Adaptive Activity Recognition Techniques with Evolving Data Streams

by

Zahraa S. Abdallah, BCompSc

Thesis

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Faculty of Information Technology
Monash University

December, 2014

To those who sacrificed themselves to give us freedom; to my brother who left us in 2013.

Adaptive Activity Recognition Techniques with Evolving Data Streams

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Zahraa S. Abdallah
December 16, 2014

Acknowledgments

The research included in this dissertation could not have been performed if not for the assistance, patience, and support of many individuals. I would like to extend my gratitude first to my supervisors: A/Prof. Shonali Krishnaswamy, Dr. Mohamed Medhat Gaber, and Prof. Bala Srinivasan for guiding me throughout the PhD journey.

My sincere thanks go for Shonali for her support, encouragement, and enthusiasm. Her invaluable ideas, attention to detail, and feedback helped me to shape this thesis.

I gratefully thank Mohamed for his insights, guidance and belief in my ability even when mine wavered. Mohamed is the first one encouraged me to start my PhD. I convey my gratitude to him for his help throughout the journey and his prompt response to my numerous queries.

I convey my deepest gratitude to Srini who has helped me through extremely difficult times over the writing of the dissertation and for that I sincerely thank him for his confidence in me. I am much indebted to him for his critical and thoughtful comments and feedback that have shaped up the resultant thesis.

I would additionally like to thank Dr. Arvind Ratnaike for his help in proofreading the thesis. His knowledge and understanding of the written word has allowed me to fully express the concepts behind this research.

I would like to thank Allison Mitchell, Michelle Ketchen, Diana Sussman, Samedin Balla, Denyse Cove, and Akamon Kunkongkapun for always being helpful and providing me with administrative and technical support during my research.

I would also like to extend my appreciation to my colleagues and fellow PhD students, Thi Pham, Dwi Rahayu, Varsha Mamidi, Misita Anwar, Joanne Mihelcic, Pari Delir Haghighi, Peter Serwylo, Adel Moslehi, Mark Creado, and many other school friends who made the candidature period more enjoyable and helped in numerous ways.

I would also like to thank Frada Burstein, Maria Indrawan, Mark Carman, and Mahbubur Rahim for their assistance in numerous capacities.

I would like to extend my deepest gratitude to my parents and my family. Without their unconditional love, support and understanding I could never have completed this doctoral degree. It is to them that I owe everything I am.

This research would not have been possible without the continuous support and courage of my soulmate, my husband Ahmed. No words can truly express how grateful I am for having him next to me in all ups and downs. My love and thanks for my kids, Salman and Malik, for giving me the joy of seeing the bright future in their eyes.

Finally, I owe a great thanks to my best friends who make happiness worth sharing and make dreams worth believing. Thanks for being by my side at all times.

Abstract

Activity recognition aims to provide accurate and opportune information on people's activities by leveraging sensory data available in today's sensory rich environments. Activity recognition has become an emerging field in the areas of pervasive and ubiquitous computing. The process of recognising activities flows through three key steps: sensing, modelling, and recognition. A typical activity recognition technique processes data streams that evolve from sensing platforms such as mobile sensors, on body sensors, and/or ambient sensors. Learning models in activity recognition are built from historical data and rely strongly on prior knowledge of activities. The learning model in this scenario is static and thus unable to cope with the evolving nature of activities in data streams.

The evolving nature of activities arises for many reasons. Intuitively, people perform activities in different ways. "Walking" for one person could be "jogging" for another. Therefore, there is no model that fits all in activity recognition. To attain an accurate recognition, a learning model has to be tuned to suit a user's personalised way of performing activities. Moreover, it is unrealistic to assume that the number of activities is static along the stream. While the learning model is built from historical data, novel activities may emerge and abandoned ones may disappear over time.

This thesis develops adaptive techniques for activity recognition that dynamically change the learning model while activities evolve. These techniques apply an incremental and continuous learning approach for both personalisation and adaptation of the learning model. As a strategy to harness the potential of activity for pervasive environments, our techniques are capable of recognising activities that evolve from data streams. The first contribution of this thesis is to build a flexible, efficient, robust, and accurate learning model that enables personalisation and adaptation with evolving data streams. This learning model is the core for all our techniques developed in this thesis.

