology by deploying tri-axial (3D) accelerometers to monitor the behaviour and movement of a broad range of species, including endangered animals and invasive pests [14]. The result is an avalanche of complex tri-axial accelerometry data streams that capture observations and measurements of a wide range of animal body motion and posture parameters. Analysis of these parameters enables the identification of specific animal behavior. However, the process of analysing tri-axial accelerometry data streams is immature with the activity recognition process largely being undertaken manually and subjectively by animal scientists.

At the same time, sensor data streams are also being captured via wireless sensor networks, composed of a large number of sensor nodes, densely deployed within regions of interest to ecosystem scientists (such as rainforests). For example, the Springbrook sensor network [9] consists of more than two hundred sensor nodes in a rainforest region in south-east Queensland. Each sensor node is equipped with sensors for wind speed and direction, temperature, relative humidity, soil moisture and leaf wetness. Exploring such sensor network data streams has the potential for answering important ecological questions as well as predicting natural hazards. However, a number of limitations are associated with sensor networks that hinder the sharing and mining of sensor data streams on the Web. For example, sensor data is often incomplete or imprecise due to the huge volume of data streams being generated and poor signal strength, limited power and bandwidth associated with wireless
networks [15]. The raw sensor data is also numerical and unstructured, and its data quality largely depends on the context of the sensor network [16]. In addition, the adoption of heterogeneous, non-standard infrastructures, and poor data representation have resulted in much sensor data being locked inside specific applications and only accessible within organizational boundaries [17]. Moreover, the sensor data is both temporal and spatial in nature [18], so its multi-dimensionality makes it more difficult to be analysed and interpreted.

Subsequently, these issues have created major problems associated with the management, analysis, and indexing of the avalanche of sensor data streams being generated by both animal-attached sensors and wireless sensor networks to observe and measure the behaviour of animals and the environment, as well as to understand how animals adapt their behaviour in a changing environment.

1.2 Two Case Studies

Within the context of this thesis, two types of sensor data have been considered because: a) they represent two of the most common classes of sensor data; b) each of them presents different challenges associated with the annotation, analysis and management of sensor data streams; and c) there is a growing demand for tools to support the integration of these two types of sensor data streams. These two categories of sensor data and associated case studies (described in detail below) were also chosen because they generate large-scale sensor data streams that require cleaning, labelling and analysis in order to answer complex scientific questions. Moreover, the chosen data streams were also available for the purposes of this thesis (i.e., the collaborators generating the data granted approval for re-use of the datasets in the research described in this thesis), hence they provide ideal test beds for evaluating the proposed approaches developed within this thesis.

1.2.1 Case Study #1 - Behaviour Recognition from Animal Attached Accelerometry Data Streams

The recording of acceleration using animal-borne electronic devices is gaining popularity in animal behaviour research [1, 19-22]. Acceleration measurement includes both static components (due to gravity) and dynamic components (due to movements), which are recorded whilst the animal carries out routine behaviours [23]. Researchers use miniaturised logging devices to measure acceleration along three axes (tri-axial) (X, Y and Z). By calculating overall dynamic body acceleration (ODBA) researchers can estimate the energy-expenditure of the animal [24-26]. Although it has been recognised that integration of activity-specific metabolic rates with behavioural modes would better reveal the interaction between an animal and its environment [26], it has rarely been carried out
because of the difficulties associated with identifying different behavioural modes from accelerometry data.

To identify behavioural modes from acceleration readings, early studies used visual observation of the animal with the accelerometry data recording device attached [19, 27, 28]. More recently, pattern recognition and machine learning algorithms have been used to classify behavioural modes from accelerometry data streams collected both from domestic and free-ranging animals [20, 21, 29]. The application of machine learning algorithms to accelerometry data has the potential to automate behavioural mode identification and quantification for free-ranging animals. The drawback, however, is that for the algorithms to accurately identify each behavioural mode in the free-ranging animal, a period of observation is required to tag the datasets with precise behaviour to generate a training corpus. In the case of free-ranging or feral animals (unlike domestic animals), it is often impossible to observe the study animal and generate a ground truth or training corpus of accelerometry data streams that have been precisely and accurately tagged.

So the first research question/hypothesis is to determine whether training corpuses, supervised machine learning algorithms and classification models generated using data from surrogate test individuals (e.g., domestic animals or animals in zoos) can be used to automatically identify behavioural modes for similar species that are rare, wild or live in environments that prohibit direct visual observation. For example, could the training corpus and classification model for a domestic dog be applied to automatically classify the accelerometry data captured from a dingo in the wild?

The second problem is that machine learning methods typically require a large volume of annotated training data in order to achieve a satisfactory performance, and thus significant time, effort and cost is associated with manually annotating animal behaviour. Hence the second aim is to design an algorithm which can acquire good performance with a relatively small volume of manually annotated data. Graph-based learning is an efficient approach for modelling data generated via various machine learning approaches: unsupervised learning [30-32], supervised learning [33] and semi-supervised learning [31, 32, 34, 35]. An important advantage of working with a graph structure is its ability to naturally incorporate diverse types of information and measurements, such as the relationship between unlabelled data and labelled data. Among these graph-based learning schemes, semi-supervised learning (SSL), i.e., learning from both labelled and unlabelled data, has been widely studied and applied to many challenging tasks [31, 33, 36] such as image classification, image ranking and image annotation. By exploiting a large volume of unlabelled data with reasonable assumptions, SSL can reduce the need for expensive labelled data and thus achieve promising results especially for noisy labels [37].
Hence, the second hypothesis is that by designing an Optimal Graph Learning (OGL)-based semi-supervised approach to automatic recognition of animal accelerometry data, high quality classification results can be achieved, without the need for a large volume of manually annotated training data.

**Data Sources for Case Study #1**

The animal accelerometry data used to apply, evaluate and optimize the proposed semantic annotation methods (SAAR and OGL described in Chapters 3 and 4, respectively) was captured by the Ecology, Conservation and Organismal Biology Lab (ECO-Lab) of the University of Queensland, in collaboration with the eResearch Group of the University of Queensland, and the College of Science at the University of Swansea. 3D accelerometry data streams were captured from different species to assist biologists to study and understand the behaviour of those species in different contexts. The data was collected by attaching 3D accelerometers to the study animals and in some cases, also capturing video of the animals as a source of ground truth for tagging and evaluation purpose [38]. The datasets comprised:

- 3D accelerometry data (and associated videos) captured from domestic dogs in Brisbane, Australia.
- 3D accelerometry data (and associated videos) captured from dingoes, cheetahs, tigers, wombats, kangaroos and echidnas at Australia Zoo.
- 3D accelerometry data streams (and associated videos) captured from Eurasian badgers at West Hatch RSPCA Centre, Somerset, UK.

**1.2.2 Case Study #2 - Reasoning over Environmental Sensor Network Data Streams**

Significant prior research into applying Semantic Web technologies to sensor data has focussed on how to extend the Open Geospatial Consortium (OGS)’s Sensor Web Enablement standards [39] to provide enhanced semantic descriptions for a single data stream [11, 40-44]. However, little attention has been paid to complex domains, such as micro-climate studies, which depend on the integration and correlation of multiple sensor data streams. The data analysis in such domains involves the study of mutual interaction between different kinds of sensor data streams. It is not restricted to enriching a single sensor data stream with semantic metadata. For example, if environmentalists want to identify the potential for a fire weather event, they have to collect and analyse multiple sensor data streams, including the air temperature, relative humidity, wind speed, and wind direction, and then determine the fire weather danger levels from a weighted combination of these parameters.
In addition, two other significant research challenges are associated with the classification and interpretation of sensor data streams. They are the issues of: (i) data quality or reliability and (ii) reasoning across multiple simultaneous streams. How can erroneous data streams be automatically detected and filtered to improve the quality of sensor data streams? How can high-level knowledge be derived from multiple low-level sensor data streams via the rule-based reasoning?

Firstly, the data quality associated with wireless sensor networks is unreliable because wireless sensor networks have several hardware restrictions, e.g., limited battery power, limited memory, limited computational capacity and limited communication bandwidth of the wireless links that connect sensor nodes and sensors [45]. Wireless sensor networks are comprised of hundreds of inexpensive and battery-operated sensors that frequently suffer from poor data quality (e.g., noise, errors, abnormal patterns, missing and redundant data). Such outliers significantly affect the accuracy and reliability of the information inferred from wireless sensor networks and adversely impact on the usefulness of the data in decision-making. Hence, one of the major challenges associated with wireless sensor networks is quality control of the sensor data streams – and more specifically the detection of segment outliers and the differentiation between outliers that are errors and outliers that represent unusual events. Using Semantic Web technologies to capture and formally represent domain expert knowledge about outliers has potential to improve outlier detection because such knowledge can be used to provide important contextual information.

Secondly, research focussed on mechanisms for extracting new or implicit knowledge hidden within wireless sensor data streams is still at a preliminary stage. For example, Wei and Barnaghi [40], demonstrated how rule-based reasoning can be performed over annotated sensor data to derive new knowledge. Sheth, Henson and Sahoo [11] illustrated how to make use of the rule-based reasoning to predict weather conditions. However, there has not been an intensive study of data stream reasoning across multiple sensor streams to infer complex events. Moreover, the stream reasoning of semantic sensor data is still a critical problem that needs to be solved to infer previously unknown knowledge, predict future events, or answer complex questions from users [46-49].

**Data Source for Case Study # 2**

The Springbrook Wireless Sensor Network project, undertaken by the CSIRO, the Department of Environment and Resource Management (DERM) and the Australian Rainforest Conservation Society (ARCS), aims to identify environmental changes in the microclimate of a specific world heritage region [9]. This project has involved the deployment of a wireless sensor network in a rainforest ecosystem in South East Queensland by a team of CSIRO scientists. It aimed to provide a valuable research platform to study the effects of invasive species on biodiversity, the ecological
processes of a rainforest and the impacts of climate change. The project employed hundreds of solar-powered sensor nodes to collect microclimate monitoring data, such as air temperature, relative humidity, light, leaf wetness, soil moisture, wind speed, air pressure, fog and cloud patterns. In addition, it adopted wireless sensor technologies to transmit the monitoring data back to a central hub, and then to databases that are accessible via authenticated online access.

The data acquired from the Springbrook Wireless Sensor Network has been employed in Chapters 5 and 6 to apply and evaluate the semantic reasoning services – firstly to detect outliers and erroneous events and then to infer high level events or knowledge (fire weather indices).

1.3 Aims, Objectives and Research Questions

The aim of this PhD thesis is to design and evaluate a suite of novel Semantic Annotation and Reasoning approaches, which are specifically designed to analyse and tag sensor data streams and infer new knowledge from them. The aim is to improve the efficiency, accuracy and quality of the semantic annotation and reasoning by integrating Semantic Web, expert knowledge, machine learning, statistical analysis, rule-based reasoning and Web 2.0 technologies. The resulting ontology-based annotation and reasoning techniques aim to facilitate the discovery, analysis, aggregation and sharing of sensor data streams (both animal accelerometry data streams and environmental sensor data streams). More specifically, this thesis describes and evaluates four novel approaches, described below, that employ innovative methods to tackle different research challenges associated with the semantic annotation and reasoning of sensor data streams, within the context of the two case studies described above.

1.3.1 Semantic Annotation and Activity Recognition

The first phase of this thesis research (described in Chapter 3) aims to design and evaluate a novel Semantic Annotation and Activity Recognition (SAAR) method to assist biologists to automatically recognize animal activities from 3D accelerometry data streams, using a combination of expert tagging and supervised learning algorithms. This phase of the thesis (which focuses on Case Study #1 above) involves the following objectives:

- To design a feature extraction method that extracts features vectors from manually annotated training data and to train a supervised learning classifier that automatically annotates newly generated accelerometry data streams;
- To design an automatic semantic annotation tool that can recognize animal activities from 3D accelerometry data streams by using a combination of expert tagging and supervised learning algorithms;
To assess the quality of results generated using supervised machine learning-based activity recognition classifiers that have been trained using manually annotated data for a variety of species (e.g., dogs and badgers);

To determine whether an activity recognition classifier trained using data from one species (e.g., a domestic dog) can be usefully applied to other species (e.g., a badger) or to a wild/free-ranging species (e.g., a dingo), of similar size and gait;

To enable the sharing, re-use, and refinement of activity recognition classifiers built for specific species, among scientists. By providing a common repository for animal accelerometry data sets, species-specific classifiers would improve over time, as more accelerometry data is uploaded, manually annotated and added to the training corpus;

To evaluate the proposed methods on real data sets provided by collaborators who are animal behaviour scientists.

In this phase of the thesis, the following research questions are explored:

- Which underlying annotation ontology can provide an annotation system for semantic annotations with a high-level of openness, interoperability, re-usability, and share-ability (e.g., the Open Annotation Collaboration (OAC) data model)?
- How can supervised learning techniques be combined with semantic annotation services to facilitate the automatic analysis of 3D accelerometry data streams?
- How can a supervised learning approach be adopted to recognize significant patterns and automatically tag sensor data streams (based on a corpus of manually tagged data streams)?
- How can semantic annotations be utilized to prepare a training data set and handle the noise and infeasibility of learning from a large sensor dataset?
- Can a tame/domestic surrogate be used to build a behavioural classification module, which is then applied to accurately identify and quantify behavioural modes within accelerometry data streams collected from other (feral or free-ranging) individuals/species?
- How can the performance of supervised machine learning classifiers and the usability of the developed systems be comparatively evaluated?

### 1.3.2 Semantic Annotation Using Graph-based Learning

One of the limitations of the approach proposed in Section 1.3.1 above, is that the quality of the automatic activity recognition results, is dependent on the size of the manually annotated training corpus. The hypothesis underpinning this aspect of the thesis, is that by using optimal graph-based learning (in conjunction with semi-supervised learning SSL), comparable classification results can be achieved with a smaller training corpus (or volume of labelled data).
An important advantage of a graph structure is its ability to naturally incorporate diverse types of information and measurements, such as the relationship between unlabelled data and labelled data. By exploiting the large volume of unlabelled data with reasonable assumptions, SSL can reduce the need for expensive labelled data and thus achieve promising results especially for noisy labels [37]. Therefore, the second phase of this thesis research (documented in Chapter 4) proposes applying an Optimal Graph Learning (OGL) algorithm for learning an optimal graph from multi-cues (i.e., initial labels and multiple-modality features). The proposed approach should be able to more accurately encode the relationships between data points (features in the accelerometry data streams), improve the efficiency of the semantic annotation process, and reduce the size of the training corpus required to accurately classify new data streams.

Hence, this phase of the thesis (which also focuses on Case Study #1 above) involves the following objectives:

- To propose an OGL approach for graph-based learning tasks;
- To incorporate OGL with a semi-supervised learning model and further extend the model to address out-of-sample and noisy label issues;
- To construct the graphs from multiple cues (labels and features);
- To improve the efficiency of animal behaviour identification given a smaller labelled training data set, by incorporating OGL with a semi-supervised learning model to automatically identify animal behaviours;
- To evaluate the OGL-based method by comparing the results with the SAAR approach; and
- To evaluate the applicability of OGL to more general image annotation task by comparing OGL with the state-of-the-art graph-based methods for image annotation;

In this phase of the thesis, the following research questions are explored:

- How can OGL be integrated with SSL to improve automated semantic annotation of sensor data streams (i.e., animal accelerometry data)?
- What is the best method to address out-of-sample extensions and noisy label issues?
- How should graphs be constructed using multiple information cues?
- How does the OGL approach compare with the SAAR approach?
- How can OGL be compared with the traditional state-of-the-art methods when applied to general image annotation?
1.3.3 Semantic Outlier and Unusual Events Detection

The third phase of this thesis research (described in detail in Chapter 5) aims to use ontologies and rules defining correlations between sensor attributes to detect and distinguish between erroneous segment outliers and genuine unusual events for wireless sensor networks. Using the case study and data described in Case Study #2 above, this component of the thesis has the following objectives:

- To define a Correlation of Environmental Sensor Properties (CESP) ontology (by extending the Semantic Sensor Network (SSN) ontology [50] developed by the Semantic Sensor Network Incubator Group) to describe correlations between specific sensor properties. For example, air temperature has a correlation with relative humidity since air temperature increases as relative humidity decreases [51].
- To design an algorithm to detect segment outliers and unusual events for wireless sensor networks by integrating statistical analysis techniques with Semantic Web technologies and domain expert knowledge about correlations. This algorithm not only takes into account the dynamic nature and variability of sensors, sensor nodes and sensor networks, but also integrates information about other types of correlations apart from spatio-temporal correlations between sensor properties.
- To develop a SOUE-Detector system based on the designed algorithm that validates the proposed methodology by applying it to a real dataset - the Springbrook wireless sensor network dataset [9]. The system should enable users to search the Springbrook dataset and retrieve data streams for a particular time period and automatically tag outlying segments as “error” or “unusual event” displayed in a visualization interface; and
- To evaluate the proposed detection algorithm in terms of precision and recall.

In addition, this research raises a series of research questions described as below:

- How can Semantic Web technologies be combined with statistical analysis to improve the quality and reliability of sensor data streams?
- How can Semantic Web technologies be usefully applied to outlier detection, analysis and filtering?
- How can domain specific expert knowledge about sensor properties correlations be captured and used to facilitate the segment outliers and unusual events detection?
- How can the dynamic nature of wireless sensor networks be handled by data management systems?
1.3.4 Semantic Fire Weather Index

The fourth and final phase of this thesis (described in Chapter 6) aims to develop a semantic reasoning approach for estimating fire weather indices over multiple environmental properties captured via wireless sensor networks. More specifically, this component focuses on Case Study #2 above, and has the following aims:

- To apply the SOUE-Detector approach to detect and filter the outliers within the raw sensor data streams to improve the quality of the data streams prior to deriving the fire weather indices;
- To develop a Fire Weather Index (FWI) ontology (in OWL, Web Ontology Language) to represent different levels of fire weather danger ratings. These fire weather danger ratings are based on input from the meteorological domain experts;
- To define First-Order-Logical inference rules for estimating fire weather indices, and then convert these rules into SPARQL inference rules by using SPARQL and OWL ontologies;
- To design an efficient storage technique for storing, querying and retrieving large volumes of sensor observation RDF triples efficiently. A novel multiple repository storage method is designed;
- To design an inference algorithm to infer the fire weather indices (FWIs) for a specific region and a given time period by combining SPARQL inference rules with an Inverse Distance Weighting [52] based approach. This combined approach enables accurate spatial distributions of FWIs to be inferred from the point data;
- To employ the proposed method to develop a Web based System that enables users to search, explore and visualize fire weather indices within a time period for a specific region - using Google Earth, Google timeline and Google pie chart visualizations.
- To evaluate the performance of the inference algorithm on real data sets (from Case Study #2);
- To evaluate the performance of the RDF triple stores and SPARQL querying for storing and querying over the derived Semantic Fire Weather Indices.

The related research questions that has been investigated are:

- How can the efficiency and accuracy of semantic reasoning be improved by making use of domain expert knowledge?
- How can rule-based semantic reasoning be exploited to assist the discovery of new or implicit knowledge to answer complex queries, predict future events, or highlight significant events in the data?
• How can this service be implemented in a scalable way that supports high performance for analysing current large-scale datasets as well as anticipated future data volumes?

1.4 Main Contributions

In order to address the problems outlined in Sections 1.2 and 1.3, innovative solutions have been proposed, designed, and implemented and these approaches are evaluated via experiments. In this section the contributions of this thesis to advancing and resolving these outstanding research problems are listed.

1.4.1 Semantic Annotation and Activity Recognition (SAAR)

As mentioned in Section 1.2.1, accelerometer data loggers are being used increasingly to assist ecologists to quantify animal activity, estimate energy expenditure, preserve endangered species and manage feral pests [19, 26]. However, little work has been undertaken to assess whether surrogate test individuals could be used to precisely qualify and quantify the association between individual behavioural modes and tri-axial accelerometry data streams to develop classifiers that can be applied to wild or free-ranging species for which there is no ground truth (videos or observations). Consequently, in the first phase (described in Chapter 3), a novel approach is designed and evaluated by combining a supervised machine learning algorithm Support Vector Machine (SVM) with expert tagging to automatically recognize animal activities from 3D accelerometry data streams (Case Study #1), and the following contributions are made:

• A novel method is proposed to enable scientists to quickly and easily analyse, tag and visualize of 3D accelerometry data streams and the synchronized videos. Also, the tagged sensor data streams are formatted in an RDF format by adopting and extending Open Annotation Collaboration data model [53], that enables them to be shared and reused between systems and users.

• A new feature extraction approach is proposed to extract features vectors from manually annotated training data to train a machine learning classifier that automatically annotates newly generated accelerometry data streams;

• The applicability of a behaviour recognition model developed using data from a tame or domestic species, to accelerometry data streams captured from feral or wild species, is determined;

• The accuracy, precision, specificity and sensitivity of behavioural classification modules for identifying behavioural modes from acceleration feature vectors collected from different individuals and species are evaluated. The evaluation results indicate that the SAAR approach
enables ecologists with little knowledge of machine learning techniques to collaboratively build classification models with high levels of accuracy, sensitivity, precision and specificity.

- A semantic annotation and activity recognition system is implemented to demonstrate the effectiveness of the proposed approach for solving real-life problems. This system supports the following functionalities:
  
  - It provides a repository on the Web where researchers monitoring animal behaviour, can upload and share their datasets, as well as search, retrieve and compare datasets from the same or different species.

  - It provides interactive graphical visualization services that enable scientists to quickly and easily view and explore tri-axial accelerometer data streams and temporally align simultaneously recorded video (where available) that can be used to verify (ground truth) behavioural activities;

  - It provides a platform by which ecologists can interactively record, share, and re-use domain expert knowledge on animal movements within tri-axial accelerometer data streams in an interoperable, re-usable manner;

  - It provides a set of Web services that can be used to analyse, tag and visualize 3D accelerometry datasets and synchronized video using terms from controlled vocabularies (pre-defined ontologies);

  - It enables ecologists to build their own automatic activity recognition models by training classifiers using features extracted from pre-annotated training sets.

1.4.2 Optimal Graph Learning for Automatic Annotation (OGL)

Graph-based learning is a promising paradigm for modelling the manifold structures that often exist in massive data sources in high-dimensional spaces. It has been shown to be effective in many emerging applications such as annotation, classification, ranking and retrieval [31, 33, 36]. The graph construction scheme essentially determines the performance of graph based learning algorithms. However, most of the existing works construct the graph empirically, and are usually based on a single information cue. The second component of this thesis (described in Chapter 4), proposes learning an optimal graph (OGL) from multi-cues (i.e., initial labels and multiple-modality features), because this method more accurately encapsulates the relationships between data points.

In this phase of the thesis, Optimal Graph Learning (OGL) is combined with semi-supervised machine learning (SSL) to try to improve the recognition of animal behaviour in accelerometry data streams (Case Study #1), whilst using smaller training corpuses. The contributions of this phase can be summarized as below:
• A series of new algorithms for various graph-based learning tasks are developed, based on the learned optimal graph. More specifically, OGL is incorporated with a semi-supervised learning model. The proposed model is further extended to address out-of-sample and noisy label issues.

• The efficiency of animal behaviour identification is improved by incorporating OGL with a semi-supervised machine learning model to automatically identify animal behaviours with smaller amounts of labelled training data set.

• Moreover, extensive experiments on real-world image datasets show the consistent superiority of OGL over the state-of-the-art graphs for traditional image annotation tasks.

1.4.3 Semantic Outlier and Unusual Events Detector for Wireless Sensor Network Data Streams (SOUE-Detector)

In the third component of this thesis (described in Chapter 5), an outlier detection approach is proposed to detect and distinguish between erroneous segment outliers and genuine unusual events for wireless sensor networks data streams (Case Study #2). The contributions of this phase can be summarized as below:

• The quality of the sensor data streams that being generated from wireless sensor networks is improved by using Semantic Web technologies to detect outliers (outlying segments) and distinguish between those outliers that are errors (and should be ignored or filtered) and those outliers that are generated by genuine but unusual events;

• A set of common vocabularies and ontologies are defined to describe correlations between specific sensor properties (e.g., the Correlated Environmental Sensor Properties (CESP)) ontology. These vocabularies enable scientists to capture domain specific knowledge about correlations between specific sensor properties to support outlier detection;

• An algorithm is designed that analyses a corpus of sensor data streams, compares data from neighbouring sensors to detect segment outliers and unusual events and automatically tags segments with “error” or “unusual event”; 

• The proposed detection algorithm is evaluated on real-world datasets (from Case Study #2) and the experimental results show that the SOUE-Detector can efficiently detect segment outliers and unusual events with high levels of precision and recall;

• A Web interface is implemented that demonstrates the effectiveness of the SOUE-Detector by enabling users to search and browse across sensor data streams and retrieve data streams
for a particular time period: “error” or “unusual event”, and display the results in a visualization interface.

1.4.4 Estimating Semantic Fire Weather Indices via Semantic Reasoning over Wireless Sensor Network Data Streams (SFWI)

The fourth and last component of this thesis (described in Chapter 6) investigates the application of semantic reasoning over multiple environmental properties captured via wireless sensor networks to calculate Fire Weather Indices (using Case Study #2 data). This phase makes the following original contributions:

- A novel rule-based reasoning mechanism is designed to mine the annotated sensor data streams to discover new or implicit knowledge using Semantic Web technologies, Resource Description Framework (RDF) [54] and SPARQL [55];
- A Fire Weather Index (FWI) ontology in Web Ontology Language (OWL) [56] is designed for describing different levels of fire weather danger ratings;
- A Semantic Fire Weather Index (SFWI) method is proposed by combining data preprocessing techniques, with semantic reasoning technology and domain expert knowledge to estimate fire weather indices from wireless sensor data streams collected from a wireless sensor network deployed in the Springbrook region of South East Queensland (Case Study #2);
- The performance of the proposed approach is evaluated in terms of accuracy and precision and the results show that the proposed approach outperforms state-of-the-art techniques in terms of accuracy, precision and query performance; and
- A Web-based system is implemented for inferring the fire weather indices for Springbrook region by employing the proposed SFWI approach. User-friendly visualization interfaces have been designed to enable users to easily visualize, browse and interact with the service to understand how fire weather index patterns change over time within a particular region.

1.5 A Common Technical Framework

To integrate the different services and technical components required to evaluate and apply the proposed methods outlined above, a common technical framework was designed. This integrated framework streamlined the development effort, reduced duplication and enabled re-use of software components (Fig.1.1).
Figure 1.1 High-level architectural view of the proposed framework

The integrated framework comprises the following major components:

- The bottom layer is the **Storage Layer** which stores different categories of scientific sensor data, including:
  - The 3D accelerometry data collected from different species (e.g., domestic dogs, dingoes, kangaroos and tigers) (Case Study #1); and
  - The environmental observation data streams (e.g., air temperature, air humidity, leaf wetness, wind speed) collected from the Springbrook wireless sensor network (Case Study #2).

  This layer employs a PostgreSQL database to store the different datasets that are encoded in TXT or CSV formats.

- The second layer, the **Machine Processing Layer**, is responsible for five main activities:
  - **Semantic Annotation Store**: where the RDF annotation data is stored in an OpenRDF Sesame repository. The technical components include Java, Simple Logging Façade for Java, Apache Tomcat, Eclipse and Sesame. Specifically, annotations generated from the annotation services are stored in an RDF Sesame repository using the OAC
data model (see section 1.5.1 below). Moreover, domain expert knowledge about correlations between sensor properties is also captured and stored in an RDF Sesame repository. In addition, the wireless sensor network data streams (from Case Study #2) are filtered and converted into RDF triples and also saved in a Sesame RDF repository.

- **Ontology Registry** - A number of existing OWL ontologies (SSN [50] and AWS [57]) are adopted and extended and new ontologies are developed (CESP) to describe sensors, sensing, the measurement capabilities of sensors, the observations that result from sensing, the deployments in which sensors are used, and the correlations between sensor properties. The ontology registry is used to store these OWL ontologies. The new OWL ontologies are developed using the Protégé ontology editor.