Based on the developed learning model, we propose a technique for recognising activities efficiently. The recognition technique is an ensemble classifier that integrates with the learning model to recognise activities based on a hybrid similarity measure approach. The merit of this approach is to bring different perspectives together for more accurate recognition, especially across users. The ensemble classifier is evaluated on benchmarked datasets for activity recognition. The evaluation demonstrated the robustness, efficiency, and

accurate recognition of activities. Our technique shows its best performance when applied across users and with noisy data. The accuracy is improved by more than 10% in these cases compared to other state-of-the-art techniques in activity recognition using benchmarked multi-dimensional datasets.

The above activity recognition technique is extended to include incremental learning for personalisation with evolving data streams. This technique leverages the flexibility of the learning model for personalisation in real time to achieve an accurate recognition with the evolving activities. Furthermore, we deploy our technique on a mobile device to demonstrate its efficiency. Although the streaming environment imposes more constraints on the recognition process, the proposed recognition technique outperforms other benchmarked incremental techniques in activity recognition. Our technique shows its best performance when applied to data that contains noise with accuracy enhancement of about 15%.

The last contribution is a technique that enables continuous learning to adapt the learning model. To fulfil this goal, our technique detects the arrival of new activities in data streams and/or the disappearance of abandoned ones. Moreover, it dynamically adapts the learning model with the detected changes for a future recognition. The developed technique is evaluated on benchmarked datasets to demonstrate its efficiency in recognising changes in activities and adaptation of the learning model accordingly. The recognition of novel activities varies depending on the characteristics of the datasets and the nature of the detected activity. This technique, as well as all techniques in this thesis, incorporates active learning to address the scarcity of labelled data especially in streaming environment by annotating only small amounts of the most informative data. Thus, this thesis takes a step forward in activity recognition dynamics in pervasive and ubiquitous computing by building efficient and adaptive techniques for recognising evolving activities.

Thesis Publications

- **Journal Papers:**

  1. Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. *Adaptive mobile activity recognition system with evolving data streams.* vol 50, part A, pages 304–317, Neurocomputing, 2015.

- **Conference Papers:**

  1. Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. *StreamAR: incremental and active learning with evolving sensory data for activity recognition.* In IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI), volume 1, pages 1163–1170, IEEE, 2012.
  2. Zahraa Said Abdallah and Mohamed Medhat Gaber. KB-CB-N classification: Towards unsupervised approach for supervised learning. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence, pages 283–290, 2011.
  3. Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. *CBARS: Cluster based classification for activity recognition systems.* In Advanced Machine Learning Technologies and Applications, volume 322 of Communications in Computer and Information Science, pages 82–91. Springer Berlin Heidelberg, 2012.
  4. Zahraa Said Abdallah and Mohamed Medhat Gaber. *DDG clustering: A novel technique for highly accurate results.* In Proceedings of the IADIS European Conference on Data Mining, pages 163–167, 2009.
  5. Zahraa Said Abdallah, Shonali Krishnaswamy, Mohamed Medhat Gaber, and Bala Srinivasan. *Mobile Activity Recognition Using Contextual Reasoning and Ubiquitous Data Stream Processing.* In Doctoral Consortium, Computer Science Week 2012 (ACSW2012), RMIT, Melbourne, Australia

- **Other publications:**

  1. Prem Prakash Jayaraman, Joao Bartolo Gomes, Hai Long Nguyen, Zahraa Said Abdallah, Shonali Krishnaswamy, and Arkady Zaslavsky. *Cardap: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog.* In Advances in Databases and Information Systems, pages 192–206. Springer, 2014.
  2. Andrey Boytsov, Arkady Zaslavsky, and Zahraa Abdallah. *Where have you been? using location clustering and context awareness to understand places of interest.* In Internet of Things, Smart Spaces, and Next Generation Networking, volume 7469 of Lecture Notes in Computer Science, pages 51–62. Springer Berlin Heidelberg, 2012.

- **Competitions:**

  – Ranked first in OPPORTUNITY Activity Recognition Challenge for Task A.
    Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Troster, Jose Del R. Millan, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, 34(15):2033-2042, November 2013.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Preamble

Sensors are becoming more pervasive. They exist ubiquitously around us and are embedded in our phones, cameras, clothing, buildings, cars, and in all kinds of everyday objects. Massive amounts of data are generated from these sensors continuously. That raises the question of how can we use this sensory data to make our lives "better"? The availability of real time sensory information through these sensors has led to the emergence of research into "Activity Recognition" (AR). Activity recognition aims to provide accurate and opportune information based on people's activities and behaviours. Activity recognition has become an emerging field in the areas of pervasive sensory data processing and ubiquitous computing. Many applications have demonstrated the usefulness of activity recognition. These include applications in health and wellness, activity-based crowdsourcing and surveillance, and targeted advertising.

- *Health and wellness* [LRE10, KNM+06a, HKAK10, STF08, DLL13]: Progressive research in activity recognition has provided the foundation for many applications in heath and wellness. In recent years, fitness tracking applications have attracted much attention in activity recognition [Yan09, BI04, EPMK08, CMT+08]. Devices and applications, such as fitbit [fit], monitor distances walked and corresponding burned calories. More advanced fitness applications aim at tracking Activities of Daily Living (ADL) [HKAK10]. Recognition and monitoring of the type and frequency of ADL is essential for creating what is known as an activity diary/log [Yan09]. These diaries help users to understand their personal lifestyle patterns and effect healthy changes (e.g. increase physical exercise, reduce number of hours sitting in front of the computer). Such

activity recognition applications are important for preventing medicine and avoiding chronic illness such as cardiovascular diseases, diabetes and obesity. Other applications in health monitoring are concerned with the remote supervision of home based patients or at risk elderly people. Medical professionals believe that one of the best ways of early detection and prevention of emerging medical conditions is to recognise changes and abnormality in different activities [LB69].

- *Activity-based crowdsourcing and surveillance* [SG00, BJPT03, WA05]: Recognising activities for crowds leads to interesting applications. A case of a large number of people running in a place where they normally walk or sit indicates a possible emergency or disaster [LPW12]. Also, surveillance applications that are able to understand and model people's activities could predict intent and motive as people interact with the environment. Therefore, activity recognition applications aim to proactively detect abnormal behaviours in busy environments [NLHW04]. An example is a suspicious person who is spending longer than usual time on a train platform.

- *Targeted advertising* [Int04, SVA06]: Activity recognition in real time is an important component of applications that interact with users to deliver context relevant information and services. Applications in this category target conveying the right message at the right place and at the right time using activity recognition. Examples include personalised advertisements or discount deals in smart shopping scenarios. The term "Know Your Customer (KYC)" [WG96], which has been used by businesses, refers to understanding customer needs and providing them with satisfying services. Activity recognition contributes to KYC analytics by inferring the general interests of customers and thus helps in providing them with relevant information and services.

Activity recognition has been widely studied using different approaches and from various perspectives. Probabilistic, statistical and logical reasoning approaches have been applied to understand and predict various user activities. Additionally, machine learning approaches based on sensory data have also been leveraged for activity recognition. The premise underlying the use of machine learning in activity recognition is that activities can be recognised and even anticipated using prior knowledge of previously collected data representing different activities. Typical activity recognition process consists of three main components: namely, *data collection and preprocessing*, *modelling*,

and finally *recognition*. The *data collection and preprocessing component* that gathers annotated sensory data evolves from diverse data sources such as on body wearable sensors, mobile sensors, and/or smart environment sensors. Then, raw sensory data is processed into features that help discriminate between activities. The *modelling component* uses the extracted features to train a baseline learning model that is then deployed to predict activities from new incoming sensory data by the *recognition component*.

Thus, state of the art activity recognition techniques rely strongly on prior knowledge and typically recognise activities based on models built from samples of the population. However, given the nature of human activities, it is essential to recognise activities that change from one user to another. Data that represents a particular activity might change from one user to another, i.e. walking for one person could be jogging for another. Also, a single user might perform one activity with different patterns, i.e. running might include fast running or jogging. New activities could emerge and abandoned activities may disappear. Therefore, it is unrealistic in activity recognition to assume that data is static over time. Dynamic changes in activities that reflect variations in user's activities are expected and natural. Change of existing activities or emergence of novel activities occur in evolving activity data. Traditional activity recognition models are built on previously collected data, ignoring the crucial post deployment refinement and adaptation to cope with aforementioned changes that naturally occur in an activity life cycle. Thus, it is important to develop techniques that effectively recognise activities that change and evolve.