- **Statistical Analysis Services** - The required statistical algorithms are implemented using a combination of Java and Matlab, R, LibSVM and Optimal Graph Learning (OGL).

- **Inferencing/Reasoning Engine** - this is developed by combining SPARQL rules and OWL ontologies to infer higher level semantic events, such as reasoning about fire weather indices for a specific region. This component is implemented by using Web 2.0 technologies (Java, JavaScript, and JSON). In addition, this component applies Semantic Web technologies (RDF, OWL ontologies, SPARQL, and multiple RDF Sesame Repository Stores) to enrich sensor data with domain-specific semantic metadata as well as to apply semantic reasoning across the sensor data streams.

- **SPARQL Query Rules**: which support the SPARQL Protocol for RDF. These rules are implemented using SPARQL CONSTRUCT and SPARQL UPDATE requests (INSERT and DELETE).

- The third **Service Layer** is composed of semantic annotation services; statistical analysis services; inferencing services and search, browse, reporting services.

  - The **manual semantic annotation service** adopts and extends the Open Annotation Collaboration (OAC) data model [53] to describe annotations in an RDF format that enables them to be shared between systems and users. Additional domain-specific ontological extensions are incorporated as required for specific studies (ontologies are to describe animal behaviours or fire weather indices). To implement the annotation service, the following technologies are adopted: JavaScript, Ajax and JSON; the integrated development environment (MyEclipse), and an annotation server (OpenRDF Sesame) [58]. The annotation server supports the storage, search, and
retrieval of manually created annotations across multiple disciplines and applications. In addition, the manually created annotations are stored in a backend RDF triple store implemented by using the Apache Tomcat Java server and Sesame 2.0.

- The **automatic semantic annotation service** applies supervised machine learning algorithms or optimal graph learning based semi-supervised machine learning algorithms to manually tagged data streams to extract feature vectors and train a classification model to automatically identify activities within the new data streams. The annotation service is developed using a combination of Java, Ajax, Flot (a plotting jQuery library) [59], HTML 5 Video Player library (Video.js) [60] with JavaScript. The automatic activity recognition component is implemented using Libsvm - a Java library for Support Vector Machines [61] and a high-level technical computing language Matlab.

- The **statistical analyses services** employ statistical analyses or machine learning algorithms at the machine-processing level (Matlab, R, LibSVM and Optimal Graph Learning (OGL)) to process new data streams and automatically classify animal behaviour and environmental sensor network streams. These services are implemented using: the programming languages - Java, JavaScript, jQuery and JSON; a platform, Protégé, which supports the creation, visualization, and manipulation of ontologies in formats including OWL, RDF(S) and XML; and an object-relational database management system (PostgreSQL) to store sensor data streams.

- The **inferencing services** employ SPARQL rules and OWL ontologies to infer higher level semantic events, such as reasoning about fire weather indices for a specific region.

- The **search, browse and reporting services** provide interfaces to users to enable them to search the RDF Sesame repositories by inputting SPARQL queries via the user interface to the SPARQL endpoint. Implementation of the these services is via Eclipse, Java, JavaScript, Apache Tomcat and OpenRDF Sesame.

- The top layer is the **User Interface Layer**, which provides user interfaces to the search, browse and reporting services, manual annotation services and map or timeline visualization services that enable users to explore and visualize the sensor data and associated classifications over space and time. It has been implemented primarily using HTML 5, AJAX and JavaScript as well as using Web-based visualization technologies (Google Earth, Keyhole Markup Language (KML), SIMILE Timelines [62], Google Timelines, and Google Pie Charts). Flot,
a JavaScript plotting library for jQuery [59] was also used for visualizing and browsing sensor data stream values graphically via timelines.

1.5.1 Underlying Semantic Annotation Model

Although a wide range of pre-existing annotation systems are available [63], the majority are stand-alone and do not support Web-based interoperability of the services or the annotations across platforms, clients or collections. In an effort to support consistency of annotation data models and interoperability across boundaries of annotation clients and users, recent effort has focussed on approaches to enable interoperability of semantic annotations through Semantic Web, Linked Data and common ontological approaches. In particular, the Open Annotation Collaboration (OAC) data model [53] is an ideal candidate as the underlying annotation data model for the research aims of this thesis, for the following reasons:

- The core entities of the OAC model are Web resources identified via HTTP URIs;
- The OAC provides a flexible and extensible data model that maximizes interoperability and sharing;
- To maximize annotation interoperability, the set of top-level classes and relationships of the OAC model can be extended and refined to meet domain-specific needs;
- The OAC model adopts the Linked Data architecture to enable the discovery and sharing of annotations across annotation clients, servers, and collections;
- The OAC model supports multiple annotation Bodies and/or multiple Targets; and
- The OAC model supports annotations that are about a part of resource through the use of Selectors (e.g., a segment of a sensor data stream).

In addition, the OAC data model is implemented using Semantic Web technologies for formally representing knowledge, including RDF, RDF Schema (RDFS) [64], and OWL ontologies. These standard formats are ideal for representing annotations on spatio-temporal data streams [11]. One of the aims of this thesis is to evaluate the application of the OAC model to annotations on sensor data streams. For example, Figure.1.2 illustrates how the OAC model can be applied to represent an annotation on a segment of an animal accelerometry data stream - that was created by a user named ‘Lianli Gao’ on Fri May 18, 2012 and whose body comprised the semantic tag “ab:Running”.

19
1.6 Thesis Outline

Chapter 2 provides a literature review of past work focussing on semantic annotations of 3D accelerometry data streams, graph-based learning, outlier detections for wireless sensor data streams, and reasoning across multiple wireless sensor network data streams.

Chapter 3 describes the design and evaluation of the Semantic Annotation and Activity Recognition (SAAR) method, which combines supervised machine learning with expert tagging to automatically recognize animal activities from 3D accelerometry data streams (Case Study #1).

Chapter 4 describes the design and evaluation of the optimal graph learning (OGL) approach based on multiple cues (i.e., initial labels and multiple-modality features), to more accurately represent similarity between features as embedded relationships between data points within graphs. It also evaluates the proposed OGL+SSL approach when applied to animal accelerometry data (Case Study #1), by comparing it with SAAR. The OGL+SSL approach is also evaluated more generally by comparing it with traditional graph-based methods in the context of image classification.

Chapter 5 presents the design, implementation and evaluation of the Segment Outliers and Unusual Events Detector (SOUE-Detector) method, which aims to detect segment outliers and unusual events from data streams generated from wireless sensor networks (using data from Case Study #2).
Chapter 6 presents the design, implementation and evaluation of Semantic Fire Weather Index (SFWI) approach to estimating fire weather indices via semantic reasoning over wireless sensor network data streams (also using data from Case Study #2).

Chapter 7 summarizes the outcomes of this thesis by identifying the main contributions, achievements relative to objectives, limitations, possible improvements for future research, and overall conclusions.
Chapter 2: Literature Review

2.1 Introduction

This chapter provides an overview of past research that has focussed on similar or related work to the four key contributions of this thesis, and outlines how these contributions aim to improve or extend on the current state of the art. As such, this chapter is structured around four primary topics:

- **Automatic semantic tagging of 3D accelerometry data streams.** Section 2.2 reviews previous work on behaviour identification approaches for both humans and animals (including captured and free-ranging animals) that involve the application of machine learning to accelerometry data. It also outlines how the research described in Chapter 3 builds on relevant past research efforts;

- **Graph-based learning methods.** Section 2.3 describes previous research and applications using Graph-based learning techniques, and outlines how this thesis plans to extend previous approaches, and compare and evaluate the results in the context of both semantic tagging of 3D accelerometry data as well as general image classification;

- **Quality control for wireless sensor network data streams.** Section 2.4 reviews the existing outlier approaches for tagging and filtering outlying data detected within wireless sensor network data streams. This section also describes major challenges and open issues associated with designing effective and efficient outlier detection algorithms for environmental sensor data streams and the proposed approach that is implemented and evaluated in Chapter 5;

- **Reasoning across multiple wireless sensor network data streams.** A comprehensive overview of previous research on rule-based reasoning for wireless sensor networks is provided in Section 2.5. In addition, the innovative aspects of the proposed approach are highlighted.

2.2 Supervised-based Automatic Semantic Annotation of Animal Sensor Data Streams

Accelerometers are being used in many fields to understand the movement and behaviour of humans [65-68] and various animals, including productive stock [1, 2, 6, 69-72] and free-ranging animals [19, 21].

To date, the majority of research on automatic recognition tools for accelerometry data has focussed on analysing accelerometry data collected from humans. For instance, an automatic activity recognition approach [67] was presented to recognize human activities (i.e., lying, standing, walking
and running) using augmented autoregressive model coefficients and artificial neural nets. An accelerometer sensor-based approach [68] was proposed to identify high-level (i.e., static, transition or dynamics) activities using means of statistical signal features and artificial-neural nets, and low-level activities (e.g., lying, walking, standing, and running) using autoregressive modelling. Another approach [66] using a hierarchical recognition model was proposed to identify six daily physical activities: resting, walking, walk-upstairs, walk-downstairs, running, and cycling. The average accuracy of detection is about 95%. Most of the prior work focusses on analysing high-frequency, high volume, 3D accelerometry data streams for humans and overcoming the associated computational challenges. Compared with previous research (for example by Khan et al. [66]), this thesis focusses on tagging behaviours from low-frequency 3D accelerometry data streams collected from animals. Smart sampling approaches are adopted to minimize the size of the data and reduce the computational complexity without losing the significant statistical properties of the data. Moreover, this thesis focusses on services that enable ecologists (with little knowledge of machine learning techniques) to build classification models using data and video collected from domestic animals. The system then enables them to apply these models to accurately classify data collected from free-ranging species, where there is no ground truth (observation data) available.

There has been previous research into using accelerometers to analyse animal behaviour to provide a measure of animal comfort [73, 74] and animal welfare [73, 75]. However, distinguishing specific behavioural modes from data collected by animal-borne tri-axial accelerometers can be a time-consuming and subjective process. Different types of accelerometers, including mono-accelerometers, bi-axial accelerometers, and tri-axial accelerometers, have been used in the past to monitor the movement patterns of production livestock. For example, the movement patterns of sows, including feeding, rooting, walking, standing, sitting, stepping, lying sternally, lying laterally, lying ventrally, high active, and medium active have been extensively studied [2, 69, 71, 76, 77]. In particular, an approach was proposed to automatically classify the five types of activities of group-housed sows, including feeding, walking, rooting, lying laterally and lying sternally by applying Multi-Process Kalman Filter based classification models over four time-series acceleration data streams (the three-dimensional axes and the length of the acceleration vector) [2].

A further approach was designed to automatically classify 19 farrowing house sows’ activities as: high active, medium active, lying laterally on one side, lying laterally on the other side, and lying sternally. This study applied the multi-process Kalman Filter to build the classifiers [69]. Compared with the first study, this study further analysed the impact of the bedding materials on sows’ activities.

Ringgenberg et. al. [71] proposed an approach to detect sows’ standing, sitting, lying ventrally and lying laterally postures by attaching two accelerometers to each of 23 sows (one accelerometer
fastened to a hind leg and the other to the back of the sow). In this study, video observations were also recorded to calculate the average time the sow spent on each activity [71].

A related study by Moreau et al. [6] examined the activities of goats, specifically, walking, resting and eating (grazing with head down and browsing with head up), by merging moving averages with threshold values. In this study, Moreau et al. proposed an automatic approach to record and classify goats’ grazing behaviour by mounting data loggers onto a harness check belt or neck collar. The classification approach was programmed by calculating moving averages for the transformed accelerometer data, the value of the impulse and the value of the amplitude, and by selecting threshold values to distinguish resting from eating and eating from walking.

Similar studies investigated the behaviours of cows, including standing, lying, ruminating, feeding, walking normally, lame walking, and walking with gait changes [1, 20, 70, 72, 73, 78-80]. For example, Cangar et al. proposed an automatic real-time monitoring technique to identify the locomotion and posturing behaviour of pregnant cows. Specifically, they aimed to classify specific behaviours such as standing or lying (including incidences of motion during lying), and eating or drinking [81]. Additionally, Nielsen et al. [1] proposed an algorithm for predicting walking and standing activities based on a moving average of the output from a 3D accelerometer.

Compared with the production livestock studies, less attention has been paid to the assessment of behavioural modes in free-ranging animals [19, 21]. Shepard et al. analysed the behaviour of captive and free-living animals, including Eurasian badgers, Imperial cormorant, leatherback turtles, lemon sharks, Magellanic penguins, cheetahs, coypus, Brazilian tapirs, giant ant-eaters, guanacos, hairy armadillos, mouflon, and llamas, by taking a running mean of total acceleration values over two seconds using an Origin Pro accelerometer and Microsoft Excel [19]. Analysis of griffon vulture behaviour [21] was conducted using five machine learning algorithms, including linear discriminant analysis, support vector machines, classification and regression trees, random forests and artificial neural networks. These algorithms are commonly used for pattern recognition and classification tasks of complex data [82].

However, the process of analysing tri-axial accelerometer data streams, to date, remains in its infancy with much of the analysis and pattern identification undertaken manually and subjectively in biological research areas, especially for studying free-ranging wild animal behaviours [19]. Very little research has been undertaken into optimizing animal behaviour classification algorithms for free-ranging animals using 3D accelerometry data sets. No previous efforts have investigated crowdsourcing or expert-sourcing approaches for annotating the training corporuses required for the machine-learning step. None of the previous approaches attempted to develop a modular, extensible,
interoperable framework based on Semantic Web standards, to support the sharing and re-use of accelerometry data and resulting classification models across studies and/or species. None of the previous approaches attempted to generate better, re-usable cross-species classification models – by enabling researchers to collaborate and share their accelerometry datasets and expertise. None of the previous approaches aimed to provide a single Web Portal that provides access to: a common repository for the accelerometry datasets; common annotation services; common machine learning approaches; and shareable classification models; based on common data formats, metadata standards and ontologies.

To summarize, Section 2.2 firstly describes the previous research on automatic recognition tools for analysing human accelerometry data streams. Next, it outlines the existing behaviour identification approaches for distinguishing specific behavioural modes from accelerometry data streams collected from productive stock including sows, goats and cows. Finally, it introduces the previous research that has focussed on the assessment of behavioural modes in free-ranging animals, and identifies the open issues and novel aspects associated with the processing of analysing tri-axial accelerometry data streams from free-ranging animals, that is the focus of Chapter 3.

2.3 Optimal Graph-based Learning for Automatic Annotation of Animal Sensor Data Streams

To date, supervised machine learning has been the predominant approach in both the human activity recognition [65-68] and animal behaviour recognition [2, 69, 71, 76, 77] research domains. However, preparing labelled data for supervised learning is expensive, even if annotated via expert-sourcing (Chapter 3), because the volume of unlabelled data is becoming unmanageable [83]. Therefore, some recent research studies have proposed the adoption of semi-supervised learning (SSL) algorithms. Instead of learning from a large amount of labelled data, SSL algorithms learn from both labelled and unlabelled data [83]. In addition, some previous research efforts have proved that graphs provide a natural way to represent data in a variety of domains [83]. Recently graph-based SSL algorithms have been successfully used to extract class-instance pairs from large labelled and unlabelled data sets for speech classification [84, 85], as well as for activity behaviour recognition in humans [84, 85].

Specifically, graph-based learning is an efficient approach for modelling data in various machine learning schemes, i.e., unsupervised learning [30-32] supervised learning [33] and semi-supervised learning [31, 32, 34, 35]. An important advantage of working with a graph structure is its ability to naturally incorporate diverse types of information and measurements, such as the relationships between features identified in unlabelled data, labelled data or both labelled and unlabelled data.
Among these graph-based learning schemes, semi-supervised learning (SSL), i.e., learning from both labelled and unlabelled data, has been widely studied and applied to many challenging tasks [31, 33, 36] such as image classification, image ranking and image annotation. By exploiting the large volumes of unlabelled data with reasonable assumptions, SSL can reduce the need for expensive labelled data and thus achieve promising results especially for noisy labels [37]. The harmonic function approach [86] and Local and Global Consistency (LGC) [87] are two representative graph-based SSL methods. The harmonic function approach emphasizes the harmonic nature of the diffusive function and LGC considers the spread of label information in an iterative way. While these two methods are transductive, manifold regularization (MR) [11, 88] is inductive. MR extends regression and SVM respectively to the semi-supervised learning methods: Laplacian Regularized Least Squares (LapRLS) and Laplacian Support Vector Machines (LapSVM); by adding a geometrically based regularization term [89].

Following the development of SSL, many applications and further refinements [37, 90-92] have been proposed. Zhang et al. extended the Linear Discriminant Analysis (LDA) to semi-supervised discriminant analysis [93] and also proposed the semi-supervised distance metric learning method [91]. Tang et al. addressed the noisy label issue for the task of semi-supervised image labelling [37], and Song et al. utilized weak-label information for cross-media retrieval [92].

Since an informative graph is critical for graph-based algorithms, its construction has also been extensively studied [31, 94, 95]. The most popular way to construct a graph is the K-nearest-neighbor (or \( \varepsilon \)-range-neighbor) method, where, for each data point, the samples are connected with its K-nearest-neighbors (or \( \varepsilon \)-range-neighbor). Then the Gaussian-kernel can be used to quantify the graphs. However, the tuning of \( \sigma \) in the Gaussian-kernel approach is empirical [94]. Recently, it has become more popular to learn a graph, using either the pairwise distance based method or the reconstruction coefficients based method. The first method is based on the Euclidean distance between two data points and the assumption that close data points should have a high similarity and vice versa. The second method assumes that each data point can be reconstructed as a linear combination of the other data points. These two methods demonstrate different strengths and weaknesses depending on the application. However, most of these graphs are constructed using single information cue (e.g., visual feature, labels), and an optimal graph that can utilize multiple cues has rarely been addressed.

Graph-construction using the pairwise distance based method or the reconstruction coefficients based method, also assumes that the data are clean, i.e., the data points are strictly sampled from the subspaces, and several approaches are able to recover the subspace structures [96]. However, in real applications, the data set may lie in the union of multiple subspaces or contain noise and outliers [95].
As a result, inter-class data points may be connected with very high weights. Hence, eliminating the effects of errors becomes a major challenge. To address these problems, several algorithms have been proposed, e.g., Locally Linear Manifold Clustering (LLMC) [97], Agglomerative Lossy Compression (ALC) [98], Sparse Subspace Clustering (SSC) [99], L1-graph [31, 100], Low Rank Representation (LRR) [101, 102], Latent Low Rank Representation (LatLRR) [103], Fixed Rank Representation (FRR) [104], L2Graph [95]. In [105], Vidal provided a comprehensive survey of these algorithms in the context of subspace clustering.

Of the above methods, SSC [99] and L1-graph [31] obtain a sparse similarity graph from the sparsest coefficients. One of the main differences between these techniques is that [99] formulates the noise and outliers in the objective function and provides more theoretical analysis, whereas [31] derives a series of algorithms upon the L1-graph for various tasks. The popular LRR model [101, 102] and its extensions [103, 104] are very similar to SSC, except that it aims to obtain a similarity graph from the lowest-rank representation rather than the sparsest one. Both $\ell_1$ and rank-minimization-based methods can automatically select the neighbors for each data point due to the sparse solution, and have achieved impressive results in numerous applications. However, their computational complexity is proportional to the cube of the problem size. Moreover, SSC requires that the corruption over each data point has a sparse structure, and LRR assumes that only a small portion of the data are contaminated, otherwise the performance will be degraded. In fact, these two problems are mainly caused by the error-handling strategy that has been adopted, i.e., removing the errors from the data set to obtain a clean dictionary over which each sample is encoded [95].

The research approach proposed in Chapter 4 is novel because a new graph-based learning, based on semi-supervised machine learning (SSL), is proposed and applied to 3D accelerometry data streams to perform animal activity recognition. Moreover, it extends the state of the art by proposing an optimal graph that utilizes multiple cues and also addresses out-of-sample and noisy data issues.

To summarize, Section 2.3 reviews the existing graph-based learning schemes for semi-supervised machine learning, and identifies issues associated with designing effective and novel Optimal Graph Learning (OGL) algorithms for classification tasks. It proposes a novel approach that has been applied and evaluated in the context of the animal behaviour recognition task (which is described in Chapter 4).

2.4 Outlier Detection for Wireless Sensor Network Data Streams

A significant problem associated with knowledge extraction from data streams generated from wireless sensor networks is data quality due to limited battery power, limited memory, limited
computational capacity and limited communication bandwidth of the wireless links that connect sensor nodes and sensors [44]. Hence there is an urgent need for services that are capable of automatically or semi-automatically detecting, tagging and filtering erroneous segments or outliers from sensor data streams.

Early research efforts aimed at detecting outliers from sensor data streams have adopted the following approaches: statistical-based approaches [106, 107]; nearest neighbour-based approaches [108]; clustering-based approaches [109]; classification-based approaches [110-112]; and Semantic Web based approaches [113]. However, these previous approaches have a number of shortcomings when applied to the detection of genuine outliers.

Firstly, the majority of previous work assumes that the sensor data is univariate and, thus, fails to take into account multivariate data [114].

Secondly, little effort has focussed on handling the dynamic nature, variability and heterogeneity of sensors (the devices that detect or measure a physical property), sensor nodes (the platforms on which multiple sensors can be attached) and sensor networks. For example, in April 2008, scientists from the Springbrook Wireless Sensor Network [9] installed nine sensor nodes with sensors measuring leaf wetness, soil moisture, air pressure, air temperature, relative humidity and wind (direction and speed). In April 2009, more sensor nodes were installed and new sensors including rainfall and light were added. In February 2011, an additional 125 sensor nodes were added and a range of new sensors started collecting information on tree growth, carbon dioxide concentrations, cloud cover and fog density. At the same time, the wind direction sensors were removed. Sensor nodes were moved between different locations at different points in time, and were re-configured with different numbers and types of sensors. Scientists wanting to detect and filter outliers from sensor network streams, require a method that keeps track of and updates the configuration data of the wireless sensor network as it changes over time.

A third shortcoming in the detection of outliers is the use of supervised machine learning algorithms such as Support Vector Machines [112, 115] and Bayesian Networks[109]. Classification of abnormal data is difficult to achieve because there is no prior knowledge available via a training dataset. This thesis research overcomes this limitation by designing new algorithms that take into account domain expert knowledge about correlations between different properties being monitored. Such correlations among sensor properties have been ignored by most previous studies, but they are extremely valuable in detecting unusual events and can be captured through domain expert knowledge. Hence, this thesis proposes a method that involves capturing and exploiting domain expert knowledge about correlations between sensor properties (e.g., between temperature, humidity, wind speed trends).
Lastly and most importantly, most previous studies assume by default that any outliers are errors [114]. They do not attempt to distinguish between the erroneous outliers and genuine outliers associated with unusual events [114, 116].

Distinguishing between outliers that are errors and outliers that are unusual events is critical to advance the development and adoption of wireless sensor networks and to provide accurate and reliable data to underpin decision support systems. In this study, an erroneous outlier, also called a true error, is defined as a segment of the sensor data stream that refers to noise-related measurements or data generated by a faulty sensor. Such a segment deviates from the usual sensor data streams and is likely to be spatially and temporally unrelated to neighbouring data streams. An unusual event is defined as a particular phenomenon that simultaneously changes the patterns of multiple types of sensor data streams so that they deviate from the normal patterns of the data streams. In addition, the sensor data streams associated with unusual events are likely to be spatially or temporally correlated to neighbouring data streams. Thus, this study focuses on distinguishing between erroneous outliers (true errors) and unusual events by making use of spatio-temporal and other types of correlations between sensor properties.

Semantic Web based approaches, including RDF, ontologies and inferencing rules, have been applied previously to reason about anomalous sensor data [113]. Specifically, in this paper, Calder et al. provided a framework for validating scientists’ or decision makers’ hypotheses about anomalous sensor data. However, the identification of anomalous sensor data depends on scientists’ or decision makers’ prior definitions of what constitutes anomalous data (e.g., valid ranges). Compared with previous work, this study does not record specific definitions about outliers or events from domain experts; rather, it documents specific correlations between sensor properties to improve the detection of genuine outliers associated with unusual events. In addition, this study provides intuitive user interfaces to enable domain scientists to express their knowledge about relationships or correlations between multiple sensor data streams measuring different properties. The assumption is that unusual events can be identified by unusual patterns across multiple sensor properties if there exists an historical correlation between those properties. This study also provides a visualization interface that combines a map interface, multiple timelines and colour coding to enable users quickly and easily to view, verify and filter out or further investigate data segments that have been identified as “outlying”.

To summarize, Section 2.4 reviews the existing outlier detection approaches for wireless sensor network data streams, and the major challenges and open issues associated with designing effective and efficient outlier detection algorithms for environmental sensor data streams. It also proposes a novel approach to outlier detection that relies on the capture and exploitation of domain-expert knowledge that defines correlations between multiple sensor data streams (described in Chapter 5).
2.5 Reasoning across Multiple Wireless Sensor Network Data Streams

To date, a number of previous research projects have focused on reasoning across multiple wireless sensor network data streams to infer complex events by integrating Semantic Web technologies [11, 12, 40, 50, 113, 117-120].

Sheth et al. [11] proposed the Semantic Sensor Web (SSW) to address the annotation of sensor data with semantic spatial, temporal, and thematic metadata to increase interoperability as well as provide contextual information essential for situational knowledge. [11] also demonstrated the application of rules to derive higher level knowledge (e.g., “Potentially icy” and “blizzard” conditions) from semantically annotated sensor data.

To enable the development of SSW and Linked Sensor Data [43], Janowicz and Compton [120] designed a generic Stimulus-Sensor-Observation ontology design pattern with core concepts (e.g., observation, sensor, stimulus, procedure, results, observedProperty and FeatureofInterest) and relations (e.g., involves, satisfies, detects, implements, produces, isProxyFor and isPropertyof) for describing observation-based data on the Semantic Web.

Compton et al. [50] later proposed the Semantic Sensor Network (SSN) ontology with 41 concepts and 39 properties (e.g., sensors, properties and features, observations, and platforms) for describing sensors, sensing, the measurement capabilities of sensors, the observations and deployments. In addition, Pfisterer et al. [118] introduced a service infrastructure that integrates vocabularies, semantic entities, and a semi-automatic generation of semantic sensor descriptions, with search services to ease the adoption of the Semantic Web of Things for end users and developers. (The Semantic Web of Things is an emerging vision in which Semantic Web technologies (such as ontologies, semantic annotation, Linked Data and semantic Web services) are applied to heterogeneous devices (sensors and actuators) connected to the Internet (i.e., the Internet of Things) to enable semantic interoperability).

In [117] Compton et al. proposed the Sensor ontology for specifying sensors and expressing complex compositions and fine details of the function and results of sensors and processes using SPARQL-DL [121].

In [113], Calder et al. proposed an approach for inferring anomalous sensor data using machine reasoning. A Coastal Environmental Sensor Networks (CESN) ontology, a Knowledge Base for sensor networks that observe coastal ecosystems and machine reasoning services were developed to deduce specific ecosystem events.
A survey by Wei and Barnaghi [40] discussed the state of the art in the design and development of the Semantic Sensor Web, and demonstrated how rule-based reasoning performed over sensor observations can provide an effective approach for dealing with missing data or uncertainty by adopting domain ontologies. The above mentioned approaches focussed on demonstrating the capability of rule-based semantic reasoning.

In [12], Thirunarayan et al. demonstrated how to enhance raw sensor data with spatial, temporal and thematic annotations to enable the detection of inconsistent sensor data. This research effort also formalized the Weather domain and developed a meta-interpreter in Prolog to explain Weather data. Cabral et al. [122] demonstrated the application of semantic reasoning to multiple sensor data streams (e.g., air temperature, wind speed, and leaf wetness) to predict the risk of botrytis (grape rot) within vineyards. Cabral et al. proposed a method for selecting and ranking sensors based on the requirements associated with environmental variables that are input to predictive analytical models. This work mainly focused on managing the difficulties associated with describing sensors and evaluating them in a dynamic environment.