In this thesis, our objective is to develop techniques that are able to extend state of the art activity recognition by enabling the AR process to: (i) detect the emergence of new and previously unseen activities, and (ii) personalise activity models built from a group of users to a set of different users. This approach can dramatically improve the interpreting of activities from evolving sensory data, hence positively impacting the accuracy, effectiveness, and robustness of diverse applications that leverage activity recognition.

The rest of this chapter is organized as follows. Section 1.2 provides an overview of activity recognition approaches and is followed in Section 1.3 by a discussion of the challenges that researchers face in the field of activity recognition. The aim and objectives of the research are described in Section 1.4. Then, the research scope and contributions are outlined in Section 1.5. Section 1.6 concludes the chapter with an outline of the dissertation organization.

1.2 Approaches for Human Activity Recognition

State of the art activity recognition research has focused on traditional supervised learning techniques [PGK+09, MS10]. The flow of the learning process through different components in activity recognition is illustrated in Figure 1.1. In this section, we explain the research areas and approaches in activity recognition through their key components.

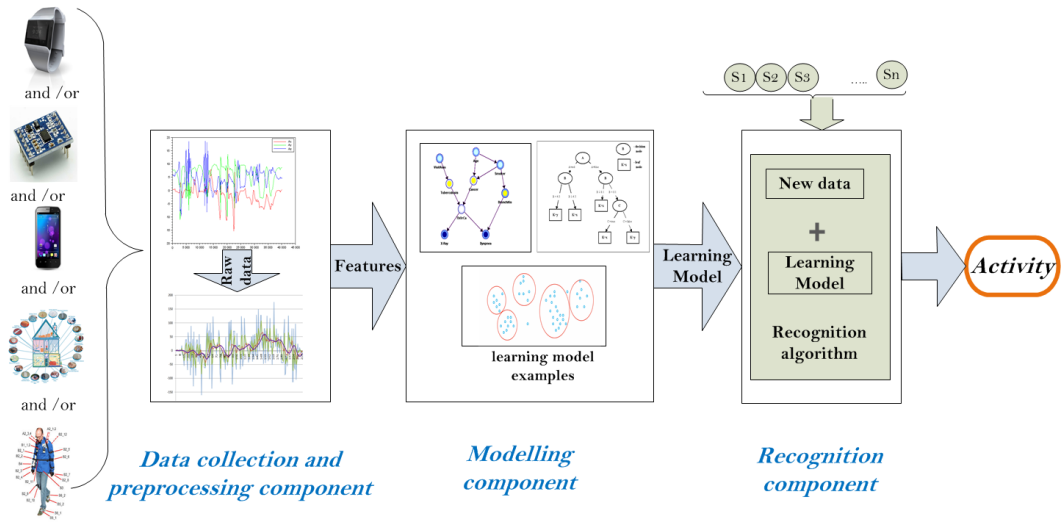


Figure 1.1: Activity Recognition Components

- *Data collection and preprocessing component*: This component represents the very initial step of any activity recognition process. Data collection includes issues related to the sensing platform such as types and locations of the sensors. Both types and locations vary based on the aim of the recognition. For instance, recognising hand gestures may require accelerometer sensors attached to hands or fingers. Variously, spatio-temporal activities require collection of GPS data that could be from a user's device that they carry. Atomic activities such as sitting and walking may also be recognised using accelerometer data embedded in the mobile device or attached as wearable sensors. Another consideration for the data collection concerns the annotation or labelling of activities in the collected data. The process of data collection is followed by a preprocessing and feature extraction steps which aim to prepare the raw collected data for the following component of modelling. The key factors that need consideration for data collection and preprocessing are as follows:

- *Type of sensors:* Various kinds of sensors are deployed for activity recognition. Three-axis accelerometer sensors are probably the most frequently used wearable and mobile sensor. They are particularly effective in monitoring actions that involve repetitive body motions, such as walking, running, sitting, standing, and climbing stairs. Location sensors such as GPS are also used in activity recognition for understanding higher level activities. Other sensors, such as inertial, gyroscopic, temperature, and microphone sensors, are deployed also for data collection in activity recognition. Some studies use only a single sensor, while others use multiple sensors of the same kind in different locations. Many other studies deploy a network of a combination of various sensors for a holistic understanding of the performed activities.
- *Location of sensors:* Sensors could be attached to a person to collect orientation and movement information. Such wearable sensors have the advantage of being with the user continuously. Several studies have also considered the use of widely-available mobile phone sensors for activity recognition. In addition to sensing, phones come with computing and communication resources. Importantly, the popularity of the mobile phones enables researchers to deploy and test the developed models for activity recognition on a large number of users. In addition to wearable sensors, ambient sensors that are deployed in physical environments such as smart homes have been widely used for activity recognition, especially in health applications and elderly people monitoring.
- *Data annotation:* Data is collected from either single or multiple users. In order to build and train the learning model, raw data is labelled with the corresponding activity that occurred while that data was being collected. Collecting data over a long period of time and from a large number of users produces generalised and robust learning models. Typically there is only a small set of labelled training data available in addition to a substantial amount of unlabelled training data. The scarcity of annotated data directly impacts the quality of the learning model and therefore the performance of the entire recognition system. However, given that annotating activity data is an expensive process in terms of human effort, researchers have started studying the feasibility of other machine learning techniques for activity recognition including unsupervised, active, and semi-supervised learning [GWT+09, LRE10, SVLS08, LC14] to overcome challenges arising from the limited availability of labelled data.

- *Data preprocessing and feature extraction:* Raw data is processed with feature selection and extraction methods to extract meaningful information that can distinguish between different activities. For example, accelerometer raw data (i.e. x, y and z component) needs to be transformed to a set of features that include the magnitude, mean, standard deviation, and number of peaks of the accelerometer readings along the three axes. Studies in activity recognition have applied various well-known feature extraction and selection techniques in the collection and preprocessing component that also includes data filtration from both noise and outliers. [HKAK10, KWM11, KLLK10]

- *Modelling component:* The modelling component is a key component for building an accurate and robust activity recognition system. A model is initially built and trained with features extracted from collected annotated and preprocessed data. The generated learning model is typically a classificatory model similar to a decision tree, Bayesian network, support vector machine, or neural network. The recognition system deploys the learning model for later recognition of unannotated data.

  Many criteria are considered for determining the efficiency of the learning model in general [WF05]. One of the most widely used criteria in activity recognition is *accuracy*, which is the ability to correctly recognise the occurring activity. Accuracy is not the only criterion. *Effectiveness* is also studied for evaluating the learning model from computational and time complexity perspectives. Efficiency is an important criterion to consider, especially when dealing with large amounts of data. Moreover, the issue of noisy and missing data is very common when dealing with sensory data. *Robustness* is another essential evaluation criterion for testing the ability of the learning model to handle missing and noisy data. As activity recognition data evolves over time, the *flexibility* criterion or the ability to adapt the model continuously to reflect changes in the data which are reflections of changing and evolving activities themselves is a significant way to evaluate the learning model.

  In addition to how a model is evaluated, there are several key issues that need consideration as discussed below:

  - *Personalised versus generalised model:* Typical AR models are trained for a subset of users and deployed to recognise activities for a large population group. Thus, either over-fitted or poorly trained model could be reasons for poor/inaccurate recognition. Therefore, it is important

to consider personalisation of the learning model to fit a specific user's way of performing activities. Personalisation is an important factor for *accurate* and *effective* recognition. Typically, walking for one user could be running for another, therefore flexibility to tune the initial model to accommodate specific user activities is crucial for capable activity recognition.

- *Model structure:* Building a fine-grained model that describes a set of different activities precisely is essential for an *accurate*, *robust*, and *efficient* recognition. Furthermore, a model with a small computational footprint is important, especially for real time recognition on devices such as mobile phones. One key challenge in choosing the model structure is keeping the balance between an accurate model representation with fine granularity and computational efficiency.
- *Continuous learning and model adaptation:* In realistic conditions, change of activities may emerge over time. A user's way of performing activities might naturally change over time. Furthermore, the set of activities a user performs changes over time, wherein new activities may emerge and also current activities may disappear gradually. Model *flexibility* to support continuous learning is essential to handle these changes and therefore maintain the efficiency of the recognition system. Model flexibility includes adding new activities, deleting existing activities or modifying user patterns. Flexible models limit the need for re-training the learning model when changes occur to activity patterns, due to either users or environmental settings. The model degradation over time is a notable challenge for machine learning systems in general [Kun04]. However, this is a key issue that needs to be addressed in the context of AR given the natural evolution of activities. Flexible and accurate activity recognition systems should have the elasticity to learn from the new incoming sensory data and adapt the model accordingly.