Within this thesis, a rule-based reasoning approach is proposed to infer fire weather indices from real wireless sensor data streams collected from the Springbrook National Park, in South East Queensland. The traditional and predominant approach to detecting and forecasting wild fires is to analyse satellite data. However, it has been proven that using wireless sensor networks to detect and forecast forest fires provides more timely and spatially accurate data than using traditional satellite telemetered data [123]. Hence a number of recent efforts have focussed on monitoring forest fires using wireless sensor networks [124-126]. Generally, these past research efforts have focussed on two levels: the wireless sensor network level (focussing on hardware and communications), and the wireless sensor network data analysis level (focussing on the data).

At the wireless sensor network level, current studies are focussing on improving the wireless sensor network configurations, deployment, communication, and hardware devices to enable more efficient fire detection and monitoring. For example, Aslan et al. [125] proposed a framework consisting of four main components: an approach for deploying sensor nodes; an architecture for the sensor network for fire detection; an intra-cluster communication protocol, and an inter-cluster communication protocol. The aim is to improve the energy efficiency, support early detection and accurate localization, enable forecast capability, and adapt sensor networks to harsh environments. Fernández-Berni et al. [127] investigated early detection of forest fires using a vision-enabled wireless sensor network. Their work includes a vision algorithm for the detection of smoke and a
low-power smart imager to stream images. They integrated these two components to generate a prototype vision-enabled sensor network node.

At the wireless sensor network data analysis level, researchers are mainly investigating the best sensor combinations (e.g., light, air temperature, air pressure, wind speed, wind direction, soil moisture, leaf wetness, relative humidity, rainfall and smoke) to detect forest fires. With the aim of improving fire hazard detection and monitoring, they also developed more advanced algorithms (e.g., clustering, summaries, threshold, statistical modelling, neural network, threshold values, Dempster-Shafer theory based algorithm) [123, 125, 126, 128-130]. Specifically, Diaz-Ramirez et al. [126] proposed two information fusion based algorithms for detecting forest fires using wireless sensor networks. The first algorithm used a threshold based method which takes temperature, humidity and light as input, while the second algorithm was a Dempster-Shafer theory based algorithm, which only took temperature and humidity as input.

The FireWatch [124] system was proposed to overcome the limitations of the traditional satellite and camera-based systems by integrating wireless sensor network technologies, computer-supported cooperation work, and a geographic information system (GIS). This system was designed to detect forest fires using wireless sensor networks but not to support fire hazard predictions.

At present, only a few approaches have directly focussed on calculating fire weather indices from the wireless sensor network data streams [131]. For example, Sabit et al. [131] have presented approaches to generate micro-scale estimates of the Fire Weather Index from wireless sensor network data streams, but they do not use Semantic Web technologies.

The Linked Stream Middleware (LSM) system developed by Le-Phuoc et al. [132] has been proposed to integrate time-dependent data with other Linked Data Resources. It supports the publishing of real-time data collections using a cloud-based infrastructure, the enrichment of sensor sources and sensor data streams with semantic descriptions, and provides a SPARQL endpoint for querying unified Linked Stream Data and Linked Data. LSM provides a generic middleware framework for managing and querying linked sensor streams, but it does not provide any rule-based reasoning (such as the reasoning services applied in this thesis to infer higher level Fire Weather Indices).

2.6 Summary

This chapter summarized previous relevant research and approaches that have been explored to enable the semantic annotation and reasoning of sensor data streams. More specifically, it focusses on related efforts in the fields of: animal behaviour recognition from accelerometry data streams; the application of optimal graph learning to automatic classification tasks; automatic outlier and event detection for
sensor network data streams, and semantic reasoning across multiple sensor network data streams for environmental applications. Moreover, this chapter has identified how the research described in the following Chapters 3-6, differs from previous efforts and aims to extend the current state of the art and overcome existing limitations.

The next chapter (Chapter 3) focuses on the first topic, which is automatic semantic tagging and analysis of 3D accelerometry data (focusing on the recognition of animal behaviour). More specifically, it describes how automatic recognition of tri-axial accelerometer data streams can be supported by integrating semantic annotation and visualization services with Support Vector Machine (SVM) techniques. It also explains how a behavioural classification module can be built using data from domestic animals - and successfully applied to the automatic classification of accelerometry data streams collected from similar or free-ranging species.
Chapter 3: Semantic Annotation and Activity Recognition for Animal Sensor Data Streams

3.1 Overview

Increasingly, animal biologists are taking advantage of low cost micro-sensor technology, by deploying accelerometers to monitor the behaviour and movement of a broad range of species. The result is an avalanche of complex tri-axial accelerometer data streams that capture observations and measurements of a wide range of animal body motion and posture parameters. Analysis of these parameters enables the identification of specific animal behaviours. However, the analysis process is immature with much of the activity identification being undertaken manually and subjectively. Consequently, there is an urgent need for the development of new tools to streamline the management, analysis, indexing, querying and visualization of such data.

This chapter presents a Semantic Annotation and Activity Recognition (SAAR) approach, which integrates semantic annotation with Support Vector Machine (SVM) techniques to automatically identify animal behaviours from 3D accelerometry data streams. This approach enables biologists to visualize and correlate 3D accelerometer data streams with associated video streams. It also enables domain experts to accurately annotate or tag segments of tri-axial accelerometer data streams, with standardized terms extracted from an activity ontology. These annotated data streams can then be used to dynamically train a hierarchical SVM activity classification model, which can be applied to new accelerometer data streams to automatically recognize specific activities. This chapter describes the design and functional details of the SAAR and the results of the evaluation experiments that assess the performance, usability and efficiency of the proposed approach. The evaluation results indicate that the SAAR enables ecologists with little knowledge of machine learning techniques to collaboratively build classification models with high levels of accuracy, sensitivity, precision and specificity. The results also indicate that the SAAR is able to use surrogate test individuals to qualify and quantify the association between individual behavioural modes and tri-axial accelerometry data streams.

3.2 Activity Recognition using Support Vector Machines

SVMs (Support Vector Machines) are well established as a successful modelling and prediction tool for both pattern classification and regression tasks. They are linear classifiers based on statistical learning theory and the concept of the maximum margin hyper-plane. In previous species activity identification studies [20, 133, 134], SVMs demonstrate relatively good performance when applied
to the classification of tri-axial accelerometer data streams from humans and cows. For SAAR, the LIBSVM library [61] was chosen because it is open source, written in Java and is simple to download and use. More specifically, it uses the C-SVC (C-support Vector Classification) algorithm [135] from the LIBSVM library because it is the simplest SVM approach.

The proposed activity recognition service is designed to perform on two levels: high level and low-level recognition. The high level recognition service identifies active and inactive activities, while the low level recognition recognizes specific activities which are sub-classes of the active and the inactive activity classes (for example, running, walking, feeding, sleeping and lying). In order to use the C-SVC algorithms to recognize tri-axial accelerometer data stream patterns automatically, application-dependent features have to be extracted.

In this study, features were extracted using a window size of 3 seconds with an overlap of 1 second (2 sampling points for a 1 Hz sampling rate) between consecutive windows. Specifically, a window contains 4 sampling points. There are three reasons for selecting this window length and overlap. Firstly, feature extraction on sliding windows with 50% overlap has been demonstrated to achieve accurate results in previous research efforts [136-139]. Secondly, it has been shown that a window of 2 seconds can capture activities [139]; hence, a window of 3 seconds with a 1 second overlap is sufficient to capture activities. Thirdly, the most efficient algorithm for calculating the Fast Fourier Transform (FFT) usually operates with a window length that is a power of two.

During high-level recognition, the proposed approach extracts the following features including standard deviation vector, signal magnitude area vector and waveform length vector. They are expressed respectively as follows:

- **Standard deviation (SD):** The standard deviations ($SD_x$, $SD_y$ and $SD_z$) measure how spread out the signal is within x-axis, y-axis and z-axis, respectively.

$$SD_x = \sqrt{\frac{1}{N-I}\sum_{i=1}^{N}(x_i - \frac{1}{N}\sum_{k=1}^{N}x_k)^2}$$

$$SD_y = \sqrt{\frac{1}{N-I}\sum_{i=1}^{N}(y_i - \frac{1}{N}\sum_{k=1}^{N}y_k)^2}$$

$$SD_z = \sqrt{\frac{1}{N-I}\sum_{i=1}^{N}(z_i - \frac{1}{N}\sum_{k=1}^{N}z_k)^2}$$

(3.1)
where $x_i$ and $x_j$ are the $i$-th and the $k$-th accelerometer values on the x-axis, where $y_i$ and $y_k$ are the $i$-th and the $k$-th accelerometer values on the y-axis, where $z_i$ and $z_k$ are the $i$-th and the $k$-th accelerometer values on the z-axis, and $N$ is the window size.

- **Signal magnitude area (SMA):** The signal magnitude area is found to be a suitable measurement of the degree of movement intensity that can distinguish between active and inactive activities using tri-axial accelerometer data [67].

$$SMA = \frac{1}{N} \left( \sum_{i=1}^{N} |x_i| + \sum_{i=1}^{N} |y_i| + \sum_{i=1}^{N} |z_i| \right) \quad (3.2)$$

- **Waveform Length (WL):** The WL is the cumulative length of the waveform amplitude, frequency and duration all within a signal window. In other words, it measures the total amount of signal vibration variance through three dimensions.

$$WL = \frac{1}{N-1} \left( \sum_{i=1}^{N-1} |x_{i+1} - x_i| + \sum_{i=1}^{N-1} |y_{i+1} - y_i| + \sum_{i=1}^{N-1} |z_{i+1} - z_i| \right) \quad (3.3)$$

During low-level recognition, the proposed approach extracts spatial-domain features (standard deviation, signal magnitude area, waveform length). In addition, this approach extracts frequency-domain features and an inheritance parameter.

The discrete Fourier transform (DFT), a transform for Fourier analysis of finite-domain discrete-time signals, is widely employed in signal processing to produce frequency information contained in a sampled signal [140]. A fast Fourier transform (FFT) is an efficient algorithm to compute DFT and it produces exactly same results as DFT [141]. Given a set of real or complex numbers $x_0, \cdots, x_{N-1}$, the DFT transforms them into the sequence of $N$ complex numbers $X_0, \cdots, X_{N-1}$. Those complex numbers represent the magnitude and phase information about the transformed sequence. This study takes the power of the magnitude of the complex FFT output as the component of the frequency-domain features.
Figure 3.1: FFT-based feature extraction algorithm

Figure 3.1 illustrates how FFT transform is used to compute frequency-domain features during low-level activity recognition. The Inheritance parameter (IP) measures whether a subclass was originally inherited from a parent class. To compute the IP value, high-level activity recognition is employed to recognize two classes: active activity and inactive activity. The value of IP is 1 if the classification result belongs to the active activity class, and -1 if it belongs to the inactive activity class.

### 3.3 Case Study

The challenge for many ecologists is to understand the movement and behaviour of animals “in the wild”. Researchers are currently using accelerometers to measure the activity levels and movement of many wild animals (including crocodiles [14], bears [142] and badgers [19]) to assist with their management and conservation. In Australia, researchers are investigating the behaviour and movement of wild dogs and dingoes in order to develop appropriate management strategies [143, 143].
The difficulty with analysing tri-axial accelerometer data from wild animals is that there is little or no observational data or video that provides the evidence for training an automatic activity recognition model. One of the aims of thesis work described in this chapter is to develop a model for domestic quadruped mammals (i.e., domestic dogs) using associated video as verification and to determine whether this model can be used to accurately recognize activities of other similar-sized quadruped mammals (e.g., badgers) or related species in the wild (e.g., dingoes), for which there is no corresponding observational video.

### 3.3.1 Data Collection

A tri-axial accelerometer (G6A), produced by Cefas Technology Limited (CTL), was used to collect dog, badger, dingo, cheetah, tiger, wombat, kangaroo and echidna data sets. This data logger is 40 mm x 28 mm x 16.3 mm, 16MB memory, 7.3g weight in air and 2.3g weight in seawater. It supports a wide range of sampling rates from 1Hz up to 30Hz. In this study, a sampling frequency of 1Hz was selected, as this is sufficient to detect changes in behaviour and can monitor animals for long periods of time without producing large volumes of redundant data.

The first phase involved collecting data from domestic dogs. The accelerometer device (G6A) was attached to the back of each dog’s neck via its collar with the X-axis pointing backwards, the Y-axis pointing left, and the Z-axis pointing upward. In order to evaluate the performance of the proposed approach, six dogs of different breeds and ages were observed, including: a four year old Border Collie, 15 kg weight, 52 cm height; a one year old Dachshund, 8.9 kg weight, 20 cm height; an eight year old Cocker Spaniel, 14 kg weight, 35 cm height; a five year old German Short-Haired Pointer, 25.8kg weight, 63 cm height; a ten year old Staffordshire Terrier-Labrador cross, 21 kg weight, 55 cm height; and a five year old Cavalier King Charles Spaniel, 7.5 kg weight, 30 cm height. During the training data collection stage, each dog was directed by its owner to perform two minutes walking, two minutes running, two minutes standing, two minutes sitting and two minutes lying. In addition, the King Charles Spaniel spent one minute foraging and one minute climbing (front paws raised to reach a treat, whilst the owner walked backwards). During the test data collection stage, each dog was directed by its owner to randomly perform the same set of activities listed above, over a period of 10 minutes. During the entire data collection phase, a camera simultaneously recorded video of the animal, which provides the ground truth for the evaluation phase.

Phase 2 involved collecting data from Eurasian badgers studies undertaken at West Hatch RSPCA Centre, Somerset, UK. During these studies, five Eurasian badgers were equipped with tri-axial accelerometers that were attached to leather collars fastened round the badgers’ necks with the X axis pointing backwards, the Y axis pointing left and the Z axis pointing upward [19]. Camera traps were
also set up to verify activities, although large periods of activity were outside the cameras’ fields of view. Where no verification by video was possible, manual annotations were made based on prior knowledge and the principals set out by Shepard et al., [19]. Six activities were annotated: walking, running, climbing, foraging, standing and lying.

During phase 3, data was collected from a range of other species at the Australia Zoo, including a domestic dingo; a Bengal tiger; an African cheetah; a hairy-nosed wombat; an eastern grey kangaroo and a short-beaked echidna. For each test subject, the accelerometer was positioned on the dorsal surface of the neck in the orientation of: X, anterior-posterior; Y, lateral axis; Z, dorsal ventral. Five activities were annotated: running, walking, standing, sitting, and lying. Table 3.1 shows how the accelerometer was attached to each test subject and Table 3.2 shows each animals’ characteristics.

Table 3.1: The subject animals with attached accelerometers

<table>
<thead>
<tr>
<th>Domestic dogs</th>
<th>Eurasian Badgers</th>
<th>Australian Dingo</th>
<th>African Cheetah</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic Dog</td>
<td>Eurasian Badger</td>
<td>Australian Dingo</td>
<td>African Cheetah</td>
</tr>
<tr>
<td>Bengal Tiger</td>
<td>Hairy-nosed Wombat</td>
<td>Eastern Grey Kangaroo</td>
<td>Short-beaked Echidna</td>
</tr>
</tbody>
</table>

Table 3.2: The characteristics for each animal used in the study (BS: Body Mass; SL: Spine Length; SH: Spine Height above the ground)

<table>
<thead>
<tr>
<th></th>
<th>BM (kg)</th>
<th>SL (cm)</th>
<th>SH (cm)</th>
<th>SL:SH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic Dog</td>
<td>14.0</td>
<td>54</td>
<td>24</td>
<td>2.25</td>
</tr>
<tr>
<td>Eurasian Badger</td>
<td>25</td>
<td>48</td>
<td>12</td>
<td>4.0</td>
</tr>
<tr>
<td>Australian Dingo</td>
<td>18.0</td>
<td>58</td>
<td>25</td>
<td>2.32</td>
</tr>
<tr>
<td>African Cheetah</td>
<td>43.0</td>
<td>108</td>
<td>43</td>
<td>2.51</td>
</tr>
<tr>
<td>Bengal Tiger</td>
<td>91.2</td>
<td>179</td>
<td>57</td>
<td>3.14</td>
</tr>
<tr>
<td>Hairy-nosed Wombat</td>
<td>23.0</td>
<td>63</td>
<td>12</td>
<td>5.25</td>
</tr>
<tr>
<td>Eastern Grey Kangaroo</td>
<td>29.5</td>
<td>113</td>
<td>15</td>
<td>7.53</td>
</tr>
<tr>
<td>Short-beaked Echidna</td>
<td>4.2</td>
<td>43</td>
<td>6</td>
<td>7.16</td>
</tr>
</tbody>
</table>
3.3.2 Implementation

The methodology comprised the eight stages described below:

1. A Web interface was developed that enables datasets (tri-axial accelerometer data in CSV format and corresponding videos in OGV format) to be uploaded to a file store on the system’s server and described using simple metadata including: Creator, dateCaptured, Species, AnimalID, Location, Coordinates, and Description. The files and the metadata are stored on the server.

2. Users can search, browse, retrieve and open specific datasets and visualize both the tri-axial accelerometer data (and associated video, if available) through a graphical user interface that comprises two panels (Plot and Video), juxtaposed one above the other – that display the tri-axial movement data streams and the video stream respectively. Simple alignment tools enable users to precisely synchronize the data streams and the corresponding video stream.

3. An ontology-based annotation service enables domain experts to tag tri-axial accelerometer data streams manually using the combined Plot and Video user interface. Tags are drawn from predefined ontologies that define the terms describing activities of interest to the researcher and of relevance to the animal being studied, such as running, walking, standing, sitting, lying. Separate ontologies can be developed for different terrestrial, marine and avian species and the most appropriate ontology can be configured at run-time. For example, the high level AnimalBehaviour Ontology comprises: top-level entity ab:Behaviour; sub-classes of ab:Behaviour are: ab:Active and ab:Inactive; sub-classes of ab:Active are: ab:Running and ab:Walking; sub:classes of ab:Inactive are: ab:Standing; ab:Sitting; ab:Lying. This ontology can be further extended for particular animals/species e.g., badgers.

4. The manually attached tags (and pointers to relevant file segments/time stamps) are stored on an annotation server in the RDF format. Through the annotation interface, users can share their tags with other users, search and retrieve specific annotations and their associated accelerometry data segments e.g. give me all segments in which animal with ID ‘abcd’ is ‘running’.

5. A user then specifies the set of tagged data streams which are to be used as the training data. The system retrieves and aggregates all of the selected segments corresponding to each tag and extracts a set of application-dependent features. The application-dependent features and related labels are then used to interactively train a hierarchical SVM classifier that recognizes both “active” (e.g., running and walking) and “inactive” (e.g., sitting, lying and standing) states as well as more specific sub-class activities.
6. When new tri-axial accelerometer data streams are uploaded, the corresponding application-dependent features are extracted and then input into the trained SVM classifier which automatically annotates the new data streams. The classification results are stored in RDF on the annotation server and displayed via the Web visualization interface for biologists to verify or correct.

7. Finally, statistical analysis tools are also provided that calculate the statistics for each activity for a single animal or a set of animals (including average, minimum, maximum time of occurrence, cumulative time of occurrence in the whole period, total number of occurrences, and the standard deviation of the duration time). These results are presented as a pie chart via the Web interface showing the percentage of time spent on each activity.

8. The system’s usability and efficiency are also evaluated by collecting and analysing users’ feedback and performance metrics.

Figure 3.2 shows the high level architectural components of the SAAR system which combines: Web 2.0 technologies (Java, JavaScript, and JSON) to maximize accessibility and interoperability; with Semantic Web technologies (RDF, SPARQL, OWL ontologies) to maximize knowledge capture, reuse and exchange through standardized vocabularies; and Support Vector Machine (SVM) to provide the machine-learning tools for automated recognition of activities.
A Web-based Plot-Video visualization interface has been developed using a combination of AJAX, Flot (a plotting jQuery library) [59], HTML 5 Video Player library (Video.js) [145] and JavaScript to enable users to interactively visualize both tri-axial accelerometer data alongside simultaneously recorded videos in an interactive plot visualization pane and a video player, respectively.

Using the Plot-Video visualization interface, users can invoke the semantic annotation service by selecting a segment of accelerometer data from the timeline or a segment of video from the video pane, and then attaching an activity class label chosen from a pull-down menu (the values of which are extracted from a pre-defined ontology). The manually created annotation is stored in an RDF triple store. More specifically, the annotation server is implemented using the Apache Tomcat Java server and Sesame 2.6.3, a Java framework for storage and querying of RDF data. Additional annotation functions such as edit, delete, refresh, search and retrieve annotations are also supported.

The activity recognition is implemented using the LIBSVM Java library. At the training stage, users interactively search and retrieve specific annotations via the following search terms: Species, Creator,
AnimalID, Description and activityTag. The SPARQL query language is used to query the annotation server and the retrieved annotations are transformed into a set of application-dependent features with representative labels based on users’ activity recognition level selection. After the specific hierarchical SVM classification model is built for all of the activity tags, new tri-axial accelerometer data can be input to the trained SVM classifier to automatically tag the input data. The predicted results are displayed in the timeline visualization pane, where experts can check and confirm or correct them. Statistical analyses of animal activity information are conducted using the annotated results to compute average duration time of a specific activity (e.g., “running”), minimum duration time of a specific activity, maximum duration time of a specific activity, total time spent on a specific activity, standard deviation of the duration time of a specific activity and total number of a specific activity that occurred during the entire duration. The generated results are displayed in a simple 3D pie chart (see Figure 3.5). This derived data can also be used to estimate data such as average daily energy consumption and daily calorie/food requirements.

### 3.3.3 User Interface

The SAAR user interface, accessible via a Firefox or Chrome Web browser, enables users to interactively:

- Zoom in or zoom out of the timeline visualization interface to precisely attach an activity tag to a segment of tri-axial accelerometer data streams (motion along the X, Y and Z axes);
- Synchronize the video player with the timeline visualization so users can attach a annotation to either a segment of tri-axial accelerometer data stream or the video and the generated annotation is attached to both segments;
- Delete, edit, or correct annotations;
- Search and retrieve annotations based on annotation content and metadata. For example: give me all annotations created by a user ‘Juana’ between the ‘2012-03-01 00:00:00’ and ‘2012-03-02 00:00:00’;
- Dynamically train an SVM activity classifier using annotated data streams and then apply this trained classification model to newly generated accelerometer data streams to automatically tag activities; and finally
- Statistically analyse the tags on a data stream to calculate relative times spent by a particular animal or species on each activity.

Figure 3.3 illustrates the SAAR Plot-Video visualization interface and the annotation interface. The top left of the interface shows the Plot interface and the tri-axial accelerometer data stream (for a
domestic dog). The X-axis data is yellow, the Y-axis data is blue and, and Z-axis data is red. Users are able to zoom in and zoom out to observe the data streams in more detail, using the mouse scroller. In the bottom left of the interface is the video player which provides play, pause and stop buttons, which enable the video to be precisely synchronized with the tri-axial accelerometer data streams. When creating an annotation, users are required to input data including: the Creator, activityTag and Description in an annotation form displayed on the right hand side of the user interface. The successfully created annotations are stored on the RDF triple store and listed in the Annotation List.

Figure 3.4 shows how users can retrieve specific annotations to train a SVM (C-SVC) activity classifier. It illustrates how a user searches and retrieves all annotations (that describe running, walking and standing) involving a specific dog actor (with ID = “germanPointer1”) to train a low-level classifier.

Figure 3.5 shows a screenshot of the results of applying a low-level dog SVM activity classifier. This classifier identified three dog activities (walking, running and standing) and the result is shown in the plot visualization with the activity type tags displayed in blue along the top. The pie chart on the right shows the statistical information about each activity. From the pie chart, it can be affirmed that this specific animal spent 29.7% (164 seconds) of his time running, 24.2% (134 seconds) of its time walking, and 46.1% (255 seconds) of its time standing.

Figure 3.3: Screenshot of SAAR Plot-Video visualization interface and the annotation interface
Figure 3.4: User interface showing retrieval of specific annotations to train a C-SVC activity classifier.

Figure 3.5: Screenshot of the SAAR Interface with dog activity identification results.
3.4 Evaluation and Results

This section describes the evaluation methods that were employed to assess the proposed SAAR approach.

- In the first stage, the effect of the volume of training data (NA = number of annotated training data sets for each behaviour) is investigated. This is significant because NA represents the amount of work that domain-experts have to do to develop the manually annotated training corpus.

- In the second stage, the SAAR approach described in Chapter 3 is compared with existing best-practice classification algorithms, including ANN (Artificial Neural Networks) [67, 68] and HMM (Hidden Markov Models) [65] on the collected dog data set.

- In the third stage, the performance of SAAR is firstly evaluated based on the results of the experiments on the dog and badger data sets, which aims to determine whether an activity recognition classifier trained using data from a given species (i.e., a domestic dog) can be accurately applied to same species (i.e., data from another dog). Secondly, the performance of domestic dog classifiers was evaluated on animal data sets collected from animals at Australia Zoo, which aim to determine whether an activity recognition classifier trained using data from one species (i.e., a domestic dog) can be usefully applied to other species (i.e., a badger, a cheetah, a tiger, a wombat, a kangaroo and an echidna) or to a wild species (i.e., a dingo) of similar size and gait.

- Finally, the usability of the SAAR system was evaluated based on feedback from a group of eight students and ecologists.

In previous studies, several methods have been proposed for assessing the performance of the supervised Machine Learning approach [146]. This study uses the following four commonly-accepted performance evaluation metrics which are calculated from the number of correctly and incorrectly recognized tags for each class. These metrics include true positive (TP), false positive (FP), true negative (TN) and false negative (FN). From these four metrics, one can calculate: accuracy ((TP+TN)/(TP+TN+FP+FN)), sensitivity (TP/(TP+FN)), precision (TP/(TP+FP)), and specificity (TN/(FP+TN)).

3.4.1 Effect of NA (Number of Annotated Training Data Sets)

In this chapter, an SVM-based method for animal behaviour recognition is proposed, which requires a number of annotated data sets for training a SVM classification model. Hence, the number of annotated training data sets (NA) is an important factor that needs to be evaluated.
Table 3.3 shows the impact of NumberOfAnnotations for each behaviour type on the Average Precision (AP), the Average Recall (AR), as well as the shortest time (ST) spent on collecting each training data set corresponding to each class of behaviour.

From Table 3.3, the following observations can be made. Firstly, the performance of SAAR is very dependent on NA. As might be expected, the performance of SAAR improves as the NA increases for each class of behaviour. The larger the manually annotated training corpus, the more accurate the machine learning algorithm. Hence, better performance requires a longer time spent preparing training data. The time trials (Table 3.6) show that the average time for an expert to attach an annotation via the Plot timeline is 19.5 seconds and via the video panel is 45.9 seconds. Therefore, in order to achieve a relatively high performance of SAAR (AP=97.4% and 95.8%), a domain-expert has to spend at least 1635 seconds (27.25 minutes) to prepare the training data set for a single class of behaviour.

**Table 3.3: Effect of NA (NA-number of annotated training data for each behaviour) on AP-average precision, AR-average recall. ST- Shortest Time for preparing Training Data Set**

<table>
<thead>
<tr>
<th>Methods</th>
<th>NA</th>
<th>AP</th>
<th>AR</th>
<th>ST (Secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAAR</td>
<td>10</td>
<td>78.5%</td>
<td>79.4%</td>
<td>327</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>83.8%</td>
<td>85.9%</td>
<td>654</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>89.5%</td>
<td>90.4%</td>
<td>981</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>94.3%</td>
<td>92.1%</td>
<td>1308</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>97.4%</td>
<td>95.8%</td>
<td>1635</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>97.5%</td>
<td>95.8%</td>
<td>1962</td>
</tr>
</tbody>
</table>

**3.4.2 Results for Dog Data Set**

Tables 3.4 and 3.5 illustrate the Accuracy and Precision results for ANN, HMM and SAAR. All of the algorithms used the same training corpus (NA=50). The experimental results show that the SAAR method (average accuracy=97.4%; average precision=95.8%) outperformed ANN (average accuracy= 91.1%; average precision=88.0%), and HMM (average accuracy= 94.9%; average precision=93.1%) in terms of both accuracy and precision.