• *Recognition component:* The final step in any activity recognition system is the recognition component where the learnt model is deployed for recognising activities from unannotated data. The following are some of the key considerations that have to be factored into the recognition component:

- *Model deployment:* The recognition process starts with the deployment of the learning model on target device. The deployment location

is commonly the platform where processing and recognising activities happens. A learning model could be deployed directly on the sensors, on a mobile device, or on the cloud server that receives data from sensors. The trade-offs of deploying the model at the source that aggregates the data (i.e. sensor) versus the server (i.e. cloud) are as follows. On the cloud, there is sufficient computational power. Thus computational efficiency many not be a key consideration. However, this brings overheads in terms of higher communication costs and reduced privacy. On the other hand, recognition on a mobile device ensures that user's data remains on the user's device thus preserving privacy. However, it is limited by the mobile device resources and capabilities.

- *Types of recognised activities:* Activities of daily living (ADL) are the typical atomic activities which users perform in their everyday life. Activity recognition systems are used to recognise ADLs such as walking, standing, sitting and jogging [TIL04]. Depending on the learning model and collected data, various activities could be recognised by the recognition component. More detailed activities including gestures such as cooking, washing, and drinking coffee are also included by many studies especially ones that use smart environmental and/or wearable sensors [WLT05].
- *Instance versus batch based processing mode:* Processing data for recognition either occurs by responding to each instance, or processing data in a batch mode within a time window. For activity recognition, people perform activities in a sequential manner (i.e., performing one activity after another). Thus, activity data is typically a sequence of chunks that represent a sequence of activities. Batch based processing in activity recognition deals with activities in batches rather than processing each instance. Although, instance based processing attains a real or near real time recognition, batch based processing preserves the connection between a sequence of instances that represents the activity and therefore boosts the recognition accuracy. Batch based processing could attain a near real time recognition with a small window size. The choice of instance and batch modes impacts the computational efficiency of the recognition technique.
- *Real-time recognition:* One of the main performance metrics of any efficient activity recognition technique is the real time recognition of a

user's performed activities. Real time recognition enables many proactive and preventative applications across different domains. For instance, applications in surveillance and health require immediate action to be initiated based on the recognition results for emergency situations. For example, detecting a sudden fall or an abnormal activity in real-time is crucial especially for elderly people. Real-time recognition enables also location-based applications that are even increasing.

1.3 Challenges in Activity Recognition

Although several important aspects of activity recognition have been studied, there are still many issues that motivate the development of new techniques to improve an activity recognition system's performance under more realistic conditions [LL13]. Thus, many crucial research problems and challenges in the field of activity recognition still need further and careful consideration. In this section, we discuss important research challenges which are the key motivation for the research proposed, developed and implemented in this dissertation. We start by discussing these challenges in more details as follows:

- *Detection of novel activities:* Activity recognition models only represent a set of activities that are seen in historical data (collected for training the model). It is very realistic in activity recognition that people perform different sets of activities that may not have occurred during the training phase. Thus there is no one model that fits all activities. An activity model that is built with historical data has to have the ability to adapt by adding or removing activities that reflect the real and current set of activities performed by a particular user. The adaptation capability in activity recognition includes the detection of unseen activities and removing activities that are no longer performed by user. Recognition of novel activities includes detection of unusual activities arising from situations such as a sudden fall for elderly people. Model adaptation is an important criterion for the flexibility, effectiveness, and accuracy of any activity recognition method. Current approaches do not cater for activities that may emerge over a period of time (post the data collection and modelling) or changes in user's patterns which are both completely realistic in the context of activity recognition. There is no notion of novelty detection for activities that evolve over time. Thus, a key question is how to develop AR techniques that have the ability to adapt and learn both incrementally and continuously to reflect changes in user activities.

- *Personalisation of existing models:* Almost all activity recognition systems are built by leveraging training data from a specific set of users for the model building phase. The developed and validated models are then deployed to a large user group for serving many applications. However, a generalised model does not represent a user's personalised way of performing different activities. When applying an activity model across users, models have be tuned for each user to be able to recognise specific and personalised user activities. Unlike the detection of entirely new activities, personalisation updates the current existing model without changing the core activity types. Therefore, the personalisation process only 'tunes' current activities for a specific user at real time without adding or deleting any of the existing model activities. There is an emerging focus on the personalisation of activity recognition models [WL12, LL13] and understanding the way that people perform activities differentially. Typical walking for one user may be jogging for another. Therefore, the personalising of an activity model is significant in order to improve the recognition accuracy when deployed to large user groups.