**Table 3.4: Accuracy achieved from applying ANN, HMM and SAAR to the well-trained dog data set**

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Standing</th>
<th>Sitting</th>
<th>Lying</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>90.77%</td>
<td>92.27%</td>
<td>96.53%</td>
<td>80.41%</td>
<td>91.51%</td>
<td>91.076%</td>
</tr>
<tr>
<td>HMM</td>
<td>94.03%</td>
<td>96.36%</td>
<td>95.25%</td>
<td>92.7%</td>
<td>96.5%</td>
<td>94.968%</td>
</tr>
<tr>
<td>SAAR</td>
<td>96.08%</td>
<td>97.69%</td>
<td>97.72%</td>
<td>97.3%</td>
<td>98.02%</td>
<td>97.362%</td>
</tr>
</tbody>
</table>
### Table 3.5: Precision achieved from applying ANN, HMM and SAAR to the well-trained dog data set

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Standing</th>
<th>Sitting</th>
<th>Lying</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>88.3%</td>
<td>86.89%</td>
<td>90%</td>
<td>96.27%</td>
<td>89.54%</td>
<td>88.0%</td>
</tr>
<tr>
<td>HMM</td>
<td>90.9%</td>
<td>87.83%</td>
<td>95.02%</td>
<td>90.62%</td>
<td>96.38%</td>
<td>93.15%</td>
</tr>
<tr>
<td>SAAR</td>
<td>93.17%</td>
<td>92.14%</td>
<td>97.24%</td>
<td>97.66%</td>
<td>98.88%</td>
<td>95.81%</td>
</tr>
</tbody>
</table>

#### 3.4.3 Evaluation Results across Species

This section describes the results from carrying out nine experiments designed to determine whether classification models can be migrated across individuals and/or species. The first and second experiments aim to determine whether an activity recognition classifier trained using data from one species can be accurately applied to another individual from the same species. The third to ninth experiments aim to determine whether an activity recognition classifier trained using data from one species can be usefully applied to other species or to a free-ranging species of similar size and gait.

- The 1st experiment was conducted on dog data with the aim of automatically identifying both high-level (active and inactive) and low-level activities (running, walking, standing, lying and sitting). Two classification models (a high-level classifier and a low-level classifier) were developed by feeding the training set (6 dogs, 10 mins of data each) into the SVM (C-SVC algorithm). Then the random dog datasets (6 dogs, over 10 mins of data each) were submitted into the classifiers and the automatically generated results were compared with the ground truth data (which had been manually tagged using the corresponding video as ground truth).

- The 2nd experiment involved training both a high level Eurasian badger activity classifier (to identify high level activities: active and inactive), as well as a low level Eurasian badger activity classifier (to identify low level activities: running, walking, standing, lying, climbing and foraging). The untagged datasets for the Eurasian badgers were then submitted into the classifiers and the results were compared with the manually tagged reference data.

- The 3rd experiment involved using the high and low level classifiers generated from dog training data to automatically tag the corresponding Eurasian badger data. These results were then compared with the results from the second experiment to see if the dog classifier could successfully be used to recognize badger activities.

- The final 4th to 9th experiments involved using the high level and low level classifiers generated from the domestic dog training data to automatically tag accelerometer data sets captured from: an Australian dingo, an African cheetah, an Bengal tiger, a hairy-nosed wombat, an Eastern Grey kangaroo, and a short-beaked echidna. The comparative results of these six experiments indicate
how the domestic dog classifiers perform on accelerometry data collected from a wide range of other species.

The performance results of the eighteen activity classification models are presented in Figure 3.6. There are two graphs for each experiment (e.g., E1a and E1b) – these represent the results from the high level classifier and low level classifier respectively.

The following results were obtained from the first and second experiments (E1 and E2):

- The results from the first two experiments (for domestic dogs and badgers) reveal that the high-level classification models (Figure 3.6 E1a and E2a) produce: accuracy > 97%, sensitivity > 96%, precision > 97% and specificity > 96%. Overall, these results are excellent and better than the low-level classifiers (Figure 3.6 E1b and E2b) which produce: accuracy > 95%, sensitivity > 78%, precision > 78% and specificity > 96%.

- In addition, the low-level dog classification model (Figure 3.6 E1b) (accuracy > 96%, sensitivity > 92%, precision > 92% and specificity > 96%) performed better than the low-level badger classification model (Figure 3.6 E2b) (accuracy > 95%, sensitivity > 78%, precision > 78% and specificity > 96%). The reason for this is that, compared with the badger dataset, the dog data set contains less noise. Domestic dogs that were led by their owners were able to perform the requested range of activities much more specifically than undomesticated badgers being monitored in the wild.

The third experiment produced the following results:

- The third experiment (application of dog classifiers to badger data sets) shows that using the domestic dog classification model to recognize Eurasian badgers’ activities does not perform as well as the other two experiments (experiments 1 and 2), especially if the high level (active/inactive) classifiers for experiment 3 are compared against experiment 2 (whose classification model was generated from badger data). Although there was a drop in performance, the results are still quite positive. The high level classifier (Figure 3.6 E3a) produced: accuracy > 92%, sensitivity > 88%, precision > 85% and specificity > 88%, whilst the low level classifier (Figure 3.6 E3b) produced: accuracy > 83%, sensitivity > 81%, precision > 79% and specificity > 85%. To conclude, migrating the classification models across species does not perform as well as species-specific classification models; however, in situations where there is no video or ground truth, it can be used as an effective first pass. The resulting labels can be corrected or refined manually by experts afterwards. The other problem with migrating a classification model across species is that the activity terms may differ. For example, the dog activity ontology does not include the terms ‘foraging’ and ‘climbing’ –
terms which are specific to the badger activity ontology. Accelerometry data from the King Charles Spaniel was deliberately captured whilst it was performing ‘foraging’ and ‘climbing’ activities. However, in general, there may not be a one-to-one mapping between terms in activity ontologies across species.

- The results from the 4th to 9th experiments show that all five of the behavioural modes (running, walking, standing, sitting and lying) were identified in five of the six test subjects using the behavioural classification module built using acceleration data collected from the domestic dog. These were the dingo, cheetah, tiger, wombat, kangaroo, and echidna. The classification module performed best (accuracy > 96%, sensitivity > 77%, precision > 83% and specificity > 94%) at behavioural mode recognition when applied to accelerometry data collected from the same species (a dingo) (Figure 3.6 E4a and E4b). Behavioural classification results were also good (accuracy > 93%, sensitivity > 83%, precision > 79% and specificity > 95%) for a different species (a cheetah), if the SL:SH (Spine Length: Spine Height) was similar (2.25:2.52) to that of the surrogate (a dog) (Figure 3.6 E5a and E5b), but the results were poor (accuracy > 70%, sensitivity > 44%, precision > 52% and specificity > 86%) in species (tiger and wombat) whose SL:SH was 1.5 to 2 fold greater than that of the surrogate (Figure 3.6 6a, E6b, E7a and E7b). Behavioural classification capacity was poor (accuracy > 57%, sensitivity > 10%, precision > 15% and specification > 56%) for individuals whose SL:SH was greater than three-fold (kangaroo and echidna) that of the surrogate (Figure 3.6 E8a, E8b, E9a and E9b). Overall, there was a significant negative linear relationship between the mean classification score and variation of SL:SH the surrogate and the study species (Figure 3.7).
Figure 3.6: The experimental results from applying a classifier trained using dog data to accelerometry datasets for different species
This research has also shown that a behavioural classification module trained using accelerometry data collected from one individual can be used to identify and quantify behaviour modes in different individuals and even different species. The performance of the behavioural classification module was highly accurate for individuals of the same species and remained at over 80% for quadruped species that were similar in body size and body shape or were phylogenetically close to the surrogate species. For each study species, SL:SH (Spine Length: Spine Height) is the ratio between spine length and minimum spine height above the ground. The dog has the lowest SL:SH of all species studies, and therefore as the SL:SH of the test subjects increased over that of the dog, the capacity of the SVM to distinguish each behavioural mode was reduced in a linear manner. The experimental results indicate that optimum performance of this approach for classifying across species occurs when the SL:SH ratio of the subject is no greater than two-times the surrogate’s SL:SH ratio. However further experimental data captured from a wider range of different species is necessary to verify this precisely.

Figure 3.7: Mean classification scores versus SL:SH (Spine Length: Spine Height)

3.4.4 Usability Evaluation of Semantic Annotation

This evaluation step involved evaluating the usability of the SAAR system by undertaking a user survey and observing users’ behaviour. Eight users (staff and students from the University of Queensland School of ITEE and Ecolab), were asked to respond to each of the following questions using a five-point Likert scale, where 1 = Strongly agree; 2 = Agree; 3 = Neither agree nor disagree; 4 = Disagree; and 5 = Strongly disagree:

- I think the annotation interface is a useful tool.
- I found the annotation interface easy to use.
- I found the suggested tags appropriate.
I found the search options useful.
I found the search interface easy to use.
I found it easy to train the automatic activity classification (SVM) engine.
I felt confident using SAAR.
I think my colleagues would learn SAAR quickly.
I would need a lot of training before I could use SAAR effectively.
The pie chart showing statistical information about each activity is useful.

SAAR’s efficiency was determined by measuring the average time it took a user to: create an annotation via the X/Y/Z visualization pane; create an annotation via the video pane; retrieve existing annotations to train a SVM classifier; apply a newly generated classifier to predict animal activities.

Each user was given a brief tutorial in the use of the SAAR system and then assigned a specific set of annotation and recognition tasks. The time taken to complete each task was recorded.

The questionnaire results were very positive. All of the users who were surveyed found the annotation interface, search interface and the pie chart to be useful, and believed that the animal research community of the University of Queensland could learn to use the SAAR system quickly. A majority, 87.5% of users, found the system, including the visualization, annotation and search interface, easy to use. Only 12.5% of users felt that they would require more time to learn to use it effectively.

Aspects that required further information or clarification included; instructions on how to operate the zoom in and zoom out functions for the timeline visualization; and explanations of the meaning of each of the search options.

### Table 3.6: Summary of time taken by user group to perform requested tasks

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Time Range</th>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1. Create a new annotation using the Plot timeline</td>
<td>15–32secs</td>
<td>19.5secs</td>
</tr>
<tr>
<td>Task 2. Create a new annotation using the video pane</td>
<td>15secs–3.5mins</td>
<td>45.9secs</td>
</tr>
<tr>
<td>Task 3. Search and retrieve annotations and input as training data to generate new classifier</td>
<td>4secs–1min</td>
<td>13.0secs</td>
</tr>
<tr>
<td>Task 4. Submit new 15 min dataset into classifier, generate automatic tags and display in visualization pane</td>
<td>4secs–1.5min</td>
<td>6.0secs</td>
</tr>
</tbody>
</table>
Table 3.6 shows the results of the time trials. The average time taken by users to create an annotation through the Plot panel was 19.5 seconds which was less than the time required to create an annotation using the video pane. This is because, typically, users had to replay video segments multiple times (rewind, pause and replay) to clarify what the animal was doing. The time to complete Tasks 3 and 4 depended on both the size of the data set and the number of input parameters. The average time for Task 3 is 13 seconds, while the average time for Task 4 is 6 seconds.

It was not possible to compare these times with other comparable systems because, to our knowledge, there are no other systems that support similar functionalities. However, these results indicate that users are able to use the SAAR quickly and effectively to complete the required tasks and that in general, SAAR significantly expedites the process of analysing large volumes of 3D accelerometry data.

3.5 Summary

This chapter presents a Semantic Annotation and Activity Recognition (SAAR) approach, which integrates semantic annotation with Support Vector Machine (SVM) techniques to automatically identify animal behaviours from 3D accelerometry data streams. The experiments show that the proposed SAAR approach is able to automatically identify animal behaviour from 3D accelerometry data streams with high levels of accuracy, sensitivity, precision and specificity. Also, the experiments show that a behavioural classification module trained using accelerometry data collected from one individual can be used to identify and quantify behaviour modes in different individuals and even different species. The proposed approach greatly benefits those researchers who are using accelerometers to quantify animal movement and behaviour.

Despite the convincing results presented above, the user tests also revealed one significant limitation. The adoption of a supervised machine learning algorithm SVM, requires ecologists to spend a significant amount of time and effort manually annotating the training data set. This is time-consuming and subjective. Therefore, the next chapter of this thesis (Chapter 4) investigates a new approach to automatic animal behaviour recognition that potentially doesn’t require a large volume of annotated data.
Chapter 4: Optimal Graph-based Learning for Automatic Annotation of Animal Accelerometry Data and General Classification Tasks

4.1 Overview

One of the limitations of the approach described in Chapter 3, is the need to provide a manually annotated training corpus of sufficient size to ensure that the classification model generates precise results. Enabling domain experts to manually attach meaningful descriptors to segments of raw accelerometry data streams is expensive and time-consuming. It is often subjective and biased (based on the individuals’ views), is not scalable for large data collection, and is prone to human error. Instead of learning from large amount of labelled data, Semi-Supervised machine Learning (SSL) algorithms learn both from labelled and unlabelled data [83]. Also, previous research has shown that graphs provide a natural way to represent data in a variety of domains [83], thus graph-based SSL algorithms have been successfully used to extract class-instance pairs from large labelled and unlabelled data sets for human activity behaviour recognition [84, 85].

The proposal underlying this chapter is that the time, cost and effort associated with manual annotation, can be reduced by applying graph-based learning (to the graphs generated from features extracted from windows over the accelerometry data streams) to automatically annotate the accelerometry data streams with behaviour annotations. Graph-based learning is a promising paradigm for modelling the manifold structures that often exist in massive data sources in high-dimensional spaces. It has been shown to be effective in many emerging applications such as multimedia annotation, classification, ranking and retrieval [31, 33, 36]. The graph construction scheme essentially determines the performance of these graph-based learning algorithms.

Most existing research in this field [31, 94, 95] constructs the graph empirically, and the graph is based on single information cue. In this chapter, the proposed approach involves learning an optimal graph (OGL) from multi-cues (i.e., initial labels and multiple-modality features), which can more accurately embed the relationships between data points (labels and features). Then, a series of new machine learning algorithms for various graph-based learning tasks can be derived upon the OGL method. More specifically, OGL is incorporated with a semi-supervised learning model. This model is further extended to address out-of-sample and noisy label issues. To evaluate this approach, two types of evaluations are performed:
• A comparison of the OGL approach with SAAR - that demonstrates that OGL is able to achieve similar performance results to SAAR but with a smaller labelled training dataset. In other words, OGL reduces the workload of domain-expert in terms of manually annotating data streams using the video as ground truth, to generate a training corpus.

• Experiments on standard evaluation image datasets (Corel5k, ESP Games and IAPRTC) that show the consistent superiority of OGL over the state-of-the-art graphs for image annotations tasks and also demonstrate the more general applicability of this approach (for classification tasks beyond animal activity recognition).

4.2 Optimal Graph Learning

The proposed approach involves integrating OGL with Semi-Supervised Learning (SSL) and evaluating it in the context of both:

1) Animal behaviour classification; and

2) General image annotation.

An overview of OGL approach, which consists of three phases, is illustrated in Figure 4.1:

• Firstly, features are extracted from the original data set (such as 3D animal accelerometry data streams or images). For the animal behaviour classification task, the features (i.e., standard deviation vector, signal magnitude area vector, waveform length vector, frequency-domain features and inheritance parameters described in Section 3.2) are extracted using a window size of 3 seconds with an overlap of 1 second (2 sampling points for a 1 Hz sampling rate) between consecutive windows. If the input is images, 15 different visual features are extracted from each image, including one GIST descriptor [147], six global colour histograms, and eight local bag-of-visual-words features.

• Secondly, a similarity graph is constructed for each feature and also on the partial tags to exploit the relationship among the data points. In this step, OGL is used to learn an optimal graph from multiple cues (i.e., initial labels and multiple-modality features). In other words, OGL constructs an optimal graph from multiple feature graphs and a partial labelled graph. The details underpinning the OGL approach are described in sub-sections 4.2-4.3.

• Finally, the learnt optimal graph is integrated with SSL to perform the task of animal behaviour classification or general image classification.
Figure 4.1: An overview of OGL

4.2.1 Terms and Notations

\( X = \{x_1, x_2, \ldots, x_n\} \) represents a set of \( n \) training instances, and \( y_i = \{0, 1\}^r \) is the label for the \( i \)-th training instance \((1 \leq i \leq n)\), and \( c \) is the number of classes. The first \( l \) points \( x_i \ (i \leq l) \) are labelled as \( Y_l \), and the remaining \( u \) points \( x_i \ (l+1 \leq i \leq n) \) are unlabelled. The goal is to predict the label \( F_u \) of the unlabelled points. Define a \( n \times c \) matrix \( F = \begin{bmatrix} F_l \\ F_u \end{bmatrix} \) with \( F_l = Y_l \) and \( F_u = \{0\}^u \).
Suppose that for each training instance, features from $v$ cues (i.e., different features) are obtained and thus each training instance has $v$ features. Let $X^t = \{x'_i\}_{i=1}^n$ denotes the feature matrix of the $t$-th view of training instances, where $t \in \{1, \ldots, v\}$.

### 4.2.2 Optimal Graph Learning-based SSL

The traditional graph based semi-supervised learning usually solves the following problem:

$$
\min_{F,F_\sim Y} \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}
$$  \hspace{0.5cm} (4.1)

where $f_i$ and $f_j$ are the labels for the $i$-th and $j$-th instances, and $S$ is the affinity graph with each entry $s_{ij}$ representing the similarity between two instances. The affinity graph $S \in \mathbb{R}^{n \times n}$ is usually defined as follows:

$$
s_{ij} = \begin{cases} 
    e^{-\|x_i - x_j\|^2 / \sigma^2}, & \text{if } x_i \in \mathcal{N}_k(x_j) \text{ or } x_j \in \mathcal{N}_k(x_i) \\
    0, & \text{else}
\end{cases}
$$  \hspace{0.5cm} (4.2)

where $\mathcal{N}_k(\cdot)$ is the $K$-nearest-neighbor set and $1 \leq (i, j) \leq n$. The variance $\sigma$ will affect the performance significantly, and it is usually empirically tuned. Also, it is derived from single information cue. To address these issues, an optimal graph $S$ should be learnt from multiple cues.

Without loss of generality, it is supposed that the multiple cues include given label information $F$ and multiple features information $X^t = \{x'_i\}_{i=1}^n$. An optimal graph $S$ should be smooth on all these information cues, which can be formulated as:

$$
\min_{S,\alpha} g(F,S) + \mu \sum_{i=1}^v \alpha_i h(X^t, S) + \beta r(S, \alpha)
$$  \hspace{0.5cm} (4.3)

where $g(F,S)$ is the penalty function to measure the smoothness of $S$ on the label information $F$ and $h(X^t, S)$ is the loss function to measure the smoothness of $S$ on the feature $X^t$. $r(S, \alpha)$ are regularizers defined on the target $S$ and $\alpha$. $\mu$ and $\beta$ are balancing parameters, and $\alpha_i$ determines the importance of each feature.

The penalty function $g(F,S)$ should be defined in the way such that close labels have high similarity and vice versa. In this study, it is defined as follows:
\[
g(\mathcal{F}, \mathcal{S}) = \sum_{ij} \| f_i - f_j \|_{2}^2 s_{ij} \tag{4.4}
\]

where \( f_i \) and \( f_j \) are the labels of data point \( x_i \) and \( x_j \). Similarly, \( h(\mathcal{X}', \mathcal{S}) \) can be defined as:

\[
h(\mathcal{X}', \mathcal{S}) = \sum_{ij} \| x_i' - x_j' \|_{2}^2 s_{ij} \tag{4.5}
\]

Note that for simplicity, the distance based method is used to learn the similarity graph here. Other options, that are based on the reconstruction coefficients methods to achieve better performance, are discussed in the next section (Section 4.2.3). Instead of preserving all the pairwise distances, preserving the pair distances of the \( K \)-nearest neighbors is considered here, i.e., if \( x_i' \) and \( x_j' \) (or \( f_i \) and \( f_j \)) are not \( K \)-nearest neighbors of each other, their distance will be set to a large constant.

The regularizer term \( r(S, \alpha) \) is defined as:

\[
\beta r(S, \alpha) = \mu \| S \|_{F}^2 + \beta \| \alpha \|_{2}^2
\]

\( S \geq 0, \quad S1 = 1, \quad \alpha \geq 0 \) and \( \alpha^T 1 = 1 \) are further constraints. Then the objective function for learning optimal graph can be obtained by replacing \( g(\mathcal{F}, \mathcal{S}) \), \( h(\mathcal{X}', \mathcal{S}) \) and \( r(S, \alpha) \) in Eq.4.3 using Eq.4.4, Eq.4.5 and Eq.4.6. And by combining Eq.4.1 with Eq.4.3, the objective function for optimal-graph based SSL is obtained, as follows:

\[
\min_{S, \mathcal{F}, \alpha} \sum_{ij} \| f_i - f_j \|_{2}^2 s_{ij} + \mu \sum_{ij} \| \alpha_i - \alpha_j' \|_{2}^2 s_{ij}
+ \mu \| S \|_{F}^2 + \beta \| \alpha \|_{2}^2
\tag{4.7}
\]

\[
\text{subject to: } \begin{cases} 
S \geq 0, \quad S1 = 1 \\
F_i = Y_i \\
\alpha \geq 0, \quad \alpha^T 1 = 1 
\end{cases}
\]

### 4.2.3 Iterative Optimization

An iterative method is proposed to minimize the above Eq. 4.7. Firstly, \( S = \sum_{i} S' / \nu \) is initialized with each \( S' \) being calculated using Eq.4.2, and \( \alpha' = 1 / \nu \) is initialized. \( S \) is further normalized as \( S = \left( D_{2}^{-\frac{1}{2}} \right)^T S D_{2}^{-\frac{1}{2}} \). Once these initial values are given, in each iteration, firstly \( F \) is updated given \( S \) and \( \alpha \), and then update \( S \) and \( \alpha \) by fixing the other parameters. These steps are described as below:

**Update \( F \)**
By fixing $S$ and $\alpha$, $F$ is obtained by optimizing Eq.4.7. It is equivalent to optimize the following objective function:

$$\min_{F, F = I - S} \sum_{ij} \left\| f_i - f_j \right\|^2 s_{ij} = \min_{F, F = I - S} \left\| F(I - S)F^T \right\|$$ (4.8)

Where $I$ is an identity matrix. Let $L = I - S$, and differentiate the objective function Eq.4.8 with respect to $F$, $LF$ is obtained:

$$LF = 0 \Rightarrow \begin{bmatrix} L_{ii} & L_{iu} \\ L_{ui} & L_{uu} \end{bmatrix} \begin{bmatrix} F_i \\ F_u \end{bmatrix} = 0$$

$$\Rightarrow \begin{cases} L_{ii} F_i + L_{iu} F_u = 0 \\ L_{ui} F_i + L_{uu} F_u = 0 \end{cases}$$ (4.9)

Then $F_u^*$ is obtained:

$$F_u^* = -L_{uu}^{-1}L_{ui}F_i$$ (4.10)

Update $\delta$

By fixing $F$ and $\alpha$, $\alpha$ is obtained by optimizing Eq.4.7. It is equivalent to optimize the following objective function:

$$\sum_{ij} \left\| f_i - f_j \right\|^2 s_{ij} + \mu \sum_{ij} \left( \alpha_i \left\| x'_i - x'_j \right\|^2 s_{ij} \right) + \mu \gamma \left\| S \right\|^2$$ (4.11)

It can be reformulated as:

$$= \min_{S, S \geq 0, S \geq 1} \sum_i \text{tr} \left( \mu \gamma s_i s_i^T + \left( a_i + \mu b_i \right) s_i^T \right)$$

$$\Rightarrow \min_{S, S \geq 0, S \geq 1} \sum_i \text{tr} \left( s_i s_i^T + \frac{a_i + \mu b_i}{\mu \gamma} s_i^T \right)$$ (4.12)

and it is equivalent to:

$$\min_{S, S \geq 0, S \geq 1} \sum_i \left\| s_i + \frac{a_i + \mu b_i}{2\mu \gamma} \right\|^2$$ (4.13)

Where $b_j = \{ b_j, 1 \leq j \leq n \}$ with $b_j = \sum_i \alpha_i \left\| x'_i - x'_j \right\|^2$ and $a_j = \{ a_j, 1 \leq j \leq n \} \in R^{1 \times n}$ with $a_j = \left\| y_i - y_j \right\|^2$. The constraints in problem (Eq.4.13) are simplex. The accelerated projected gradient method is used to linearly solve this problem. The critical step of the projected gradient method is to solve the following proximal problem:
\[
\min_{x \in \mathbb{R}, c' \in \mathbb{R}} \frac{1}{2} \|x - c\|_2^2
\]  
(4.14)

An efficient approach is introduced to solve this problem. The Lagrangian function of problem (Eq.4.14) is written as

\[
\frac{1}{2} \|x - c\|_2^2 - \tau (x^T 1 - 1) - \rho^T x
\]  
(4.15)

where \( \tau \) and \( \rho \) are Lagrangian coefficients. Suppose the optimal solution to the proximal problem (Eq.4.14) is \( x^* \), the associated Lagrangian coefficients are \( \tau^* \) and \( \rho^* \). Then according to the KKT condition, the following equations are obtained:

\[
\forall i, \quad x_i^* - c_i - \tau^* - \rho_i^* = 0
\]  
(4.16)

\[
\forall i, \quad x_i^* \geq 0
\]  
(4.17)

\[
\forall i, \quad \rho_i^* \geq 0
\]  
(4.18)

\[
\forall i, \quad x_i^* \rho_i^* = 0
\]  
(4.19)

Eq.4.16 can be written as \( x^* - c - \tau^* 1 - \rho^* 1 = 0 \). According to the constraint \( 1^T x^* = 1 \), \( \tau^* \) is calculated by \( \tau^* = \frac{1 - 1^T c - 1^T \rho^*}{n} \). So \( x^* = (c - \frac{11^T}{n} c + \frac{1}{n} - 1^T \rho^* 1) + \rho^* \).

Denote \( \tilde{\rho} = \frac{1^T \rho^*}{n} \) and \( u = c - \frac{11^T}{n} c + \frac{1}{n} \), then \( x^* \) is wrote as \( x^* = u + \rho^* - \tilde{\rho} 1 \). So \( \forall i \), the following equation is obtained:

\[
x_i^* = u_i + \rho_i^* - \tilde{\rho}
\]  
(4.20)

According to Eq.4.17- Eq.4.20, \( u_i + \rho_i^* - \tilde{\rho} = (u_i - \tilde{\rho})_+ \) is obtained. Then the following equation is obtained:

\[
x_i^* = (u_i - \tilde{\rho})_+
\]  
(4.21)

So, the optimal solution \( x^* \) can be calculated, if \( \tilde{\rho} \) is calculated.
The Eq.4.20 is wrote as \( \rho^*_i = x^*_i + \bar{\rho}^* - u_i \). Similarly, according to Eq.4.17 - Eq.4.19, \( \rho^*_i = (\bar{\rho}^* - u_i) \) is known. Suppose \( c \) is a \( m \)-dimensional vector, then \( \bar{\rho}^* = \frac{1}{m} \sum_{i=1}^{m} (\bar{\rho}^* - u_i) \) is obtained. Defining a function as:

\[
  f(\bar{\rho}) = \frac{1}{n} \sum_{i=1}^{n} (\bar{\rho} - u_i)_+ - \bar{\rho}
\]

(4.22)

So \( f(\bar{\rho}^*) = 0 \) and the root finding problem is able to be solved with Newton method to obtain \( \bar{\rho}^* \).

Then each \( S_i \) can be efficiently solved, and graph \( S \) can be updated.

**Update \( \alpha \)**

By fixing \( F \) and \( S \), \( \alpha \) is obtained by optimizing Eq.4.7. It is equivalent to optimize the following objective function:

\[
  \min_{\alpha \geq 0, \alpha^i \geq 1} \mu \sum \alpha_i \left( \sum_{y} \left( x'_i - x''_i \right)^2 s_{y} \right) + \beta \| \alpha \|^2
\]

\[
  \Rightarrow \min_{\alpha \geq 0, \alpha^i \geq 1} \mu d \alpha + \beta \| \alpha \|^2
\]

(4.23)

Where \( d = \{d_t, 1 \leq t \leq v\} \) with \( d_t = \sum_{y} \left( x'_i - x''_i \right)^2 s_{y} \). It can be reformulated in the form of problem Eq.4.15 and can be solved similarly to obtain \( \alpha \). Next, updating \( F \), \( S \) and \( \alpha \) iteratively until the objective function Eq.4.17 converges, as shown in Algorithm 1.

### 4.3 Extensions of OGL

In this section, several issues associated with employing graph based learning methods in real applications are discussed.

#### 4.3.1 Noisy labels

The user provided instance labels may contain some noise. To address this issue, instead of limiting that the predicted label \( F_i \) to be strictly equal to the given hard labels \( Y_i \), an soft error term \( \| F_i - Y_i \|_F \) is introduced to release this constraint. Then, by fixing \( S \) and \( \alpha \), \( F \) is obtained by solving:

\[
  \min_{F} \sum_{i,j} \left( f_i - f_j \right)^2 s_{y} + \mu \| F_i - Y_i \|_F^2
\]

\[
  = \min_{F} tr F^T (I - S) F + tr \left( (F - Y)^T U (F - Y) \right)
\]

(4.24)
where \(U \in \mathbb{R}^{n \times n}\) is the diagonal matrix. By setting the derivative of the Eq. 4.24 w.r.t \(F\) to zero, the following equations are calculated:

\[
F^* - SF^* + U \left( F^* - Y \right) = 0
\Rightarrow F^* = (I + U - S)^{-1}UY
\] (4.25)

Where \(Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}\) with \(Y_u = (0)^{u \times c}\). Experimental results show Eq. 4.25 has superior performance over Eq. 4.10.

### 4.3.2 Out-of-sample Extension

Out-of-sample refers to learning an annotation function that is able to label new data points. This can be achieved by adding a fitting model and a regularizer to the objective function Eq. 4.7. e.g.,

\[
\|X W + \mathbf{1}b - F\|_F^2 + \eta \|W\|_F^2,
\]

where \(W \in \mathbb{R}^{nc \times c}\), \(b \in \mathbb{R}^{1 \times c}\) and \(\mathbf{1}\) is a vector of all ones. To obtain the optimal solution \(W^*\) and \(b^*\), the derivatives of the objective function with respect to \(W\) and \(b\) equal to zero are set, thus:

\[
b^* = \frac{1}{n} (\mathbf{1}^T F - \mathbf{1}^T X W)
\] (4.26)

\[
W^* = (X^T L_{c} X + \eta I)^{-1} X^T L_{c} F
\] (4.27)

where \(X\) is the concatenation of different features \(X',\) and \(L_{c} = I - \mathbf{1}\mathbf{1}^T\). Then \(\|X W + \mathbf{1}b - F\|_F^2 + \eta \|W\|_F^2\) can be reformulated as:

\[
\text{tr} \left( F^T BF \right)
\] (4.28)

Where \(B = L_{c} - L_{c} X \left( X^T L_{c} X + \eta I \right)^{-1} X^T L_{c}\). Then, by adding this fitting model, \(F\) can be obtained by solving:

\[
\min_{F} \text{tr} F^T \left( I - S + \omega B \right) F + \text{tr} \left( F - Y \right)^T U \left( F - Y \right)
\Rightarrow F^* = (I + U - S + \omega B)^{-1}UY
\] (4.29)

where \(\omega\) is the parameter for the fitting model. Then the annotation function \(W\) and \(b\) are obtained. Note that other fitting models can also be applied here, e.g., SVM, fast image tagging [36]. In [36], they address the incomplete tagging problem by introducing a term \(\tilde{B}\) to enrich the existing tags. Then a co-regularized learning scheme is adopted to jointly learn the annotation function \(\tilde{W}\) and tag.
enrich function $\tilde{B}$, as follows: $\sum \|Bf_i - \hat{x}W\|^2$. For simplicity, the least square regression model is adopted to tackle the out-of-sample problem. However, the performance can be further improved by incorporating better fitting models, which is also demonstrated in the experiments.

### 4.3.3 Different Graph Construction Model

In our work, distance based method is utilized to construct the similarity graph in Eq.4.4 and Eq.4.5 for simplicity. It can be further extended by using different graph construction models. One possible way is to adopt the reconstruction coefficients methods, which can be calculated by solving:

$$\min_s \sum_i \|x_i - D_i s_i\|, \text{s.t.} \ 1^T s_i = 1$$

(4.30)

where $s_i \in \mathbb{R}^n$ is the coefficient of $x_i$ over $D_i$ and $D_i$ consists of the $k$ nearest neighbors of $x_i$ in Euclidean space.

Recently, some studies have exploited the inherent sparsity of sparse representation to obtain a block-diagonal affinity matrix, e.g., SSC[99] and the L1-graph [31]. In [31] the L1-graph is proposed for image analysis, which solves the following problem:

$$\min_s \|s_i\|_1, \text{s.t.} \ 1^T s_i = 1$$

(4.31)

Where $s_i \in \mathbb{R}^n$ is the sparse representation of $x_i$ over the dictionary $X$ and $\delta$ is the error tolerance.

Another recently proposed method, LRR [101, 102], aims to find the lowest-rank representation, rather than the sparsest, by solving:

$$\min_{S,k} \text{rank}(S) + \lambda \|E\|_1, \text{s.t.} \ X = XS + E$$

(4.32)

where $S \in \mathbb{R}^{n \times n}$ is the coefficient matrix of $X$ over the data set itself and $E$ is the reconstruction error. These graph construction methods have reported superior performance over the distance-based graph construction method. The OGL approach proposed here can be further improved if these graph construction methods are adopted.

### 4.4 Experimental Evaluations

To evaluate the performance of OGL-SSL, two types of evaluation are carried out:

- Firstly, the results from applying OGL-SSL are compared with the results produced by SAAR (described in Chapter 3) – when attempting to automatically recognize animal behaviour from 3D accelerometer data streams.
Secondly, OGL-SSL is compared with other graph construction methods when applied to the task of automatic image annotation. OGL-SSL is also compared with state-of-the-art none graph-based image annotation methods. In addition, the effect of parameter $K$ (number of neighbours) is evaluated.

### 4.4.1 Comparison of OGL with SAAR

In this experiment, the dog data which was collected in the first stage of this thesis is utilized (see Section 3.3.1). The method for extracting features, graphs and labels from 3D animal accelerometry data streams using OGL is described in Section 4.2. The average precision (AP) and average recall (AR) of the results generated from the OGL approach are listed in Table 4.1 alongside the results generated using SAAR (SVM).

From Table 4.1, several observations are made:

- For both SVM and OGL, the performance improves as the volume of annotated training data for each behaviour (NA) increases.

- Both SVM and OGL perform best when NA is equal to 50, with AP=97.4%, AR=95.8% for SAAR/SVM and AP=97.1%, AR=96.2% for OGL. The average precision is slightly higher for SAAR/SVM and the average recall is slightly higher for OGL.

- As the NA decreases from 50 to 10, the performance of SVM decreases quickly (AP falls from 97.4% to 78.5% and AR falls from 95.8% to 79.4%), while OGL performance is less adversely impacted (AP only falls from 97.1% to 87.7 and AR only falls from 96.2% to 88.1%).

- To achieve better performance, the SAAR/SVM approach is much more dependent on a large volume of annotated training samples.
Table 4.1: Results on the dog dataset (NA - number of annotated training data for each behaviour; AP – average precision; AR - average recall)

<table>
<thead>
<tr>
<th>Methods</th>
<th>SVM</th>
<th>OGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>AP(%)</td>
<td>AR(%)</td>
</tr>
<tr>
<td>10</td>
<td>78.5</td>
<td>79.4</td>
</tr>
<tr>
<td>20</td>
<td>83.8</td>
<td>85.9</td>
</tr>
<tr>
<td>30</td>
<td>89.5</td>
<td>90.4</td>
</tr>
<tr>
<td>40</td>
<td>94.3</td>
<td>92.1</td>
</tr>
<tr>
<td>50</td>
<td>97.4</td>
<td>95.8</td>
</tr>
</tbody>
</table>

4.4.2 Comparison of OGL with Previous Graph-based SSL Methods

To perform this evaluation, the following three image datasets commonly used for classification evaluations are employed:

- **Corel 5k.** This dataset is an important benchmark for image annotation [148]. It contains around 4,999 images manually annotated with 1 to 5 keywords. The vocabulary contains 260 words. A fixed set of 499 images are used as testing, and the rest is used for training.

- **ESP Game.** This data set contains a wide variety of images including logos, drawings, and personal photos. It is obtained from an online game where two players, who cannot communicate outside the game, gain points by agreeing on words describing the image [36]. This way the players are encouraged to provide important and meaningful tags to images. A subset of 20,000, out of the 60,000 publicly available images, are used.

- **IAPR TC12.** This set of 20,000 images accompanied with descriptions in several languages was initially published for cross-lingual retrieval [36]. It can be transformed into a format comparable to the other sets by extracting common nouns using natural language processing techniques.

**Feature Extraction.** For the first three datasets, 15 different visual descriptors for each dataset are used. These include one GIST descriptor [147], six global color histograms, and eight local bag-of-visual-words features.

**Baseline Methods and Evaluation Metrics.** To evaluate the performance of the optimal graph, OGL is firstly compared with different graph construction methods, i.e., LGC [87], LLE [149], L2graph [95]. Further experiments are conducted to compare all these methods on larger datasets by out-of-sample extension. The extended methods are referred to as $LGC^T$, $LNP^T$, $L2Graph^T$. OGL is also compared with TagProp [150] and fastTag [36] algorithms, which currently achieve the best
performance on the benchmark datasets. To test different fitting models, SSL is also combined with fastTag to compare the performance. Besides, several previously reported results for reference, i.e., CRM [151], NPDE [152], NPDE [152], SML [153], MBRM [154], JEC [155] are included (see Table 4.2 and Table 4.3).

The proposed models are evaluated using standard performance measures that evaluate retrieval performance per keyword, and average the precision and recall over all keywords.

**Precision and recall for fixed annotation length.** Each image is annotated with the 5 most relevant keywords. Then, the mean precision P and recall R over keywords are computed.

**The mean average precision.** The mean average precision (MAP) over keywords is calculated by computing for each keyword the average of the precisions measured after each relevant image is retrieved.

**Results for the Corel 5k Dataset**

In the first set of experiments, OGL is compared to a number of other methods when applied to the Corel5k data set. The results, displayed in Table 4.2, show the precision (P), recall (R) and Mean Average Precision (MAP) for OGL compared with other approaches. Several observations can be made by analysing the results:

- By learning an optimal graph from multiple cues, the performance for image annotation is improved compared with other graph construction methods.

- Compared with other state-of-the-art none-graph-based methods, the OGL method outperforms most of the existing methods, and can achieve comparable performance with the current best result. For a fair comparison, the results from TagProg are reported without the metric learning step. FastTag achieves better performance than OGL since FastTag uses a complex iterative algorithm to learn the annotation function, while OGL utilizes a simple least square regression for graph transduction. After integrating FastTag with our OGL as an out-of-sample extension, our method achieves the best performance.

- The out-of-sample extension degrades the performance of all methods, and reduces the performance gaps of different methods. This is probably due to the linear fitting model.
### Table 4.2: Result on Corel5k

<table>
<thead>
<tr>
<th>Methods</th>
<th>P(%)</th>
<th>R(%)</th>
<th>MAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGC</td>
<td>29.3</td>
<td>34.9</td>
<td>35.5</td>
</tr>
<tr>
<td>LLE</td>
<td>31.4</td>
<td>36.4</td>
<td>37.8</td>
</tr>
<tr>
<td>L2Graph</td>
<td>29.5</td>
<td>29.3</td>
<td>36.4</td>
</tr>
<tr>
<td>OGL</td>
<td>32.2</td>
<td>38.3</td>
<td>38.6</td>
</tr>
<tr>
<td>LGC(T)</td>
<td>28.1</td>
<td>29.5</td>
<td>22.1</td>
</tr>
<tr>
<td>LLE(T)</td>
<td>27.0</td>
<td>29.5</td>
<td>22.3</td>
</tr>
<tr>
<td>L2Graph(T)</td>
<td>28.6</td>
<td>29.4</td>
<td>22.2</td>
</tr>
<tr>
<td>OGL(T)</td>
<td>28.9</td>
<td>29.9</td>
<td>22.6</td>
</tr>
<tr>
<td>CRM</td>
<td>16.0</td>
<td>19.0</td>
<td>#</td>
</tr>
<tr>
<td>NPDE</td>
<td>18.0</td>
<td>21.0</td>
<td>#</td>
</tr>
<tr>
<td>SML</td>
<td>23.0</td>
<td>29.0</td>
<td>#</td>
</tr>
<tr>
<td>MBRM</td>
<td>24.0</td>
<td>25.0</td>
<td>#</td>
</tr>
<tr>
<td>JEC</td>
<td>25.0</td>
<td>29.0</td>
<td>#</td>
</tr>
<tr>
<td>TagProp</td>
<td>30.0</td>
<td>33.0</td>
<td>#</td>
</tr>
<tr>
<td>FastTag</td>
<td>32.0</td>
<td>43.0</td>
<td>#</td>
</tr>
<tr>
<td>OGL+FastTag</td>
<td>32.6</td>
<td>43.8</td>
<td>#</td>
</tr>
</tbody>
</table>

**Results for the ESP Game and IAPR TC12 Datasets**

From the results in Table 4.3, several similar observations (to those above) can be made:

- OGL outperforms other graph construction methods.
- OGL+fastTag achieves the best performance in terms of precision. But it is slightly beaten by TagProp in terms of recall (R). This is because TagProp sacrifices precision for recall.
Table 4.3: Results from using the ESP Game and IAPR TC-12 Data Sets

<table>
<thead>
<tr>
<th>Methods</th>
<th>ESP Game</th>
<th>IAPR TC-12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P(%)</td>
<td>R(%)</td>
</tr>
<tr>
<td>LGC</td>
<td>43.5</td>
<td>18.7</td>
</tr>
<tr>
<td>LLE</td>
<td>43.1</td>
<td>18.5</td>
</tr>
<tr>
<td>L2Graph</td>
<td>42.8</td>
<td>18.5</td>
</tr>
<tr>
<td>OGL</td>
<td><strong>44.3</strong></td>
<td><strong>19.6</strong></td>
</tr>
<tr>
<td>MBRM</td>
<td>18.0</td>
<td>19.0</td>
</tr>
<tr>
<td>JEC</td>
<td>24.0</td>
<td>19.0</td>
</tr>
<tr>
<td>TagProp</td>
<td>39.0</td>
<td><strong>24.0</strong></td>
</tr>
<tr>
<td>FastTag</td>
<td>46.0</td>
<td>22.0</td>
</tr>
<tr>
<td>OGL+FastTag</td>
<td>41.6</td>
<td>22.8</td>
</tr>
</tbody>
</table>

Effect of $K$

The number of neighbors $K$ is an important parameter for OGL (see E.q.2) and it affects the performance of OGL significantly. Figure 4.2 illustrates the performance variance with different number of neighbors $K$ and shows that the parameter $K$ is important to the performance. For LLE, LGC and OGL, the performance decreases as $K$ increases. For L2Graph, as $K$ increases, the performance improves. For Corel5k data set, the optimum value of $K$ for OGL is 5. For ESP Game and IAPR TC12 data sets, the optimum value of $K$ is 10.

(a) Corel5k
4.5 Limitations of OGL

Experimental results on animal datasets indicate that compared with SAAR/SVM, OGL can achieve similar performance results with a much smaller annotated training dataset. Classification experiments on image datasets (Corel5k, ESP Games and IAPRTC), demonstrate the consistent superiority of OGL over the state-of-the-art graphs for image annotation tasks. Despite the convincing results presented above, OGL has some limitations that demand further research efforts. During the training stage, the time cost for graph construction of OGL is huge, and hence limits the scalability of OGL. As the volume of training data increases, the graph construction time will become excessive, and the memory cost for storing the constructed graph will be very high. On the other hand, a small volume of training data will degrade the performance of the OGL. Thus, further research is required to determine the optimum trade-off between training data size and speed of graph construction. Further research is to design a more efficient, faster graph construction algorithm that is scalable for large training datasets. In addition, a linear regression model was adopted to learn the annotation
function for new data points. This is a basic model and it is anticipated that the performance could be improved by utilizing more complex models. Further research is necessary to confirm this.

4.6 Summary

Because of the limitations associated with manually-annotated corpuses required for SVM (as implemented within SAAR as described in Chapter 3) – an approach to data stream classification involving optimal graph learning from multiple cues (i.e., initial labels and multiple-modality features) was proposed.

The underlying hypothesis of this chapter is that a graph approach (OGL) can more accurately represent the relationships between data points (labels or features). Hence OGL is combined with the semi-supervised learning (SSL) model. This model is further extended by addressing out-of-sample and noisy label issues. To evaluate this approach, OGL is firstly applied to the 3D accelerometer data streams to semi-automatically recognize animal behaviour. The results indicate that compared with SVM (SAAR), OGL can achieve similar performance results with a smaller annotated training dataset. In other words, OGL reduces the workload of the domain expert in manually annotating the training corpus.

The second hypothesis is that the OGL approach would also perform better than state-of-the-art image classification methods. Experimental results carried out using three publicly available image datasets indicate the superiority of the OGL+SSL algorithm compared with standard graph construction methods.

Chapters 3 and 4 propose, implement and evaluate innovative methods for automatically annotating sensor data streams collected from accelerometers attached to individual animals that are moving throughout a region of study or particular ecosystem. The aim is to help ecologists understand how the animal is behaving over time within its habitat.

In the next two chapters, the focus shifts to analysing sensor data streams that are collected from wireless sensor networks, in order to understand patterns in changing environmental conditions (e.g., micro-climates) in a particular ecosystem or region of interest. The next two chapters also tackle two new and significant challenges associated with wireless sensor network data streams:

- automatic detection, tagging and filtering of erroneous data;
- reasoning across multiple annotated sensor network data streams to infer higher level knowledge.
Chapter 5: Semantic-based Detection of Segment Outliers and Unusual Events for Wireless Sensor Networks

5.1 Overview

Environmental scientists have increasingly been deploying wireless sensor networks to capture valuable data that measures and records precise information about our environment. One of the major challenges associated with wireless sensor networks is the quality of the data – and more specifically the detection of segment outliers and unusual events. Most previous research has focused on detecting outliers that are errors caused by unreliable sensors and sensor nodes. However, there is an urgent need for the development of new tools capable of identifying, tagging and visualizing erroneous segment outliers and unusual events that occur within sensor data streams. This chapter presents a SOUE-Detector (Segment Outlier and Unusual Event-Detector) system for wireless sensor networks that combines statistical analysis using Dynamic Time Warping (DTW) with domain expert knowledge (captured via an ontology and semantic inferencing rules). The resulting Web portal enables scientists to efficiently search across a collection of wireless sensor data streams and identify, retrieve and display segment outliers (both erroneous and genuine) within the data streams. This chapter describes, firstly, the detection algorithms, the implementation details and the functionality of the SOUE-Detector system. Secondly it evaluates the proposed approach using data collected from a sensor network deployed in the Springbrook National Park in Queensland, Australia. It also compares the SOUE-Detector with existing outlier detection approaches. The experimental results show that the SOUE-Detector can efficiently detect segment outliers and unusual events with high levels of precision and recall.

In the remainder of this chapter, the SOUE-Detector system is described in more detail. Section 5.2 describes the proposed methodology including the case study, data collection and the proposed approach. Section 5.3 describes the Correlated Environmental Sensor Properties (CESP) ontology. Section 5.4 provides detailed information about the detection algorithm for identifying erroneous outliers and unusual events. Section 5.5 describes the SOUE-Detector system architecture and Web portal. In Section 5.6, the evaluation results (precision and recall of the system) are presented and discussed. Finally, Section 5.7 provides a brief conclusion.
5.2 Methodology

5.2.1 Case Study and Data Collection

The case study employed in this research is Case Study #2, described in Section 1.2.2 - the Springbrook Wireless Sensor Network deployed in the Springbrook National Park, located about 96 kilometres south of Brisbane in the south-east region of Queensland, Australia [156]. The project aims to provide a research platform for monitoring how the microclimate and biodiversity of the Springbrook plateau changes over time. Hundreds of solar-powered sensor nodes have been installed and each sensor node carries several sensing devices that collect different environmental variables. These variables include air temperature, relative humidity, air pressure, leaf wetness, soil moisture, wind speed, wind direction and light. The richness of the Springbrook data set makes it an ideal testbed for developing and evaluating the proposed approach for detecting outliers and unusual events. For this case study, 2.5 years’ of data were acquired (from January 2010 to June 2012) from the Springbrook Wireless Sensor Network project. More specifically, three types of sensor observations (air temperature, relative humidity and air pressure) were collected by Vaisla WXT520 weather transmitter. The sampling rate of each sensor was one sample every 10 minutes.

5.2.2 Approach

The methodology for developing the SOUE-Detector and applying and evaluating it comprises the following five steps:

1. Develop a PostgreSQL database to store raw sensor data streams collected from the wireless sensor networks. Specifically, each sensor observation includes the following metadata: the ID of the sensor node, the ID of the sensor device, the name of the sensor device, the sampling date/time of the observation, the type of sensor property, the observation value and the observation measurement unit. Each sensor node has the following metadata: the ID of the sensor node, its geographical location (latitude and longitude), its installation date/time and removal date/time. In addition, each sensor device has the following metadata: the ID of the sensor device, the ID of sensor node to which the sensor device is connected, the sensor observation property, installation date/time and removal date/time.

2. Develop an ontology of Correlated Environmental Sensor Properties (CESP). This ontology defines concepts that describe different correlation types in terms of strength, direction, shape, space-time, composition and complexity.

3. Design and implement the detection algorithm. This step comprises: defining a spatial neighbourhood matrix that describes the spatial correlations between sensor nodes; defining of
spatial neighbourhood matrices that describes the spatial correlations between sensors; detection of suspicious data streams (using Dynamic Time Warping-based similarity computation); development of a user interface for domain experts to define correlation rules; and identification of unusual events from suspicious data streams.

4. Develop a Web-based system to enable users to search wireless sensor data streams for a given time period, and then visualize the detected erroneous outliers and unusual events via a Google Earth Map and timeline interface.

5. Evaluate the performance of the proposed detection algorithm by calculating the precision and recall of the results on eight datasets.

5.3 An Ontology of Correlated Environmental Sensor Properties

Within the Springbrook sensor network (and many other environmental sensor networks), some of the sensor properties have one or more correlations with other properties. For example, humidity and barometric pressure are related to air temperature. Capturing the correlations between sensor properties provides valuable knowledge that can be exploited to improve the accuracy and efficiency of detecting genuine outliers and unusual events. Hence, the first step in detecting outliers and events is to develop a Correlated Environmental Sensor Properties (CESP) ontology [157] that describes the correlations between environmental sensor properties. The CESP is designed by extending the Semantic Sensor Network Ontology (SSN) [50] developed by the W3C Semantic Sensor Network Incubator Group [50] to describe sensors, sensing, the measurement capabilities of sensors and the observations. The SSN ontology is a high-level ontology that does not support the description of the low-level properties, such as the relationships between sensor properties. Hence for the work presented in this chapter, the CESP ontology has been developed by extending the SSN ontology to precisely describe the specific sensor properties and their relationships within a sensor network.
Figure 5.1: The CESP ontology consists of an object property *cesp:Correlation* with a set of sub-object properties (LHS) and a sensor property class *ssn:Property* with a set of sub-classes (RHS)

The CESP ontology (see Figure 5.1) consists of two components: a sensor property class (*ssn:Property*) and an object property (*cesp:Correlation*), which is defined as a relationship between two sensor properties. The subclasses of *ssn:Property* are derived from the Climate and Forecast (CF) metadata conventions [158] that provide a definitive description of climate and forecast data variables and the spatial and temporal properties of the variables. For example, for each of the CF names (e.g., *air_temperature*) a corresponding class e.g., *cesp:air_temperature* was created. A sub-class of *ssn:Property* (e.g., *cesp:air_temperature, cesp:relative_humidity, cesp:air_pressure*) must be observed by a sensor. The property *cesp:Correlation* links one sensor property to another sensor property. In order to accurately describe the correlations between sensor properties, five types of sub-properties of *cesp:Correlation* [159-161] were defined:

1. Strength – *cesp:hasVeryStrongCorrelation, cesp:hasStrongCorrelation, cesp:hasMediumCorrelation, cesp:hasWeakCorrelation, cesp:hasVeryWeakCorrelation*
2. Direction – *cesp:hasPositiveCorrelation, cesp:hasNegativeCorrelation*
3. Shape/Form – *cesp:hasLinearCorrelation, cesp:hasCurvilinearCorrelation, cesp:hasScatteredCorrelation*
4. Space-time – *cesp:hasSpatialCorrelation, cesp:hasTemporalCorrelation, cesp:hasSpatioTemporalCorrelation*
5. Composition – *cesp:hasPartialCorrelation, cesp:hasSimpleCorrelation, cesp:hasMultipleCorrelation*
5.4 Detection Algorithm

In this section, the algorithms used for detecting outliers and unusual events are described. Firstly, the terms and notations used in the algorithms are defined (Section 5.4.1). Next, the similarity computation method that uses Dynamic Time Warping (DTW) to compute the similarity between two windows of data streams is described (Section 5.4.2). Following this, the five steps in the proposed detection algorithm, shown below, are described in detail in Sections 5.4.3 - 5.4.7 respectively:

1. A sensor node matrix is constructed to document the spatial neighbourhood relationships between sensor nodes by defining a threshold value that determines whether two sensor nodes are neighbours or not. If the distance between two sensor nodes is less than or equal to the threshold value, then they are neighbours. If the distance between two sensor nodes is greater than the threshold value, then they are not neighbours (Section 5.4.3)

2. A matrix that documents spatial relationships between sensors is constructed. Such matrices are built by integrating the sensor node matrix constructed in the first step with the sensor node configuration (that defines which sensors are attached to which sensor nodes) of a wireless sensor network. (Section 5.4.4)

3. Next, suspicious data within the sensor data streams is detected by: choosing a sensor type (e.g., temperature) and determining the similarities between neighbouring data streams for each sensor type. The algorithm determines whether the sensor is collecting suspicious data by applying a predefined rule to the calculated similarity results. This is repeated for all sensors in the network. (Section 5.4.5)

4. Domain experts are able to define correlation rules through a graphical user interface. The user interface enables a user to specify relationships (e.g., very strong, strong, medium, weak, very weak) among sensor properties. The specified relationships are saved and used in the next/final step. (Section 5.4.6)

5. Finally, the domain expert rules (from the previous step) are combined with the detected suspicious results to determine whether a suspicious segment data stream is a true error or an unusual event. (Section 5.4.7)

5.4.1 Notations and Definitions

Let \( \{ s_i \}_{i=1}^{m} = \{ s_1, s_2, \ldots, s_m \} \) denote all the sensors deployed in the wireless sensor network, where \( s_i = (s_{i1}, s_{i2}, \ldots, s_{in}) \) indicates the \( i \)-th type of sensor employed in the wireless sensor network, \( s_{ij} \) indicates the \( i \)-th type of sensor installed on the \( j \)-th sensor node \( sn_j \), and \( n \) is the total number of
78

sensor nodes deployed in the wireless sensor network. For instance, \( S_1 \) denotes all the relative humidity sensors deployed in the wireless sensor network and \( S_2 \) denotes all the air temperature sensors deployed in the wireless sensor network. During a time period \( t \), \( S_{ij} \) has collected a set of sensor observations \( O^j_i = \left[ t_1, \ldots, t_g \right]^T \) where \( g \) is the total number of collected sensor observations.