- *Scarcity of annotated data:* Typically, there is only a small set of labelled training data available in addition to a substantial amount of unlabelled training data. The process of annotating such data or finding ground truth is tedious, time-consuming, erroneous, and may even be impossible in some cases. If training data is collected from a specific set of users, an over-fitted model will be produced. Because of the difficulties involved in annotating training data, research in activity recognition suffer from either poor or over-fitted models which could be a key reason for inefficient and inaccurate recognition. Therefore, semi-supervised, active and incremental learning are increasingly being investigated for activity recognition to overcome the limitation of scarcity of annotated data [SVLS08].

- *Streaming nature of sensory data:* Data deployed for activity recognition typically comes from sensors. The sensory data comes as a continuous sequence of streaming data. In the context of activity recognition, sensory data received from sensing devices for activity recognition requires real-time processing with limited resources using sensors or a mobile device. Dealing with streaming data imposes additional challenges in activity recognition. These challenges include handling the high speed of data,

processing data in real-time, and accommodating the change in data as activities evolve and change.

- *Situation inference:* Recognising the situation and the context of the activity in addition to the kind of activity gives complex but useful insights that can support more advanced applications. Such inference includes incorporating various sensing and reasoning components for a higher level understanding of complex activities. The automatic recognition of contextual activities is a non-trivial process. It requires a holistic approach from sensor fusion to integration of learning from different sources. The context or situation and the kind of performed activity are commonly closely tied. For example, an activity cannot be classified as running if the subject is in the middle of a lake or in a traffic jam. The combination of accelerometer data and a stream of location estimates from the GPS can recognise both the activity as well as the mode of transportation of a user and therefore help enhance the overall accuracy of activity recognition and the making of inferences about the user's situation [RB11, SRBF06, SZC13].

- *Privacy:* Maintaining privacy while processing data of activity recognition is a well-known issue [CCH+08]. The deployment of the recognition application must protect the user's privacy as well as the privacy of those with whom the user comes in contact with. Privacy is a key issue for methods in activity recognition that tend to send information to backend servers for processing. It also became a concern in applications that send data to third parties for enabling features based on the recognised activities such as in the personalised advertising scenario [SRM12]. These methods need to be particularly mindful of the need to preserve a user's privacy according to the level of consent provided or according to emerging legislation that governs the usage of personal data. Data privacy has to be considered for the three processes of collection, processing, and recognition of sensed data.

In this research, our main focus is on developing innovative research solutions to address the challenges of the *detection of novel activities*, *personalisation of existing models*, *streaming nature of data*, and *scarcity of annotated data*. The key strategy of our research is to develop techniques that are able to address these challenges by performing *incremental* and *continuous learning* from *streaming sensory data*. The developed techniques must be *accurate*, *efficient*, *robust*, and *flexible* in terms of their performance.

1.4 Aims and Objectives

In a typical activity recognition process, historical data is collected and preprocessed. The learning model is built to recognise activities that have occurred in the past. The model is then deployed to detect currently occurring activities. This approach intuitively does not cater for the recognition of 'new' and emerging activities, since the model is not trained to recognise these activities. Moreover, it fails to support personalisation of existing activities, since the deployment phase focuses on the recognition rather than the re-learning and adaptation of the model. Considering the streaming nature of sensory data, both adaptation and personalisation have to be implemented in streaming settings to be able to capture the evolving changes of activities as they emerge.

Therefore, in this dissertation, we propose, develop and evaluate new adaptive techniques in activity recognition for complementing and extending the previously learnt models with evolving data streams. The flexible learning model has the ability to learn *incrementally* and *continuously* from incoming streaming data. Typically such data streams arrive at a high rate and intensity and are processed as they come using one pass analysis techniques [GZK05]. State of the art stream learning techniques aim to discover patterns, trends, concept drifts and predictions based on continuous, unbounded data. Thus in stream mining, learning the model from the data is done using the incoming continuous stream rather than stored and collected data. Therefore, the model building occurs incrementally and continuously. This learning paradigm is the key in addressing challenges of recognising new unseen activities and personalisation of the generalised model. Additionally, as with any AR process, our approach must also address the challenge of the scarcity of labelled data by incorporating active learning capabilities.