The complete set of sensor observations collected by the \( S_i \) during the time \( t \) is expressed as \( O^j_i = \left( O^1_i, \ldots, O^n_i \right) \).

5.4.2 Dynamic Time Warping based Similarity Computation

In this study, Dynamic Time Warping (DTW) [162-164] is used to compute the similarity between two sensor observation data streams, as it is a well-established and widely used algorithm for comparing similarity between two discrete sequences of continuous values.

Suppose that two sensors \( S_{ij} \) and \( S_{ik} \) have been installed in the wireless sensor network, \( S_{ik} \in NH \left( S_{ij} \right) \) and \( S_{ij} \in NH \left( S_{ik} \right) \), where \( NH \left( s_n \right) \) indicates the neighbour sensor nodes of \( s_n \). During a short time period \( \Delta t \), both \( S_{ij} \) and \( S_{ik} \) respectively collected \( u \) sensor observations:

\[
O^\Delta i_{ij} = \left[ t_1, t_2, \ldots, t_u \right] \quad \text{and} \quad O^\Delta i_{ik} = \left[ t_1, t_2, \ldots, t_u \right],
\]

where \( t_1 < t_2 < \cdots < t_u \) are the timestamps at when sensor observations are collected, and \( o^\Delta i_{ij,1}, o^\Delta i_{ij,2}, \ldots, o^\Delta i_{ij,u} \) and \( o^\Delta i_{ik,1}, o^\Delta i_{ik,2}, \ldots, o^\Delta i_{ik,u} \) are the corresponding captured sensor observations. In order to efficiently compute the trend similarity \( \text{sim} \left( O^\Delta i_{ij}, O^\Delta i_{ik} \right) \) between \( O^\Delta i_{ij} \) and \( O^\Delta i_{ik} \), this study projects them onto a two dimensional Cartesian coordinate system where it treats time \( t_\lambda \) (\( 1 \leq \lambda \leq u \)) as an x-coordinate value and observations \( o^\lambda_{ij} \) (\( o^\lambda_{ik} \)) as a y-coordinate value. The difference \( \left( t_{\lambda+1} - t_\lambda, o^\lambda_{ij} - o^\lambda_{ik} \right) \) \( \left( t_{\lambda+1} - t_\lambda, o^\lambda_{ij} - o^\lambda_{ik} \right) \) between the two connecting successive points can be expressed as a vector \( v^\lambda_{ij} = \left( t_{\lambda+1} - t_\lambda, o^\lambda_{ij} - o^\lambda_{ik} \right) \) \( \left( v^\lambda_{ik} = \left( t_{\lambda+1} - t_\lambda, o^\lambda_{ij} - o^\lambda_{ik} \right) \right) \). Eventually, these two sensor observation data sets are respectively transformed into two sets of vector sequences:

\[
v^\lambda = \left( v^\lambda_1, \ldots, v^\lambda_a, \ldots, v^\lambda_u \right) \quad \text{and} \quad v^\Delta = \left( v^\Delta_1, \ldots, v^\Delta_b, \ldots, v^\Delta_{u-1} \right),
\]

where \( 1 \leq a \leq (u-1) \) and \( 1 \leq b \leq (u-1) \).
Using the DTW, a \((u-1) \times (u-1)\) matrix can be obtained, where element \((a,b)\) can be computed by Euclidean Distance between the end points of the two vectors or the angle \(\theta (0 \leq \theta \leq \pi)\) between two vectors [165]. However, the Euclidean Distance is not able to handle vertical shift that exists between the vectors under comparison. Compared with the Euclidean Distance, the angle not only considers the direction of the vectors, but also handles the vertical shift between the vectors. Thus, this work adopts the angle \(\theta\) between two vectors (e.g. \(v^u_a\) and \(v^u_b\)) to compute the degree of similarity. More specifically, the angle \(\theta(v^u_a, v^u_b)\) between vector \(v^u_a\) and \(v^u_b\) is defined as:

\[
\theta(v^u_a, v^u_b) = \cos \left( \frac{v^u_a \cdot v^u_b}{\|v^u_a\| \|v^u_b\|} \right)
\]

Then, an alignment between \(O^u_{ij}\) and \(O^u_{ik}\) can be represented by a warping path \(W = \{w_1, w_2, \cdots, w_l, \cdots, w_K\}\), where \(1 \leq l \leq K\) and \((u-1) \leq K \leq (2u-3)\). For each \(w_l\), it must satisfy three constraints: Boundary condition, Continuity condition and Monotonic condition [162, 166, 167]. In fact, many possible monotonical alignment paths from \((1,1)\) to \((u-1,u-1)\) can be generated to meet the three constraints. To determine an optimal warping path to minimize the cumulated distance \(D(a,b)\) between \(O^u_{ij}\) and \(O^u_{ik}\), a dynamic programming algorithm is an effective approach:

\[
D(1,1) = 0
\]

\[
D(a,b) = \min \{D(a-1,b-1), D(a-1,b), D(a,b-1)\} + \theta(v^u_a, v^u_b)
\]

With \(D(a,b)\), the following equation is used to calculate the similarity between \(O^u_{ij}\) and \(O^u_{ik}\) is:

\[
sim(O^u_{ij}, O^u_{ik}) = \begin{cases} 
0, & \text{if } \frac{D(u-1,u-1)}{K} > \frac{\pi}{2} \\
\cos \left( \frac{D(u-1,u-1)}{K} \right), & \text{otherwise}
\end{cases}
\]

5.4.3 Construction of Spatial Neighbourhood Matrix for Sensor Nodes

In this section, an approach for constructing a matrix \(U\) that documents the spatial neighbourhood relationship among sensor nodes deployed in the wireless sensor network is described. Formally, \(U\) can be expressed as:
\[
U = \begin{pmatrix}
    u_{11} & \cdots & u_{1n} \\
    \vdots & \ddots & \vdots \\
    u_{n1} & \cdots & u_{nn}
\end{pmatrix} \in \mathbb{R}^{n \times n}
\]

(5.5)

where \( u_{ij} \in \{0,1\} \) ( \( 0 \leq i \leq n \) and \( 0 \leq j \leq n \)). If \( u_{ij} = 1 \), it indicates that \( sn_j \in NH(sn_i) \). If \( u_{ij} = 0 \), it indicates that \( sn_j \notin NH(sn_i) \). The distance between \( sn_i \) and \( sn_j \) (each located at a precise latitude and longitude: \((lat_i, long_i)\) and \((lat_j, long_j)\)) is \( D(sn_i, sn_j) \) and \( \partial \) is a predefined neighbourhood threshold value. To build \( U \), if \( D(sn_i, sn_j) \leq \partial \), then \( u_{ij} = 1 \). If \( D(sn_i, sn_j) > \partial \), then \( u_{ij} = 0 \). Specifically,

\[
D(sn_i, sn_j) = 2rb
\]

(5.6)

where \( r = 6,378,137 \) is the diameter of the earth in meters according to WSG84 system and \( b \) is defined as:

\[
b = a \tan \left( 2 \left( \sqrt{d} \cdot \sqrt{1 - d} \right) \right)
\]

(5.7)

where \( d \) is defined as:

\[
d = \sin \left( \frac{\pi}{360} (lat_i - lat_j) \right) \cdot \sin \left( \frac{\pi}{360} (lat_i - lat_j) \right) + \cos \left( \frac{\pi}{180} lat_i \right) \cdot \cos \left( \frac{\pi}{180} lat_j \right) \cdot \sin \left( \frac{\pi}{360} (long_i - long_j) \right) \cdot \sin \left( \frac{\pi}{360} (long_i - long_j) \right)
\]

(5.8)

5.4.4 Construction of Spatial Neighbourhood Matrices for Sensors

In this section, an algorithm for constructing spatial neighbourhood matrices for sensors is detailed. For a type of sensors \( s_i = (s_{i1}, s_{i2}, \cdots, s_{in}) \) where \( 1 \leq i \leq m \), a corresponding spatial neighbourhood matrix \( A_i \in \mathbb{R}^{n \times n} \) will be constructed. There are \( m \) types of sensors in total, thus \( m \) matrices \( \{A_i\}_{i=1}^m \) will be constructed in total. Formally,

\[
A_i = E_i \cdot U = \begin{pmatrix}
    (a_{i1})_{11} & \cdots & (a_{in})_{1n} \\
    \vdots & \ddots & \vdots \\
    (a_{i1})_{n1} & \cdots & (a_{in})_{nn}
\end{pmatrix} \in \mathbb{R}^{n \times n}
\]

(5.9)
where \((a_{i,j})_{k} \in \{0,1\}\) (\(1 \leq k \leq n\)) describes the spatial neighbourhood relationship between two sensors \(s_{ij}\) and \(s_{ik}\). If \(s_{ik}\) and \(s_{ij}\) exist, and \(s_{ik} \in NH(s_{ij})\), then \((a_{i,j})_{k} = 1\). If either \(s_{ik}\) or \(s_{ij}\) does not exist, or \(s_{ik} \not\in NH(s_{ij})\), then \((a_{i,j})_{k} = 0\). In addition,

\[
E_i = (e_i)^T e_i = \begin{pmatrix}
(e_i)_1 \ast (e_i)_1 & \cdots & (e_i)_1 \ast (e_i)_n \\
\vdots & \ddots & \vdots \\
(e_i)_n \ast (e_i)_1 & \cdots & (e_i)_n \ast (e_i)_n
\end{pmatrix} = \begin{pmatrix}
(e_i')_{11} & \cdots & (e_i')_{1n} \\
\vdots & \ddots & \vdots \\
(e_i')_{n1} & \cdots & (e_i')_{nn}
\end{pmatrix} \in \mathbb{R}^{m \times n} \tag{5.10}
\]

where

\[
e_i = (e_i)_1, \ldots, (e_i)_j, \ldots, (e_i)_n) \in \mathbb{R}^{\text{json}} \tag{5.11}
\]

where \((e_i)_j \in \{0,1\}\) and \(1 \leq j \leq n\). If \((e_i)_j = 1\), it indicates that \(s_{ij}\) is employed in the wireless sensor network. If \((e_i)_j = 0\), it indicates that the sensor node \(sn_j\) did not have an \(i\)-th type of sensor installed on it.

### 5.4.5 DTW-based Similarity Matrices Construction and Suspicious Data Detection

It is given that the \(m\) type of sensors deployed in the wireless sensor network have collected a set of sensor observations \(\{O_{ij}\}_{i=1}^{m} = \{O_{ij}^{t_1}, \ldots, O_{ij}^{t_g}\}_{i=1}^{m}\) during a time period \(t\). If sensor \(s_{ij}\) exists, then

\[
O_{ij} = \begin{pmatrix}
t_1, \ldots, t_g \\
o_{ij}^{t_1}, \ldots, o_{ij}^{t_g}
\end{pmatrix}^T.
\]

If \(s_{ij}\) does not exist, then \(O_{ij} = []\). To detect suspicious data, firstly the DTW-based similarity matrices \(\{SM_i\}_{i=1}^{m}\) for the wireless sensor network needs to be constructed, where \(SM_i\) is a matrix describing all the similarity relationships between two elements of \(\{O_i^{t_1}, \ldots, O_i^{t_g}, \ldots, O_i^{t_n}\}\):

\[
SM_i = \begin{pmatrix}
(sm_i)_{11} & \cdots & (sm_i)_{1n} \\
\vdots & \ddots & \vdots \\
(sm_i)_{n1} & \cdots & (sm_i)_{nn}
\end{pmatrix} \tag{5.12}
\]
where \((sm)_j^k\) (\(1 \leq j \leq n\) and \(1 \leq k \leq n\)) describes the similarity trend between \(O^j_t\) and \(O^k_t\).

Specifically, if \((a_i)_j^k = 0\), then \((sm)_j^k = \emptyset\). If \((a_i)_j^k = 1\), then the \((sm)_j^k\) is represented as:

\[
(sm)_j^k = \left(\left(sm_j^k\right)_1, \ldots, \left(sm_j^k\right)_j, \ldots, \left(sm_j^k\right)_k\right) \in \mathbb{R}^{j \times k}
\]  

(5.13)

Where \(1 \leq \ell \leq \kappa\). To compute the \((sm)_j^k\), this study takes two sensor observation data sets \(O^j_t\) and \(O^k_t\) as input, then divides them into \(\kappa\) sliding windows, where \(\kappa = \frac{2g}{\eta} - 1\) (\(g\) is the number of sensor observations and \(\eta\) is an even integer and denotes the predefined size of a sliding window).

In addition, each window contains an overlap of \(\frac{\eta}{2}\) observations between consecutive windows.

Finally, the \(O^j_t\) and \(O^k_t\) are converted to:

\[
O^j_t = \left\{O^j_{\Delta t_1}, \ldots, O^j_{\Delta t_\kappa}\right\}
\]

(5.14)

\[
O^k_t = \left\{O^k_{\Delta t_1}, \ldots, O^k_{\Delta t_\kappa}\right\}
\]

(5.15)

where \(\Delta t_\ell\) indicates a short time period \([t_{(\ell-1)\eta+1}, t_{\ell\eta+1}]\), \(O^j_{\Delta t_\ell} = \left\{t_{(\ell-1)\eta+1}, \ldots, t_{\ell\eta+1}\right\}^T\) and \(O^k_{\Delta t_\ell} = \left\{o^j_{(\ell-1)\eta+1}, \ldots, o^j_{\ell\eta+1}\right\}^T\). The \(((sm)_j^k)_\ell\) is computed by Eq.5.4, where

\[
(((sm)_j^k)_\ell) = \text{sim}(O^j_{\Delta t_\ell}, O^k_{\Delta t_\ell}).
\]

Once \(\{SM\}_{j=1}^m\) is obtained, it can be readily used to detect suspicious data. To detect suspicious data, firstly this study needs to construct a set of suspicious matrices \(P = \{P_1, \ldots, P_m\}\), where \(P_i = \left\{(p_i)_1, \ldots, (p_i)_j, \ldots, (p_i)_n\right\}(1 \leq j \leq n)\). Specifically, if \(s_{ij}\) does not exist, then \((p_i)_j = \emptyset\). If \(s_{ij}\) does exist, then

\[
(p_i)_j = \left\{(p_i)_j, \ldots, (p_i)_j, \ldots, (p_i)_j\right\} \in \mathbb{R}^{j \times k}
\]

(5.16)
where \((p_{ij})_{j\ell} \in \{0, 1\}\) and \(1 \leq \ell \leq \kappa\). If \((p_{ij})_{j\ell} = 0\), it indicates that the segment of data streams \(O_{ij}^\ell\) is normal. If \((p_{ij})_{j\ell} = 1\), it indicates that the \(O_{ij}^\ell\) is suspicious. To compute the \((p_{ij})_{j\ell}\), the following function is used:

\[
(p_{ij})_{j\ell} = \begin{cases} 
0 & \text{if } \sum_{k=0}^{n} Z\left(\left(\frac{1}{2}\sum_{l=0}^{n} S\left(\left(sm_{ij}\right)_{l}\right)\right) \right) \\
1 & \text{otherwise}
\end{cases} \tag{5.17}
\]

Where \(Z\) and \(S\) are:

\[
Z(x) = \begin{cases} 
0, & x < \beta \\
1, & \text{otherwise}
\end{cases} \tag{5.18}
\]

\[
S(x) = \begin{cases} 
0, & \text{if } x = 0 \\
1, & \text{otherwise}
\end{cases} \tag{5.19}
\]

where \(\beta\) is the predefined similarity threshold value. The rationale for choosing the function in Eq.5.17 is that: if a data stream segment of sensor (property A) exists, and its pattern is similar to less than half of its neighbours’ patterns (for the same property A), then it is suspicious, otherwise it is normal.

### 5.4.6 Capturing Domain Expert Knowledge through Correlation Definition

To capture domain experts’ knowledge about relationships between properties, the experts are provided with the Protégé ontology editing tool and user interface to define sensor property correlations using terms from the CESP ontology. Figure 5.2 shows a screen shot of an expert using Protégé to create a correlation (cesp:air_temperature cesp:hasNegativeCorrelation cesp:relative_humidity) which specifies that air temperature has a negative (inverse) correlation on relative humidity. Figure 5.2 also shows that air temperature has a strong correlation with relative humidity (cesp:air_temperature cesp:hasStrongCorrelation cesp:relative_humidity). After the domain experts have defined the correlations between properties, this information is stored in the sensor property knowledge base (an RDF triple store).
Given these domain expert rules, this study can then construct a relationship matrix $Y$. Formally, $Y$ is defined as

$$Y = \begin{pmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mm} \end{pmatrix} \in \mathbb{R}^{m \times m} \quad (5.20)$$

where $y_{ii} \in \{0,1\}$ describes the correlation between $i$-th sensor property and $i'$-th sensor property, $1 \leq i \leq m$ and $1 \leq i' \leq m$. If the $i$-th sensor property has a specific correlation (e.g. strong correlation or medium correlation (defined by an expert)) with the $i'$-th sensor property, then $y_{ii} = 1$, otherwise $y_{ii} = 0$. In addition, if $i' = i$, then $y_{ii} = y_{i'i} = 0$. The value of $y_{ii}$ is obtained by submitting a SPARQL query to the sensor property knowledge base. Take the strong and medium correlation as an example, if $i$-th sensor property stands for the air temperature property, and $i'$-th sensor property stands for the relative humidity property, then the following SPARQL query is submitted to the sensor property knowledge base to calculate the value of $y_{ii}$:

```
ASK {

}
If the query returns true, then $y_{ii} = 1$. If the query returns false, then $y_{ii} = 0$. Once $Y$ is fully calculated, it can then be used to detect segment outliers and unusual events.

### 5.4.7 Detection of Segment Outliers and Unusual Events

Finally, rules are defined and applied to distinguish between segment outliers that are errors and genuine outliers (unusual events). For example: if a segment of observations of sensor property A is identified as suspicious data, and >50% of the corresponding segments of sensor properties that have a medium or strong correlation with A are also identified as suspicious data, then one infers that an unusual event has occurred. Otherwise, this segment of observations is an erroneous outlier.

Specifically, for each $(p_i)_j$ of $P = \{(p_1)_j, \ldots, (p_i)_j, \ldots, (p_m)_j\}$ of $\{p_{ij}\}_{i=1}^m$, if $(p_i)_j \neq []$, then one needs to determine whether the segment of observation $O_{\Delta t}^{ij}$ contains any outliers and unusual events.

If $(p_i)_j = 0$, $O_{\Delta t}^{ij}$ is a segment of normal data. If $(p_i)_j = 1$, then following function can be used to determine whether $O_{\Delta t}^{ij}$ indicates an erroneous outlier or an unusual event:

$$ r_{ij} = f(O_{\Delta t}^{ij}) = \begin{cases} R_1, & \text{if } C_1 \geq C_2 \frac{1}{2} \\ R_2, & \text{otherwise} \end{cases} \quad (R_1 \text{ indicates an unusual event})$$

where

$$ C_1 = \sum_{i=1}^{m} F\left((p_i)_j, y_{ii}\right) \quad (5.22) $$

$$ C_2 = \sum_{i=1}^{m} F_{R}\left((p_i)_j, y_{ii}\right) \quad (5.23) $$

where

$$ F(x, y) = \begin{cases} 1, & \text{if } x=1 \text{ and } y=1 \\ 0, & \text{otherwise} \end{cases} \quad (5.24) $$

and
\[ F_R(x, y) = \begin{cases} 1, & \text{if } x \text{ is not null, } y=1 \\ 0, & \text{otherwise} \end{cases} \] (5.25)

Therefore, for each observation \( O^j_t \) a decision matrix \( r^j_t = (r^{j1}_t, \ldots, r^{j\mu}_t, \ldots, r^{j\zeta}_t) \) will be constructed. If \( r^{j\mu}_t = R_1 \) then \( O_{\Delta t}^j \) indicates an unusual events. If \( r^{j\mu}_t = R_2 \), then \( O_{\Delta t}^j \) is an erroneous segment outlier.

Finally, a decision matrix \( \{R_i\}_{i=1}^m \) with \( R_i = (r^{i1}_t, \ldots, r^{i\mu}_t) \) is constructed. In addition, the computational complexity for detecting segment outliers and unusual events is \( O(m^*n^2) \).

### 5.5 Implementation

#### 5.5.1 System Architecture

Figure 5.3 provides an overview of the architecture of the SOUE-Detector system. As described in the common technical Framework (Figure 1.1, Chapter 1), the system utilizes the PostgreSQL object-relational database management system for storing sensor observations and the open source Java framework Sesame, an RDF triple repository, for storing sensor property knowledge base. The SOUE-Detector Web portal provides a Web interface to enable users to search for sensor observations for a particular time period, and view the corresponding segment outliers and unusual events. The server component is built using JSP and Java. The server interfaces with users through a Web browser (e.g., Google Chrome), a Google Earth map interface and a Google line chart interface, enabling spatial-temporal searching and visualization across the data.

In addition, the PostgreSQL database stores the wireless sensor network configurations and associated sensor node matrix and sensor matrices, which document the spatial neighbourhood relationships among sensor nodes and sensors, respectively. Date/time stamps are also recorded with these matrices and are updated/recalculated whenever the wireless sensor network configuration is changed. Whenever the system receives a sensor network modification message, the system attaches end date/time stamps to the previously generated sensor node and sensor matrices; then, the system recalculates the new sensor node and sensor matrices and saves them in the database with the new current start date/time stamps. Past matrices are not deleted because they are relevant for the processing of historical archived data streams.
5.5.2 Web Portal and User Interface

A Web Portal was developed to provide access to the wireless sensor database, RDF triple store and associated services. Figure 5.4 illustrates screenshots of the search interface, the Google Earth interface and the timeline interface. Users are able to specify a period of interest. The datasets within this time range are retrieved and the proposed detection algorithm is applied to the search results to detect segment outliers. The detection results are displayed in a visualization timeline interface. The Google Earth map visualization interface uses different sensor icons to represent each sensor’s status. For example, a blue icon indicates a sensor with normal data, a yellow icon indicates a sensor with segment outliers, and an orange icon indicates a sensor with unusual events. The timeline visualization interface uses different colours to highlight: normal data (grey), erroneous segment outliers (red) and unusual events (royal blue).

The LHS of Figure 5.4 illustrates an example with erroneous segment outliers. The user specified a time period between 2011-06-04 00:00:00 and 2011-06-04 16:00:00. The system retrieved the data streams for this period for all sensors/sensor nodes, and then applied the proposed detection algorithm to the search results. The system detects outliers in the data from sensor node 1 (a relative humidity and an air temperature sensor). The detection results are displayed in a Google Map interface in Figure 5.4 (a-1). Figure 5.4 (a-2) shows the sensor data collected from sensor 1-relative humidity, displayed within a timeline with segment outliers marked in red. Clicking the “Show Neighbourhood Sensors”
button, retrieves data from the neighbourhood sensors, including relative humidity sensors 26, 143, and 210, and displays these data streams as well (See Figure 5.4 (a-2)). Figure 5.4 (a-3) shows the detection results for sensor 1-air temperature and its neighbourhood sensors’ collections. It is known that the relative humidity property is strongly correlated with the air temperature. By comparing Figure 5.4 (a-2)) and Figure 5.4 (a-3), the user can see that:

- Relative humidity sensor 1 differs from the pattern displayed by its neighbouring relative humidity sensors (26, 143, 210) (which are all similar) – so it is identified as a true error;
- The pattern of air temperature sensor 1 is similar to its neighbouring air temperature sensors 143, 210 and 26), so it is identified as normal data.

The RHS of Figure 5.4 illustrates an example of unusual event detection. The user specified a time period between 2011-06-25 00:00:00 and 2011-06-25 10:00:00. The detection results are presented in Figure 5.4 (b-1), (b-2), and (b-3). Figure 5.4 (b-1) shows that some unusual event occurred at sensor node 10. Selecting relative humidity sensor 10 and air temperature sensor 10 (and their neighbouring sensors), generates the results displayed in Figure 5.4 (b-2) and Figure 5.4 (b-3). The unusual events are marked in blue. Figure 5.4 shows the following:

- Figure 5.4 (b-2) reveals that the pattern for relative humidity sensor 10 is different from the patterns for its neighbouring sensors (1, 26 and 143), which are all similar;
- Figure 5.4 (b-3) reveals that the pattern for air temperature sensor 10 is different from the patterns for its neighbouring sensors (1, 26 and 143), which are all similar.

These results indicate that the relative humidity and air temperature both changed simultaneously at sensor node 10. Therefore, it can be assumed that an unusual event, as opposed to a true error, occurred at node 10.
(a-1): Searching data and representing detection results in Google Earth
(b-1): Searching data and representing detection results in Google Earth

(a-2): Timeline user interface with segment outliers
(b-2): Timeline user interface with unusual events
5.6 Experimental Evaluations

5.6.1 Evaluation Metrics

To evaluate the system, data collected from 36 sensor nodes deployed in the Springbrook project [156] as shown in Figure 5.5, is used. For each sensor node, three different sensors are deployed: air temperature, relative humidity, and air pressure. The sampling rate is one sample every 10 minutes. Because this project does not have ground truth information for segment outliers and unusual events for the real datasets, this study generated eight test datasets by adding segment outliers and unusual events to a cleaned version of the dataset that does not contain any abnormal data for the period 2011-08-18 to 2011-09-18.

The first test dataset is created by adding erroneous segment outliers to the cleaned dataset. The generation process comprises two steps:

1. Firstly, eight air temperature data streams that do not overlap temporally were randomly selected. Next, for each selected data stream, a segment with 12 observations was chosen and replaced with an air temperature segment outlier, generated by randomly choosing 12 values.
(between 0 and 25 °C). The relative humidity and air pressure data streams were unchanged. Next, this first step was repeated 15 times across the duration of the segment (one month).

2. Secondly, eight air humidity data streams that do not overlap temporally were randomly selected. For each selected data stream, a segment with 12 observations was chosen and replaced with a relative humidity segment outlier generated by randomly choosing 12 values (between 40 and 100%). The air temperature and air pressure data streams were unchanged. Next, this second step was repeated 15 times across the duration of the segment (one month).

The first test dataset contains 155,520 relative humidity observations, 155,520 air temperature observations, 155,520 air pressure observations, and 720 segment outliers.

The remaining seven test datasets were created by adding unusual events with seven types of correlations to the cleaned data sets, including: cesp:hasStrongCorrelation, cesp:hasMediumCorrelation, cesp:hasPositiveCorrelation, cesp:hasLinearCorrelation, cesp:hasCurviLinearCorrelation, cesp:hasSimpleCorrelation and cesp:hasComplexCorrelation. Specifically, to create each test dataset, eight sensor nodes were randomly selected. For each sensor node, the relative humidity and air temperature data streams were selected. Next, a 2-hour time period was chosen, and the relative humidity and the air temperature segment data streams collected during this given 2-hour time period were replaced with 12 artificially generated relative humidity values and 12 artificially generated air temperature values. The air pressure data streams were unchanged. The above steps outlined above were repeated 30 times across the duration of the test data sets. At the conclusion, each test dataset contained 155,520 relative humidity observations, 155,520 air temperature observations and 155,520 air pressure observations, and 1,440 unusual events with a specific type of correlation (TD2 cesp:hasStrongCorrelation (test dataset 2 contained unusual events with the cesp:hasStrongCorrelation correlation), TD3 cesp:hasMediumCorrelation, TD4 cesp:hasPositiveCorrelation, TD5 cesp:hasLinearCorrelation, TD6 cesp:hasCurviLinearCorrelation, TD7 cesp:hasSimpleCorrelation and TD8 cesp:hasComplexCorrelation).
5.6.2 Evaluation Results and Discussions

In the proposed detection algorithm, a similarity threshold value was used to decide whether two segment sensor observations are similar or not. If the similarity value is greater than or equal to the similarity threshold value, they are regarded as similar. For both experiments, the set values were $\vartheta = 300\text{ m}$ (predefined neighbourhood threshold value) and $\eta = 12$ (predefined sliding window length). $\vartheta = 300$ guarantees that every sensor has at least three neighbourhood sensors and a window of 12 samples has an optimum potential to capture segment outliers and unusual events. Figure 5.6 shows the precision and recall for detecting erroneous segments from test data set 1, while Figures 5.7 to 5.13 show the precision and recall for detecting unusual events from test data set 2 to test data set 8. The results from Fig.5.6 show that the recall is very high (87.5-100%) when using a similarity threshold greater than 80%. However, the precision of detecting segment outliers decreases when the similarity threshold increases above 90%. The results from Figures 5.7 to 5.13 reveal that the recall for detecting unusual events is very high (81.46-100%) when using a similarity threshold greater than 84%. However, the precision for detecting unusual events decreases significantly as the similarity threshold rises above 90%. Overall, when the similarity threshold value is set to between 84% and 90%, the detection algorithm for detecting unusual events performs best (recall >81.46% and precision>80%).

In the evaluation, the performance of both the outlier and unusual event detection methods were evaluated using similarity threshold values $\beta$ in the range of 72-98%. The evaluation results show that the performance of the proposed approach is very sensitive to the similarity threshold values.
More specifically, the results indicate that setting the similarity threshold value to 90% gains the best performance (precision=96.87% and recall=96.88%) for erroneous outlier detection, while setting the similarity threshold value to 88% gains the best performance (precision in the range of 91-93.7% and recall in the range of 88.75-91.05%) for unusual event detection. More generally, in order to obtain optimum performance, similarity threshold values should lie in the range 84-90%. Overall, the evaluation results reveal that the proposed approach is able to efficiently and accurately detect both erroneous outliers and unusual events by making use of sensor data trend similarities and correlations between sensor properties. In other words, such evaluation results reveal that integrating Semantic Web technologies and statistical algorithms with domain expert knowledge about sensor property correlations can improve the detection of outliers and unusual events within sensor data wireless sensor data streams.

![Figure 5.6: Precision and recall for detecting erroneous segment outliers](image1)

![Figure 5.7: Precision and recall for detecting unusual events with cesp:hasStrongCorrelation](image2)
Figure 5.8: Precision and recall for detecting unusual events with cesp:hasMediumCorrelation

Figure 5.9: Precision and recall for detecting unusual events with cesp:hasPositiveCorrelation

Figure 5.10: Precision and recall for detecting unusual events with cesp:hasLinearCorrelation
Figure 5.11: Precision and recall for detecting unusual events with \textit{cesp:hasCurvilinearCorrelation}

Figure 5.12: Precision and recall for detecting unusual events with \textit{cesp:hasSimpleCorrelation}

Figure 5.13: Precision and recall for detecting unusual events with \textit{cesp:hasComplexCorrelation}
5.6.3 Comparison of the SOUE-Detector with Previous Outlier Detection Techniques

In order to evaluate the proposed SOUE-Detector approach, its performance is compared with two alternative outlier detection techniques: (i) a computational intelligence technique that uses an artificial neural network (ANN) algorithm [168]; and (ii) a data mining technique that uses an SVM algorithm [112]. Table 5.1 shows that the SOUE-Detector approach (Precision = 96.87% and Recall=96.88%) outperforms ANN (Precision=89.72% and Recall=91.83%) and SVM (Precision =93.58% and Recall=86.92%).

Table 5.1 : Comparison of different outlier detection techniques

<table>
<thead>
<tr>
<th>Technique Category</th>
<th>Techniques</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Intelligence</td>
<td>Artificial Neural Network</td>
<td>89.72</td>
<td>91.83</td>
</tr>
<tr>
<td>Data Mining</td>
<td>Support Vector Machine</td>
<td>93.58</td>
<td>86.92</td>
</tr>
<tr>
<td>SOUE-Detector</td>
<td>Semantic Rules</td>
<td><strong>96.87</strong></td>
<td><strong>96.88</strong></td>
</tr>
</tbody>
</table>

5.7 Summary

In this chapter, a SOUE-Detector system has been developed that can efficiently detect genuine outliers and unusual events for real wireless sensor network data streams by combining DTW statistical analysis with domain expert knowledge and Semantic Web-based rules. In addition, the Web-based visualization interfaces enable scientists to quickly explore and easily understand the quality of their collected data streams and sources of problematic data streams. This provides useful information for scientists that enables them to adjust their wireless sensor network configurations to collect more accurate data streams. In addition, the Protégé user interface enables domain experts to record their knowledge about sensor property correlations that can be used to distinguish between erroneous outliers and events. Compared with previous related work, the proposed approach has the following advantages. Firstly, it takes into account multivariate sensor data and the relations between the variables. Secondly, it combines domain expert knowledge with statistical analysis to generate improved precision and recall. Thirdly, this approach considers other types of correlation other than just spatio-temporal correlations. Lastly, the proposed approach handles the challenges associated with changes to wireless sensor network configurations over time.

The next chapter focusses on the last topic: semantic reasoning over environmental wireless sensor data streams to infer high-level knowledge. More specifically, it describes how a standardized rule-based reasoning approach can be used to infer Fire Weather Indices (i.e., fire danger ratings) from the cleaned environmental sensor data streams generated by the Springbrook region of South East Queensland.

6.1 Overview

Bushfires have been responsible for some of the most devastating natural disasters in Australia and are estimated to cause damage with an average annual cost of $77 millions [169]. Fire weather indices play a significant role in issuing warnings and in estimating the potential danger associated with predicted wild fires [129]. The two most widely used and accepted systems are the McArthur Forest Fire Danger Index (used in Australia) and the Canadian Fire Weather Index (used in North America and the Australia Bureau of Meteorology (BoM)) [129]. Indices are calculated by combining three weather parameters: wind speed, relative humidity, and air temperature. Although these two fire weather index systems are the most robust and widely adopted, current implementations have some limitations. For example, most existing implementations use weather parameters collected from widely distributed sensor nodes, tens of kilometres apart. Hence, the collected data is not dense enough to estimate an accurate fire weather index for a specific region. Moreover, the fire danger maps are typically only updated once per day, which means hourly variations in bushfire risk over the course of a 24-hour are not possible. For example, the Canadian Fire Weather Index takes the noon Local Standard Time (LST) values of weather parameters as input. The potential cost of imprecise information can be significant when making decisions associated with hazard evacuation plans and fire-fighting operations. Therefore, there is an urgent need to develop more accurate methods for estimating fire weather indices at higher spatio-temporal resolutions.

In recent years, the number of wireless sensor networks deployed in different environments has rapidly expanded due to the decreasing cost and size and increasing reliability of micro-sensor technologies. Sensors are being used to monitor the safety and security of buildings and spaces [170, 171], to measure humans’ physical, physiological, psychological, cognitive and behavioural processes [172], and to capture observations and measurements of the environment parameters [156]. In recent years, wireless sensor networks consisting of coordinated autonomous sensors have been deployed to monitor forest physical parameters (air temperature, relative humidity, wind speed, leaf wetness, air pressure, wind direction, solar radiation and so on) [156, 173, 174]. As a result, an avalanche of raw sensor network data streams about forest environments has been collected which...
provides a valuable research platform for scientists or researchers to study or understand the microclimate and associated fire weather indices within a focussed area.

However, estimating fire weather indices from wireless sensor network data streams is a very challenging problem. Firstly, wireless sensor network data streams are often incomplete or imprecise due to the fading signal strength, hazard node faults, inaccuracies of measurement, and limited energy and wireless bandwidth [15, 175]. Moreover, the large volumes of complex, numerical and unstructured sensor data streams generated a major challenge to processing in real-time or near-real-time. In addition, heterogeneous, non-standard infrastructure, and poor data representation have resulted in many sensor data streams being locked inside specific proprietary applications and inaccessible to the wider community.

The objectives of the work described here are to improve the quality of the data (in terms of reliability, precision and resolution), as well as the method for deriving environmental indicators i.e., fire weather indices from wireless sensor network data streams. The objective of this study is to apply Semantic Web approaches to this problem by implementing and evaluating a rules-based approach that enables us to harness domain expert knowledge to improve the precision of fire weather index calculations. For example, a bush fire hazard expert specified the following general rule for a particular region near Springbrook: when temperature is greater than 32 degree Celsius, and the wind speed is greater than 25 m/s and relative humidity is less than 50%, it indicates an Extreme fire weather index.

The remainder of the chapter is structured as follows. Section 6.2 presents the proposed methodology, including the case study and methodological steps. Section 6.3 describes the data processing steps and storage, which includes detecting and removing outliers, annotating data streams with terms from a set of OWL ontologies, developing an FWI ontology, and storing the RDF triples in optimized RDF storage. Section 6.4 describes how meteorologists’ knowledge is combined with semantic reasoning technology to infer accurate fire weather indices. Section 6.5 provides details about the system’s technical architecture, functionality, and the user interface. Section 6.6 describes the evaluation process and evaluation results. Lastly, a conclusion is drawn in Section 6.7.

6.2 Methodology

6.2.1 Case Study

The Springbrook National Park is one of Queensland’s five World Heritage listed areas and covers 6,197 hectares restored from agricultural grassland to native rainforest vegetation. Located about 96 kilometres south of Brisbane in the state of Queensland, it is a place of exceptional natural beauty
and ecological importance. The Springbrook Wireless Sensor Network project [176] (Case Study #2 in Section 1.2.2) being undertaken by the CSIRO, Queensland Department of Environment and Resource Management, and Australia Rainforest Conservation Society has employed more than 180 sensor nodes attached to several hundred solar-powered sensor devices in the Springbrook National Park region (bounded by 28° 14’ 1512” – 28° 13’ 138”S latitude, and 153° 15' 60” –153° 16 ‘43”E longitude) (see Figure 6.1). This tagged region is an area of approximately 0.13 square miles.

More specifically, in this project CSIRO scientists installed different types of sensor devices to monitor physical parameters, such as air temperature, relative humidity, wind speed, leaf wetness, soil water potential, total solar radiation, and wind direction. The Springbrook sensor data set is ideal for this research because it records those parameters specifically required to calculate fire weather indices. For this study, two and half years’ of sensor data streams (from January 2010 to June 2012) were collected from the Springbrook Wireless Sensor Network project [176]. The collected datasets contain three types of weather variables: air temperature, wind speed, and relative humidity, captured by Vaisla WXT520 weather transmitters. Each transmitter is sampled at 10-minute intervals.
6.2.2 Methodological Steps

The proposed methodology for undertaking this research can be sub-divided into the following seven steps described below:

1. Firstly, two and half years of wireless sensor network data streams were harvested from the Springbrook Wireless Sensor Network project. This dataset is used to evaluate the performance of the proposed approach. In addition, a comprehensive set (241) of First-Order-Logic inference rules for calculating fire weather indices were collected from experts from the meteorological domain.

2. A data pre-processing step was applied to detect and remove the outliers within the collected data streams. Then the cleaned data streams were converted into RDF triples which are stored in the optimized repository (multiple RDF triple stores).

3. An OWL ontology for describing wireless sensor networks sensor data streams as well as the fire weather indices (FWI) and input parameters was developed, which involved extending and refining the existing (SSN) ontology [49].

4. The 241 First-Order-Logic rules (captured in Step 1 above) were converted to SPARQL inference rules by using the defined FWI ontology and other OWL ontologies. The SPARQL inference rules are saved in the repository.

5. A rules-based inferencing algorithm is applied to generate fire weather indices and an Inverse Distance Weighting-based neighbourhood region prediction algorithm was developed and applied to calculate more accurate raster-based fire weather indices for a specific region at a time period (from point data).

6. A Web-based search interface was developed to enable users to search fire weather indices for a specific region within a time period. A visualization interface consisting of Google Earth, timeline and pie charts was developed. The Google Earth map displays animated fire weather index search results, while the pie charts are used to compare entire periods, daytime and night-time average fire weather indices.

7. Finally, three aspects of the system were evaluated. The first evaluation involves a comparison of the quality of the results generated from the proposed system with the McArthur Forest Fire Danger Index and the BoM Daily Fire Weather Index values. The second evaluation involves the assessment and optimization of SPARQL querying and RDF triple store configuration to obtain scalable performance. The third evaluation involves the assessment of the SFWI system usability by acquiring user feedback from eight users.
6.3 Data Processing and Storage

6.3.1 Data Pre-processing

Data quality is a significant issue associated with wireless sensor networks due to the limited resources available (power, memory, computational capacity and communication bandwidth), and the harsh environmental conditions [111, 177-179]. In particular, outliers can significantly affect the accuracy of the fire weather index calculations. Hence, data pre-processing is critical to detect and remove outliers from the raw sensor data streams. To detect outliers, this study takes advantage of the fact that air temperature, humidity and wind speed readings from sensors geographically close to each other, follow the same patterns [178, 180]. More specifically, the outlier detection approach proposed in the previous chapter (Chapter 5) is used to detect and filter the outliers within the raw sensor data streams to generate cleaned wireless sensor data streams. In addition, the following parameters are used: predefined neighbourhood threshold value ($\delta = 300$ m), predefined sliding window length ($\eta = 12$) and the similarity threshold value ($\beta = 90\%$).

6.3.2 Conversion of Cleaned Sensor Observations to RDF Triples

To date, there exist a wide variety of existing sensor and observation ontologies to describe sensor networks, sensor devices and sensor observations with machine-interoperable semantics. Such ontologies include the Semantic Sensor Network Ontology (SSN) [50], and OntoSensor (Russomanno, Kothari et al. 2005). The SSN ontology, developed by the W3C Semantic Sensor Network Incubator group, is the most comprehensive ontology and provides the top-level classes and properties for representing sensors, the measurement capabilities of sensors, and the sensor observations. More specifically, the SSN has defined the core concepts and relations (sensors, features, properties, observations and systems) and has been aligned to the DOLCE-Ultra Lite (DUL) ontology [181, 182]. In addition, a number of groups have developed extensions to SSN that this thesis can leverage. The W3C Semantic Sensor Network Incubator group developed an Automatic Weather Station (AWS) ontology [57] to specify different sensor types, including $aws$:TemperatureSensor, $aws$:WindSensor and $aws$:HumiditySensor. The W3C Semantic Sensor Network Incubator group also developed a Climate and Forecast (CF) ontology [183] that defines climate data variables, such as $cf$:air_temperature, $cf$:relative_humidity and $cf$:wind_speed. NASA developed a unit ontology [184] that defines vocabularies for physical properties and corresponding units of measurements, such as $unit$:degreeCelsius, $unit$:metrePerSecond and $unit$:percent etc. Figure 6.2 illustrates how these ontologies are combined to describe sensor data streams for the Fire Weather Index application described here.
Before SPARQL inferencing can be applied to estimate fire weather indices, the cleaned sensor observations (formatted in CSV files) were converted to RDF triples. The combined SSN, AWS, DUL, and Unit ontologies were employed to convert the CSV formatted sensor observations to RDF triples. For instance, suppose that there exists a temperature observation: ‘2012-01-02 03:50:00, air_temperature, AT_1, SN_1, 13.5, °C’. Parsing of this string is performed as follows: 2012-01-02 03:50:00 is the time when an observation was collected; air_temperature indicates the property of the observation; AT_1 is the ID of the sensor device; SN_1 is the ID of the sensor node that the sensor was attached to; 13.5 is the observation value; and the °C is the observation unit. After parsing, this observation is converted to RDF triples with spatial, temporal and semantic metadata (as shown in Figure 6.2).

![Figure 6.2: Instance of a temperature observation represented using the SSN, AWS, DUL, CF and Unit ontologies](image)

### 6.3.3 Fire Weather Index Ontology

A Fire Weather Index (FWI) ontology was developed to define fire weather index classes and relative properties. Fire weather indices can be categorized into five high level categories: fwi:Low, fwi:Moderate, fwi:High, fwi:VeryHigh and fwi:Extreme. The definitions of the five categories are as follows:

- **fwi:Low**: fires can be easily controlled and there will be no risk to life or forest.
- **fwi:Moderate**: fires can be easily controlled but still present a threat.
- **fwi:High**: fires can be controlled but present a threat.
- **fwi:VeryHigh**: fires can be difficult to control and present a real threat
• *fwi:Extreme*: fires will likely be uncontrollable and fast moving with flames that may be higher than roof tops.

Moreover, these five high level classes can be further sub-divided into 15 subclasses: *(fwi:Min-Low, fwi:Mid-Low, fwi:Max-Low), (fwi:Min-Moderate, fwi:Mid-Moderate, fwi:Max-Moderate), (fwi:Min-High, fwi:Mid-High, fwi:Max-High), (fwi:Min-VeryHigh, fwi:Mid-VeryHigh, fwi:Max-VeryHigh), (fwi:Min-Extreme, fwi:Mid-Extreme, and fwi:Max-Extreme)*.

The FWI ontology was also aligned to the Provenance (PROV) ontology [185], that provides a set of classes, properties, and restrictions to represent related provenance information. A “fwi:FireEvent_1” is a *prov:Activity* and has two properties, *prov:atLocation* and *prov:atTime* that record the associated place and time for each FireEvent. Figure 6.3 illustrates how the FWI ontology can be applied to describe a high fire weather event that was observed at Sensor Node 2.

Figure 6.3: Applying the FWI ontology to describe the calculation of a high fire weather index from three sensors (WindSpeed WS_2, RelativeHumidity RH_2, AirTemperature AT_2) on SN-Node2

6.3.4 Storage Techniques

It is widely acknowledged that RDF graph-based triple stores are not very efficient in terms of query and reasoning performance, thus this study proposes a multiple repository storage approach to improve the query and reasoning performance of RDF graph based triple stores. This multiple repository storage consists of an RDF graph catalogue repository, and a set of RDF graph sub-repositories. The catalogue repository is used to store catalogue information about the sub-repositories (e.g., sub-repositoryID, sub-repository graphs, graph maximum Time, graph minimum Time, sub-repository data type etc.). There are four sub-repositories for the four different data types: relative humidity, air temperature, wind speed and a FWI repository. The relative humidity repository, wind speed repository and air temperature repository are used to store relative humidity RDF graphs,
wind speed RDF graphs, and air temperature RDF graphs, respectively. When a user uploads a set of raw sensor observations to the system, the system firstly identifies and filters erroneous segments and then converts the data streams to RDF graphs. Next, the system automatically generates a unique ID for each graph (that defines the place/time/indicator context). Based on the uploaded data type, the system chooses a sub-repository to save this RDF graph. If the RDF graph is successfully uploaded to the repository, the system saves the corresponding graph storage information (context, maxTime and minTime, repositoryID) to the catalogue repository to enable efficient searching and retrieval. Figure 6.4 illustrates how raw sensor observations are saved to each corresponding repository. The FWI repository is used to store the SPARQL inferencing results that calculate the fire weather indices. Detailed information describing the storage of inference results is explained in section 6.4.2.

<table>
<thead>
<tr>
<th>Algorithm 1: Storing sensor observations to the multiple repository triple store</th>
</tr>
</thead>
</table>
| **Input:**  
| O indicates one type of sensor observation data set  
| TP indicates the type of the sensor observation  
| **Output:** true or false. (If storage is successful, then return true. If storage failed, then return false)  
| O1=preprocessing (O); //remove the outlier from the sensor observations  
| RDFGraph=Conversion(O1); //convert the cleaned sensor observations to RDF graph  
| Context=getContext(RDFGraph); //generate a context for the generated RDF graph  
| RepositoryId=getRepositoryId(TP); //look for the target repository Id  
| R= SaveToRepository(RepositoryId, RDFGraph, Context); //save the RDF graphs to the corresponding repository  
| If R==true, then  
| maxTime = getMaxTime(RDFGraph); //get the maximum sampling time  
| minTime = getMinTime(RDFGraph); //get the minimum sampling time  
| UpdateCatalog(RepositoryId, Context, maxTime, minTime); //update the provenance information of the RDF graphs to the catalogue repository  
| Return true;  
| else  
| Return false;  
| end |

Figure 6.4: Algorithm for saving sensor observations to the Multiple Repository RDF triple store

6.4 Semantic Inferencing

6.4.1 Defining SPARQL Inference Rules

Inferencing is a mechanism by which a set of rules that represent domain expert knowledge are used to logically derive additional domain specific knowledge. SPARQL [186] is a standard RDF query
language recommended by the World Wide Web Consortium (W3C) and considered as an essential graph-matching query language. More specifically, a SPARQL query consists of a body, a complex RDF graph pattern matching expression (which may include basic graph patterns, group graph patterns, optimal graph patterns, alternative graph patterns and patterns on named graphs), and a head, an expression that indicates how to construct an answer to the query [187]. More specially, SPARQL has four types of query forms (Select, Construct, Ask and Describe) to define four types of result formats.

SPARQL can be used for inferencing because the SPARQL Construct query form returns an RDF graph that is formed by taking each query solution in sequence, substituting for the variables in the graph pattern, and combining the triples into a single RDF graph. In other words, the Construct query form can derive new triples when the graph patterns match. The returned RDF graph can be directly inserted into a repository by using the Insert graph update operation. Consider the following example using a SPARQL Construct query form and Insert graph update operation to infer a fire weather index value.

A total of 241 fire weather index rules were collected from collaborating meteorologists. These rules were represented as SPARQL queries and then stored in the multiple repository storage. For example, one of the rules is: ‘if relative humidity $\geq$ 80% AND 17.5 m/s $\leq$ wind speed $\leq$ 24.4 m/s AND 32 °C $\leq$ air temperature $\leq$ 41 °C, for a given location at time T, then at time T this location has a High fire weather index.’ This rule statement can be interpreted into a SPARQL rule expressed as follow:

$$\text{Construct} \{ \ ?\text{FireEvent}_1 \ \text{prov:atLocation} \ \text{?node}. \ ?\text{FireEvent}_1 \ \text{prov:atTime} \ ?T. \ ?\text{FireEvent}_1 \ \text{rdf:type} \ \text{fwi:High}. \}$$

$$\text{Where} \{ \ ?\text{RH}_\text{OB1} \ \text{ssn:ObservedProperty} \ \text{cf:relative_humidity}. \ ?\text{RH}_\text{OB1} \ \text{ssn:ObservationSamplingTime} \ ?T. \ ?\text{RH}_\text{OB1} \ \text{dul:UnitOfMeasure} \ \text{unit:percent}. \ ?\text{RH}_\text{OB1} \ \text{dul:hasDataValue} \ ?\text{RH}_\text{OB1V}. \ ?\text{RH}_\text{OB1} \ \text{ssn:ObservedBy} \ ?\text{RH\_Sensor1}. \ ?\text{RH\_Sensor1} \ \text{ssn:deployedOnPlatform} \ ?\text{node}. \ ?\text{WS}_\text{OB1} \ \text{ssn:ObservedProperty} \ \text{cf:wind_speed}. \ ?\text{WS}_\text{OB1} \ \text{ssn:ObservationSamplingTime} \ ?T. \}$$
FILTER( ?RH_OB1 \(>=\) 80 AND \(<=\) 100 AND ?WS_OB1 \(>=\) 17.5 AND \(<=\) 24.4 AND ?AT_OB1 \(>=\) 32 AND \(<=\) 41 )}

Figure 6.5: An example of applying SPARQL reasoning to an RDF dataset to infer a High fire weather index.
Suppose that three types of sensor observations (relative humidity: 85%, wind speed: 23.3 m/s, and air temperature: 40°C) were collected from the sensor node 1 at 2012-01-02T12:00:00, then these sensor observations can be expressed in RDF format as shown in Figure 6.5. When the above inference rule is applied to this data set, then it generates new triples (shown in Figure 6.5 shaded grey), which are the inference results that are saved to the FWI Repository.

### 6.4.2 Inferencing Algorithm

The Canadian Forest Fire Weather Index system takes Noon Stand Time weather parameters as input. Hence, it only produces the daily fire weather indices at noon local time, which limits its accuracy and usefulness in decision-making. The methodology used here (Figure 6.6) enables users to query fire weather indices in real time and at 10-minute intervals. Moreover, the proposed system only performs the inferencing operation when it receives a query from a user. When users input their query time parameters to the system, the system searches the FWI Repository that stores all the historical inferred fire weather indices.

```plaintext
Algorithm 2: Inferring fire weather index

<table>
<thead>
<tr>
<th>Input:</th>
<th>[t₁, t₂] – a period of time where t₁ &gt; t₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>Fire Weather Indices (FWIs) within the period [t₁, t₂]</td>
</tr>
</tbody>
</table>

hasAll=CheckFW(FWIRepository, t₁, t₂);
If hasAll==false, then
  timeRanges=getMissedTimeRanges(FWIRepository, t₁, t₂);
  temporaryRepository=createATemporaryRepository();
  repositoryIds={airTemperatureRepold, windSpeedRepold, relativeHumidityRepold};
  For each repositoryId in the repositoryIds list, do
    graphIds=getRDFGraphsStorageInfo(CatalogRepository, repositoryId, timeRanges);
    RDFTriples=getRDFTriplesFromRepo(repositoryId, graphIds, timeRanges);
    SaveRDFTriplesToTemporaryRepository(RDFTriples);
  end For
  newFWIs=inferringFWIsAtTemporaryRepository(SPARQLRules, temporaryRepository);
  SaveFWIs(FWIRepository, newFWIs);
  destoryTemporaryRepository(temporaryRepository);
end If
FWIs=searchFWIs(FWIRepository, t₁, t₂);
Return FWIs;
```

**Figure 6.6: Inference algorithm for inferring new FWIs**
If the *FWI Repository* contains the requested fire weather indices then it returns them. If not, the system retrieves the time range information for the missing fire weather indices and creates a temporary RDF graph repository as a data hub to estimate the fire weather indices. Next, the system visits all the sub-repositories to retrieve and upload related sensor data combinations to the temporary repository. Then, a SPARQL inference operation applies the inferencing rules to infer all the fire weather indices. Finally, the inferred fire weather indices are saved in the *FWI Repository* and the requested fire weather index results are presented to the user.

### 6.4.3 IDW-Based Neighbourhood Region Prediction

Chapter 5 and section 6.3.1 described the pre-processing step to remove outliers within the sensor data streams. However, data loss often occurs in wireless sensor networks due to random link faults, hazard node faults, inaccuracies of measurements, calibration errors, and fading signal strength etc. [15, 175]. Therefore, an additional goal is to reduce inaccuracies of inference results that occur due to data loss. Data loss may occur when the system converts point data (observations from individual sensor readings) into spatially distributed estimates of FWI. Therefore, a neighbourhood region prediction approach is adopted that assumes that the fire weather index at a specific place is most strongly influenced by nearby sensor nodes and least by distant sensor nodes. This approach utilizes a common technique – Inverse Distance Weighting (IDW) [188] for interpolation between a known scattered set of points.

Let \( f(sn_i) \) be the fire weather index at sensor node \( sn_i \), \( \{f(sn_1), f(sn_2), \ldots, f(sn_N)\} \) be the fire weather indices set, and \( N \) be the total number of fire weather indices. Hence, the following formula can be used to calculate the fire weather index \( f(L) \) at a specific location \( L \):

\[
f(L) = \sum_{i=1}^{N} w_i(L) f(sn_i) \sum_{j=1}^{N} w_j(L)
\]

where \( i(1 \leq i \leq N) \) and \( j(1 \leq j \leq N) \) are both index, and

\[
w_i(L) = \frac{1}{d(sn_i, L)^p}
\]

Where \( p \) is the power parameter (typically, \( p = 2 \)), and \( d(sn_i, L) \) is the distance between sensor node \( sn_i \) and location \( L \).

\[
d(sn_i, L) = r \cdot a \cos(\sin(x) \cdot \sin(x_i) + \cos(x) \cdot \cos(x_i) \cdot \cos(y - y_i))
\]
Where $x$ and $y$ are the latitude value and longitude value of the location $L$, $i_x$ and $i_y$ are the latitude value and longitude value of the sensor node $sn_i$, and $r$ (typically, $r = 6373.8$ kilometres) is the radius of the Earth. This IDW-Based Neighbourhood Region Prediction algorithm is applied to the inferred fire weather indices, prior to the display and animation of FWI spatial distributions within the Google Earth mapping interface (described in Section 6.5.2).

### 6.5 Implementation

#### 6.5.1 System Architecture

Figure 6.7 depicts the principle technical components of the proposed Semantic Fire Weather Index (SFWI) system and the data flows between them. As outlined in Section 1.5, the system architecture combines Web 2.0 technologies (Java, JavaScript, and JSON) and Web-based visualization technologies (Google Earth, Keyhole Markup Language (KML), Google timeline and pie charts) to maximize accessibility and interactivity. In addition, the Semantic Web technologies (RDF, OWL ontologies, SPARQL, and RDF Sesame Repository Stores) are applied to enrich sensor data with domain-specific semantic metadata, to reason across the sensor data streams, and to discover implicit knowledge such as fire weather index.

![Figure 6.7: High-level architecture view of the SFWI system](image)

A large volume of microclimate sensor data collected by Vaisala Weather Transmitter WXT520 devices is harvested from the Springbrook project: including 2.5 years of relative humidity sensor data, air temperature sensor data and wind speed sensor data. The data processing step performs data
cleaning (implemented by Java) to remove the outliers and conversion to RDF (implemented in Java). The generated RDF triples are stored in the relative humidity, air temperature and wind speed repositories respectively. The inference component performs the SPARQL reasoning to infer the fire weather indices at specific points (sensor nodes) and saves the results in the FWI repository.

A search interface was developed to enable users to search and retrieve fire weather index from the FWI Repository by specifying a region and time period. After a query region is specified, the IDW-Based Neighbourhood Region Prediction is applied to generate raster-based visualizations dynamically displayed in a Web-based Google Earth interface. In addition, a browser-based timeline enables users to display the fire weather trends over the query period (at 10 min intervals). In addition, three pie charts are generated that enable users to quickly compare average fire weather indices for the entire period, as well as FWI average values for daytime and night-time.

![Google Earth visualization interface](image1)

![Statistical analysis results](image2)

**Figure 6.8: User interface for displaying SFWI Search and Browse results**
6.5.2 User Interface

The user interface (shown in Figure 6.8) (accessible via a Firefox or Chrome Web browser), enables users to interactively:

- Search and retrieve the fire weather indices for a specific region and time range. For example, users can specify the region via the mapping interface and the time range via the search fields between ‘09/01/2012 12:00:00’ and ‘13/02/2012 12:00:00’. The search results are displayed overlaid on the specified region using Google Earth layers (See Figure 6.8 (a)).
- Statistically analyse the fire weather indices to compare daytime and night-time results. The results are displayed as pie charts (See Figure 6.8 (b)).

The Google Earth animations dynamically show how FWI values vary over time. The pie charts on the right show the statistical information about the average, daytime and night-time fire weather indices. The pie chart shows that typically FWIs are higher during daylight hours (15% low-, 62.5% low, and 22.5% moderate) than during the night time (64.7% low-, 15.7% low, 2% low+, 5.9% moderate+ and 11.8% high-). However, the pie chart also illustrates that although night-time is more likely to show lower fire weather indices, in some cases, higher fire weather indices may appear during the night.

6.6 Evaluation

6.6.1 Evaluation of the Accuracy and Precision of Fire Weather Index Calculations

The accuracy and precision of the proposed approach are evaluated, by comparing its results with the McArthur Forest Fire Danger Index (FFDI) and the Australian Bureau of Meteorology’s (BoM) Daily Fire Weather Index (which is based on the Canadian Fire Weather Index).

Firstly, six months of data (from 01/01/2012 00:00:00 to 01/07/2012 00:00:00) was collected from the Springbrook project. These data sets consist of three data streams: air temperature, relative humidity and wind speed. This evaluation involved calculating the similarity between the fire weather indices calculated by this system and the FFDI estimates for the same region. A comparison of results for each month is presented in Figure 6.9. They reveal that the proposed method and the FFDI produced very similar results: Jan.2012 similarity ≥98%, Feb. 2012 similarity ≥99%, Mar. 2012 similarity ≥96%, Apr. 2012 similarity ≥97%, May similarity ≥99.5%, and Jun. 2012 similarity ≥99%.

These results prove that the inference rules (as defined by the domain experts) can accurately estimate fire weather indices.
Secondly, this study compared the BoM’s Daily Fire Weather Indices (BoM-FWIs) with this system’s Semantic Fire Weather Indices (SFWI) (that change every ten minutes) for a five day period (from 09/01/2012 00:00:00 to 14/01/2012 00:00:00). The aim is to assess the precision of the FWI calculations. The results, shown in Figure 6.10, reveal that the implemented method is able to provide results that are more precise with higher temporal resolution. The results also show that FWI values
have strong relationships with time. More specifically, FWI values in the early afternoon are higher than in any other period.

Figure 6.10 demonstrates that the SFWI system is able to infer more accurate and precise fire weather index values with finer spatio-temporal resolution than the traditional approaches.

### 6.6.2 Evaluation of the RDF Triple Stores and SPARQL Query Performance

This experiment is conducted on a computer which has Intel (R) Core(TM) i5 2.93 GHz 8 processors, 16 GB RAM and the 64-bit Windows 7 operating system. The performance of the SPARQL inference and query engine was also evaluated by measuring the speed of SPARQL query and inference. 144,966 triples of weather parameters were generated. Next, a copy of the triples was stored in a single RDF repository (1R), the standard approach. A copy of the triples was also stored in the multiple repository storage (MR). To conduct this evaluation, eight SPARQL queries over different time periods (e.g., one hour, six hours etc.) were executed on the single RDF (1R) repository to infer fire weather indices. The corresponding execution time results are calculated and presented in Figure 6.11 labelled (NQ-1R) (New Query – 1 Repository). Secondly, these eight SPARQL queries were re-run again on the single RDF repository. The execution time results presented in Figure 6.11 are labelled as (RQ-1R) (Repeat Query – 1 Repository). Thirdly, these eight SPARQL queries were run on the multiple repository storage to infer the corresponding fire weather indices. The execution time results are presented in Figure 6.11 and labelled as (NQ-MR) (New Query – Multiple Repository). Lastly, these eight SPARQL queries were re-run again on the multiple repository storage. The execution time results are presented in Figure 6.11 labelled as (RQ-MR) (Repeat Query - Multiple Repository). The overall results illustrated in Figure 6.11 show that the multiple repository approach outperforms the single repository approach. Moreover, the proposed method performs significantly better than the single repository approach in terms of running repeated queries over a long period. Therefore, the proposed multiple repository approach demonstrates greatly improved querying speed.
Figure 6.11: Execution time for running SPARQL queries over different time periods for new queries and repeat queries over single and multiple repositories respectively (NQ-1R, RQ-1R, NQ-MR, RQ-MR)

6.6.3 Evaluation of System Usability

The system usability was assessed by soliciting feedback from eight users via a questionnaire and by observing user behaviour during a set of test tasks. The eight users (who were a mix of ecologists, engineers and information scientists) were asked to respond to the following questions on the questionnaire after they had completed a given set of tasks using the system:

- Q1-I found the user interface for searching fire weather indices easy to use;
- Q2-I found the Google Earth visualization and animations useful;
- Q3-I found the timeline visualization useful;
- Q4-I found the daily, day time and night time charts useful;
- Q5-I could understand the spatial and temporal patterns of fire weather indices better after using the system.

Users were asked to provide a response to each question from a five-point Likert scale: 1=Strongly agree; 2=Agree; 3=Neither agree nor disagree; 4=Disagree; 5=Strongly disagree.

Table 6.1: The evaluation results of usability – percentage of responses that were 1 or 2

<table>
<thead>
<tr>
<th>Questions</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Feedback (1-2 on Likert Scale)</td>
<td>100%</td>
<td>100%</td>
<td>87.5%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Table 6.1 shows the percentage of responses to the 5 questions that were either 1 (Strongly Agree) or 2 (Agree) on the Likert Scale. Table 6.1 shows that user feedback from the questionnaire was very positive. All of the users found that the user interface was intuitive and easy to use, and that the Google Earth visualizations and the pie charts visualizations were useful for exploring variations and trends in fire weather indices over time and location. Users requested that the timeline visualization should be improved to support zoom in and zoom out functionality. They felt this would be more useful for meteorologists or regional fire safety agencies to understand how the fire weather indices fluctuate over a long period or between regions. Moreover, a significant number of users requested the ability to overlay the Google Earth visualization fire weather index map with additional layers including: road/street maps, Digital Elevation Models (DEM) and land cover maps (forest cover or grassland). They felt that this would greatly assist firefighters and residents in developing firefighting, evacuation, and rescue plans.

6.7 Summary

In conclusion, many microclimate wireless sensor networks collect observations and measurements about environment physical parameters which provide ideal data for estimating bushfire risks. However, the quality of wireless sensor network data largely depends on the configuration of the sensor network. Existing fire weather index analyses that take noon time weather parameters collected from the distributed weather stations as input, cannot provide residents, firefighters or fire officials with accurate fire weather index information for a local region. Hence, the Semantic Fire Weather Index (SFWI) system has been developed. It overcomes the limitations of the existing fire weather indices by estimating fire weather indices from micro-scale wireless sensor network data streams. Specifically, semantic reasoning is integrated with domain expert knowledge to estimate fire weather indices for a specific region and time. Data pre-processing and an IDW-based algorithm were developed to support and improve the accuracy of the data underpinning the SFWI system. The FWI ontology was developed to enable the description of different level fire weather indices. In addition, a visualization interface was developed that enables fire managers, and the public to access the fire weather index data via an easy-to-use mapping and timeline interface. The outcome is an extensible framework and a robust foundation for future advanced wireless sensor networks that can be used to enhance the development of fire rescue systems or other environmental decision support systems.

The next and final chapter (Chapter 7) assesses the achievements of this thesis relative to the objectives described in Section 1.3. It lists the original contributions made to the field, discusses limitations with the implemented approaches, presents some directions for future research, and finally draws a conclusion about the outcomes of this thesis.
Chapter 7: Future Work and Conclusion

7.1 Overview

This chapter serves three main purposes:

- It summarizes the main contributions to the field from this thesis including the experimental results and the extent to which the outcomes met the objectives outlined in Chapter 1;
- It discusses the lessons learned from this thesis and identifies open issues requiring further research;
- It provides concluding remarks about the overall thesis.

7.2 Major Contributions

In Chapter 1, two case studies and four major goals associated with Semantic Annotation and Reasoning for sensor data streams, the focus of this thesis, were described. These high level goals were:

- To design, develop and evaluate a novel Semantic Annotation and Activity Recognition (SAAR) method to assist ecologists to automatically recognize animal activities from 3D accelerometry data streams, using a combination of expert tagging and supervised learning algorithms;
- To design, develop and evaluate an optimal graph based learning (OGL) approach to identify animal activities from 3D accelerometry data, given a smaller labelled training data set.
- To design, develop and evaluate an outlier and unusual event detection approach (SOUE-Detector) to support the identification, tagging and visualization of outliers within the environmental sensor data streams by combining statistical analysis with domain expert knowledge (captured via semantic annotations)); and
- To design, develop and evaluate a semantic rule-based reasoning method (SWFI) that enables fast and accurate reasoning of complex events (Fire Weather Indices) from multi-variate sensor data streams acquired from a wireless sensor network.

All four components were implemented within a common technical framework (outlined in Section 1.5).

In the following four sub-sections, the outcomes are compared to the objectives outlined in Section 1.3, to determine how successfully the thesis met its original aims.
7.2.1. Semantic Annotation and Activity Recognition

The Semantic Annotation and Activity Recognition (SAAR) system (described in Chapter 3) was designed to assist ecologists to identify animal behaviour from 3D accelerometry data streams across species. All of the objectives listed in Section 1.3.1 were met.

The result is a Web-based semantic annotation and activity recognition system, which supports storing, visualizing, and annotation of tri-axial accelerometer data streams, to enable collaborative animal activity recognition and analysis. It enables automated semantic annotation of the 3D accelerometry data streams by applying Support Vector Machine learning techniques to manually annotated corpuses of training data (tagged with behaviour terms e.g., *running*, *walking*, *standing*, *laterally recumbent*, *sternally recumbent*). More specifically, it provides a repository on the Web where researchers can upload and share their tri-axial accelerometer datasets, and also search, retrieve and compare datasets from the same or difference species. It provides an interactive graphical user interface that enables biologists to quickly and easily view and explore 3D accelerometry data streams and temporally align simultaneously recorded video (where available) that can be used as ground truth to verify behaviours. It also provides a set of Web services and visualization interface that enable the tagging of tri-axial accelerometry datasets and synchronized videos using terms from a controlled vocabulary (configured at run-time). These annotated data streams can then be used by scientists to train their own automatic activity classification models, which can be applied to new accelerometer data streams to automatically recognize animal activities. The evaluation results showed that the SAAR system enables ecologists with little knowledge of machine learning techniques to collaboratively build classification models with high levels of accuracy, sensitivity, precision and specificity – that can be shared among the community.

In implementing SAAR, a feature extraction approach was used to extract *standard deviations, signal magnitude areas, waveform lengths, inheritance parameters* and *frequency-domain features* from the tagged data streams. These features were then used to train SVM activity classifiers to automatically tag newly uploaded 3D accelerometry data streams.

The experimental results from the evaluation of SAAR (Section 3.4) reveal that:

- The larger the manually annotated training corpus, the better the performance of the SVM-based automatic activity recognition algorithm. In order to achieve high performance (Ave. Precision=97.4% and Ave. Recall =95.8%), a domain expert would need to spend approximately 27 minutes preparing a training set for each activity;

- The SAAR method outperforms best practice classification algorithms (ANN and HMM) in terms of both accuracy and precision;
High-level classification models (that identify active and inactive) produce better results (precision > 97%) than the low-level classifiers (that identify running, walking, standing, lying and sitting) (precision > 78%);

Results on data acquired from domestic animals are better than the results using data from undomesticated animals because the animals do as their owners/trainers tell them and there is less noise;

The classification modules performed best (> 90% precision) when applied to accelerometry data collected from the same species on which it was trained;

Behaviour classification modules trained using accelerometry data collected from one individual can be used to identify and quantify behaviour modes in different individuals and even different species. Hence, a tame surrogate animal (e.g., domestic dog) can be used to build a behaviour classification module capable of accurately identifying and quantifying behaviour modes using accelerometry data streams collected from other free-ranging and similar-sized species (e.g., a dingo).

Behaviour classification performance across species degrades if the Spine Length: Spine Height (SL:SH) ratio of the subject animal is more than double the SL:SH of the surrogate used to train the classifier.

### 7.2.2. Optimal Graph-based Learning (OGL) for Automatic Annotation

In Chapter 4, a novel OGL approach for recognizing animal behaviour in 3D accelerometry streams is designed, developed and evaluated.

To be specific, OGL is combined with the semi-supervised machine learning (SSL) to automatically identify animal behaviours from 3D accelerometry data. The aim is to achieve high classification performance with a smaller labelled training data set (than SAAR requires). The proposed optimal graph learning model is further extended to address out-of-sample and noisy label issues. To further enhance performance, the graphs are constructed from multiple cues. The proposed OGL+SSL method is also evaluated by comparing it’s performance with other graph construction methods, when applied to image classification tasks. All of the objectives listed in Section 1.3.2 are met.

The experimental results (Section 4.4) reveal the following:

- The proposed OGL+SSL approach can more effectively and accurately tag animal behaviours with a smaller labelled training data set, than SAAR.
• The experiments applying OGL+SSL to image annotation on real world datasets (i.e., Corel5k, ESP Game and IAPR TC12) demonstrate the superiority of OGL over existing graph construction methods, including LGC, LLE, L2Graph, CRM, NPDE, SML, MBRM, JEC and TagProp.

7.2.3. Semantic Outlier and Unusual Events Detector

Chapter 5 describes the implementation and evaluation of the SOUE (Semantic Outlier and Unusual Event) Detector system – a novel approach to detecting outliers and unusual events in sensor data streams by combining statistical analysis (using Dynamic Time Warping) with domain expert knowledge (captured via an ontology and semantic inferencing rules).

The resulting Web portal enables domain experts to: document their expert knowledge about correlations between specific sensor properties (and capture them in the Correlation of Environmental Sensor Properties (CESP) ontology); execute the DTW algorithm to detect errors and unusual events; and efficiently search across a collection of wireless sensor data streams to retrieve and display segment outliers (both errors and unusual events). A mapping and timeline interface enables scientists to search and browse across the detected outlying segments to assist with verification and differentiation of true errors and unusual events.

To handle the dynamic nature of wireless sensor networks, a sensor tracking feature was also implemented. Whenever the system receives a sensor network modification message (meaning that sensor nodes and/or sensors have been changed), the system attaches end date/time stamps to the previously generated sensor data node and sensor matrices, and then recalculates the new sensor node and sensor matrices and saves them in the database with the new date/time stamps. The sensor network configuration tracking data is then taken into account when determining outliers by comparing data streams from neighbouring sensor nodes and sensors.

All of the objectives outlined in Section 1.3.3 have been met.

The experimental and evaluation results (Sections 5.6.2 and 5.6.3) showed that:

• The SOUE-Detector can efficiently detect segment outliers and unusual events with high levels of precision and recall, by exploiting sensor data trend similarities and correlations between sensor properties;

• The performance is very sensitive to the similarity threshold values. In order to obtain optimum performance, similarity threshold values should lie in the range 84-90%.
The SOUE-Detector outperforms two alternative outlier detection techniques that employ an artificial neural network (ANN) and SVM respectively.

7.2.4. Semantic Fire Weather Index (SWFI) System

The SFWI (Semantic Fire Weather Index) system (described in Chapter 6) adopts a rules-based approach for calculating Fire Weather Indices (a.k.a. fire danger ratings) from sensor data streams generated from a wireless sensor network deployed in Springbrook national park (Case Study #2).

The SFWI implementation enables domain experts (bush fire experts) to define SPARQL inference rules for estimating fire weather indices from sensor properties such as air temperature, humidity, rainfall and wind-speed. A Fire Weather Index ontology was developed to standardize the representation of different levels of fire weather danger ratings. A novel multi-repository method was also developed to enable faster storage, querying and retrieval of large volumes of sensor observations in multiple RDF triple stores. In addition a novel inferencing algorithm that combines SPARQL inferencing with Inverse Distance Weighting was developed to improve the accuracy of the fire weather index inference results. The interactive Web-based search interface enables users to access and query the computed fire weather indices in real time and at 10 minute intervals, via both maps and timelines. The outcome is an extensible framework and a robust foundation for processing multiple sensor data streams from wireless sensor networks to infer higher-level information that can inform environmental decision support systems.

All of the objectives in Section 1.3.4 were met.

The evaluation of the SFWI system (Section 6.6) indicated the following:

- The implemented approach outperforms both the McArthur Forest Fire Danger Index (FFDI) and BoM’s Daily Fire Weather Indices (BoM-FWIs), in terms of accuracy, precision and spatio-temporal resolution.
- The proposed multiple repository storage approach (which stores different environmental parameters (air temperature, wind speed and relative humidity) in different RDF triple stores) performs significantly better than the single repository approach in terms of running repeated queries over a long period.

7.3 Lessons Learnt and Future Research

In addition to the contributions described above, a number of lessons have been learnt and future research activities have been identified that are likely to lead to further improvements in the state-of-the-art of semantic annotation and reasoning over sensor data streams.
Semantic Annotation for Animal Activity Recognition

The SAAR system supports automatic semantic tagging of 3D accelerometry data (focussing on the recognition of for animal behaviour) by integrating semantic annotation and visualization services with Support Vector Machine (SVM) techniques. Further research is required in order:

- To investigate more scalable approaches for analysing data files with higher sampling frequencies (> 1Hz) or that cover extended periods (>2 hours). The SAAR system currently only supports accelerometry data of sampling rate 1Hz. With some species, this sampling rate would be insufficient to recognize specific activities. Also, the zoom in and zoom out functionalities associated with the Plot panel are very slow when displaying higher sampling rates (> 1 Hz) or large data sets e.g., 3-5 hour data sets.

- To integrate the Plot and Video visualization interfaces with Google Map to enable simultaneous visualization of tri-axial accelerometer data streams, videos and GPS location information. A growing number of ecologists are attaching GPS acoustic and satellite tags [3, 189] that track GPS location on a larger scale as well as other sensors (that measure body temperature, heart rate, bioacoustics etc.) to animals, in addition to accelerometers. SAAR could usefully be extended to support the integration, visualization and analysis of these additional parameters – in order to detect more complex behaviours, including interactions between animals, such as mating, fighting or territorial marking;

- To focus on generating an energy expenditure distribution map (by analysing both animals’ day and night movements and to predict animal health statuses), and authentication and access control protocols over the datasets and associated tags. The uploaded data sets, to date, are openly available via the SAAR Web site; however, many researchers would prefer to limit access to their experimental data only to project partners, at least until the data has been published; and

- To evaluate the classification results using different types of SVMs (e.g., nu-SVC, regressing SVM) and different kernel functions – to determine which SVM and kernel function produces the best results. This study, to date, has only evaluated SVMs using the C-SVC algorithm.

Optimal Graph Learning-based Annotation of Sensor Data Streams

The OGL+SSL approach developed in Chapter 4, demonstrated the feasibility of this approach for classifying animal accelerometry data sets and image collections. Future research aims include:
• Optimizing the speed and performance of the OGL+SSL approach by determining the optimum trade-off between the size of the training corpus (labelled data set) and the speed of graph construction;

• Investigating the design of a more efficient, faster graph construction algorithm that is scalable for larger training corpuses;

• A basic linear regression model was adopted to learn the annotation function for new data points. It is anticipated that the performance could be improved by utilizing more complex models in this step;

• Applying and evaluating OGL+SSL to other challenging classification tasks, such as image ranking, object detection and video annotation.

Data Quality Enhancement of Sensor Data Streams

The SOUE-Detector has only been evaluated on test datasets collected from the Springbrook wireless sensor network that have had erroneous outliers and unusual events artificially inserted for testing purposes. Future work aims include:

• Evaluating the proposed algorithms and system on real-world (not artificially manipulated) data streams from the Springbrook and other wireless sensor networks that have not been cleaned and contain both erroneous data streams and unusual events;

• Evaluating the scalability and speed of performance of the proposed algorithms when applied to data sets that are much bigger than the test data i.e., larger volumes of sensor data streams that cover longer periods, have higher sampling rates, or include more observed parameters;

• Linking the SOUE-Detector with the Springbrook network “gateway” system to enable real time detection of segment outliers and unusual event detection and subsequent generation of notification services for decision-making.

• Taking into account the 3D nature of sensor locations when calculating the distance between sensor neighbours. The Springbrook data set did not include height location information for sensor nodes but ideally height differences between sensors should be considered when determining the spatial neighbourhood matrices (Section 5.4.4).

Rule-based Semantic Reasoning

The SFWI system overcomes the limitations of the existing fire weather index by estimating fire weather index from micro-scale wireless sensor network data streams. Future research includes developing a user interface that enables meteorologists to easily enter, save and publish their domain
expert knowledge via human-readable rules that can be translated to corresponding SPARQL rules. Future research is also required to validate or quantify rules and correlations between observed variables using statistical analysis tools (e.g., R regression tools) and to integrate SPARQL inference with fuzzy inferencing technologies - to provide users with more accurate fire weather indices information and estimates of uncertainty. Finally, other environmental data sets such land cover (forest land cover and grassland land cover), digital elevation models, creek and road networks could usefully be combined with the sensor network data streams to assist with the development of Bushfire Protection Plans.

More generally, since the implementation of the research described in this thesis, there have been additional technical advancements that should be considered as fruitful areas for future research in this field. In particular, one aspect that this thesis did not focus on is enabling real-time annotation and reasoning over sensor data streams. Approaches that could be explored to support real-time semantic processing include integrating the open source framework, Massive Online Analysis (MOA) [190], which includes a collection of machine learning algorithms designed for real-time processing of big data streams. A faster RDF triple store with built-in parallel optimisation (e.g., Virtuoso [191]) could also be used instead of RDF Sesame. The emergence of the RDF streaming field [46, 47, 192] also has significant potential for improving the speed of linking relevant events (extracted from real-time data streams) on-the-fly, enabling faster extraction of near-real time information and knowledge for decision support in critical situations.

7.4 Concluding Summary

This thesis began by arguing that ontology-based semantic annotations of sensor data streams are critical to: the discovery of significant data and events; for reasoning across annotated sensor data streams to deduce new or implicit knowledge; and to answer complex queries that rely on the integration of multiple, real-time, multivariate data streams. Chapter 1 argued that there was an urgent need for more effective semantic annotation and reasoning services for sensor data streams that are faster, more accurate and can handle the increasing volumes of data streams that need analysis.

This thesis delivers a set of innovative approaches to the Semantic Annotation and Reasoning of Sensor Data – that advance the state of the art, by applying and evaluating a number of novel techniques in the context of two case studies and their associated sensor datatypes: animal accelerometry datasets and environmental data streams from a wireless sensor network.

From a conceptual perspective, a set of ontologies has been presented to support sensor data annotation, management and analysis – both generally (through high level ontologies) and within specific applications (through domain specific ontologies and ontological extensions and
refinements). Existing ontologies that have been extended and refined include the Open Annotation Collaboration (OAC) Data Model and the SSN (Semantic Sensor Network) ontology. New ontologies that have been developed include: the Animal Behaviour ontology; the CESP ontology to describe sensor properties and correlations between sensor properties, and a FWI ontology to define weather indices and relative properties.

From an implementation perspective, specific algorithms and tools for realising semantic annotation and reasoning within two domains: the animal behaviour domain and the environmental domain, have been designed and built. Four novel approaches have been proposed, implemented and evaluated within a common technical framework: the SAAR system for storing, visualising, annotating and automatic recognition of animal activities from tri-axial accelerometer data streams; the OGL+SSL approach for improving the classification of animal accelerometry data streams using optimal graph learning combined with semi-supervised machine learning; the SOUEDector system to identify, tag, retrieve and display segment outliers (both erroneous and genuine) in sensor data streams; the SFWI system to specify and implement rules for inferring and visualizing higher level environmental indicators (Fire Weather Indices) that assess the risk of critical situations (i.e., bush fires).

The original hypothesis of this thesis was that the efficiency, accuracy and quality of semantic annotation of sensor data streams can be improved by combining domain expert knowledge (captured as ontology-based, semantic annotations) with supervised, semi-supervised and optimal graph machine learning technologies. Moreover by applying rule-based reasoning over the annotated sensor data streams, erroneous data in the sensor data streams can be detected and filtered and higher level knowledge, such as complex critical events can be more accurately detected. The original contributions described in Chapters 3, 4, 5 and 6, validate the original hypothesis and advance the state of the art in semantic annotation and reasoning services for sensor data streams.
References


N. B. Cook, T. B. Bennett, and K. V. Nordlund, "Monitoring Indices of Cow Co


[92] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-Media Hashing for Large-Scale Retrieval from Heterogeneous Data Sources," in SIGMOD, New York, USA, 2013, pp. 785-796.