In analysing the need for personalisation and adaptation in activity recognition, it is essential for our techniques to be operational on the device of deployment (where new sensory data are continuously generated and used for activity recognition). Thus, our proposed solutions must also be *efficient* in addition to being *robust*, *accurate* and *flexible* such that it can be deployed on a self-contained and limited resource platform such as the mobile device.

In summary, the objectives of this research are as follows:

- ***Objective 1:*** Build an efficient baseline framework for activity recognition that is accurate, flexible, robust and computationally efficient.

- ***Objective 2:*** Propose, develop and evaluate a *personalisation* technique for activity recognition for detecting realistic changes from one person to another and through *incremental learning.*

- ***Objective 3:*** Propose, develop and evaluate a novelty detection technique for recognising novel activities through *continuous learning* from sensory data streams.

- ***Objective 4:*** Handle the streaming nature of sensory data and attain real time recognition on a limited resource device.

- ***Objective 5:*** Ensure that the techniques proposed and developed in this research are effective with limited labelled data by incorporating the principles of *active learning.*

1.5 Research Contribution and Scope

The research area in activity recognition is rich with many open questions and challenges. The contribution of our research is in both the modelling and recognition processes in activity recognition. We address the aforementioned challenges in activity recognition by developing three innovative techniques. The developed techniques enables adaptation and personalisation for recognising activities from evolving data streams. The scope and contributions of this thesis according to the different AR processes/components are described as follows:

- *Modelling component*: This research proposes, designs and implements a baseline framework for activity recognition that is flexible, accurate, robust and efficient. The baseline framework (BLFW) allows adaptation and personalisation of the learning model that is initially built from the historical data. The cluster based structure of the framework enables the flexibility to add and remove activities with continuous and incremental learning.

- *Recognition component*: We develop an ensemble classifier and integrate it with our baseline framework for efficient recognition of activities. We term our baseline modelling and recognition technique "CBARS" which stands for "**C**luster **B**ased **A**ctivity **R**ecognition **S**ystem". In CBARS, we propose the use of a hybrid similarity measure approach for our ensemble classifier. The use of an ensemble classifier brings different perspectives of the data together, thus helping to achieve accurate and robust recognition especially with activities across different users.

We evaluate CBARS using benchmarked publicly-available datasets to experimentally show its efficiency in terms of accuracy and robustness for activity recognition. CBARS is developed to first establish the efficacy of our proposed AR approach. While it does not by itself support personalisation and detection of novel activities, CBARS is designed to enable achieving our research objectives. Based on CBARS, we propose two novel extensions to achieve personalisation and adaptation. Our proposed extensions enable CBARS to be operational on streaming data and learn incrementally and continuously. We also incorporate active learning in all our proposed techniques to cope with the key issue of limited labelled data in AR.

We first extend CBARS for recognising activities and implements continuous and incremental learning from data streams for recognising personalised activities. Our extension to CBARS is called STAR that stands for "STream learning for Activity Recognition". STAR presents a real time recognition and personalisation of activities that evolve in data streams. The system incrementally learns from evolving data and continuously refines the learning model for more accurate and efficient learning of the users' personalised activities.

The dynamics of our system go behind personalisation in streaming environment. We additionally aim to discover entirely new activities or remove abandoned ones in real time. Therefore, we developed our technique for novelty detection and concept evolution in activity recognition; coined COSTAR. This system monitors the evolving data for detecting novel and unusual activities in a continuous and incremental approach to adapt model accordingly in real time. Given the challenge of the scarcity of labelled data, all of our techniques incorporate active learning with recognition in order to select only a subset of data to be labelled and thus address the issue of labelling cost.

We present theoretical framework, algorithms and experimental evaluation for both STAR and COSTAR using benchmarked datasets in activity recognition. We analyse their efficiency in terms of accuracy, flexibility, and computational efficiency for recognising personalised and novel activities.

The conceptual illustration of our proposed techniques is shown in Figure 1.2. BLFW which is the base of all developed techniques is built offline in the modelling component from training data. CBARS integrates BLFW with a novel ensemble classifier for accurate recognition of activities especially across

different users. STAR enables an online extension of CBARS in streaming environment through incremental learning for personalisation. STAR aims to continuously learn from stream of personalised existing activities; COSTAR enables adaptation for detecting novel activities and incrementally assimilates the recognised novel activities with the recognition model.

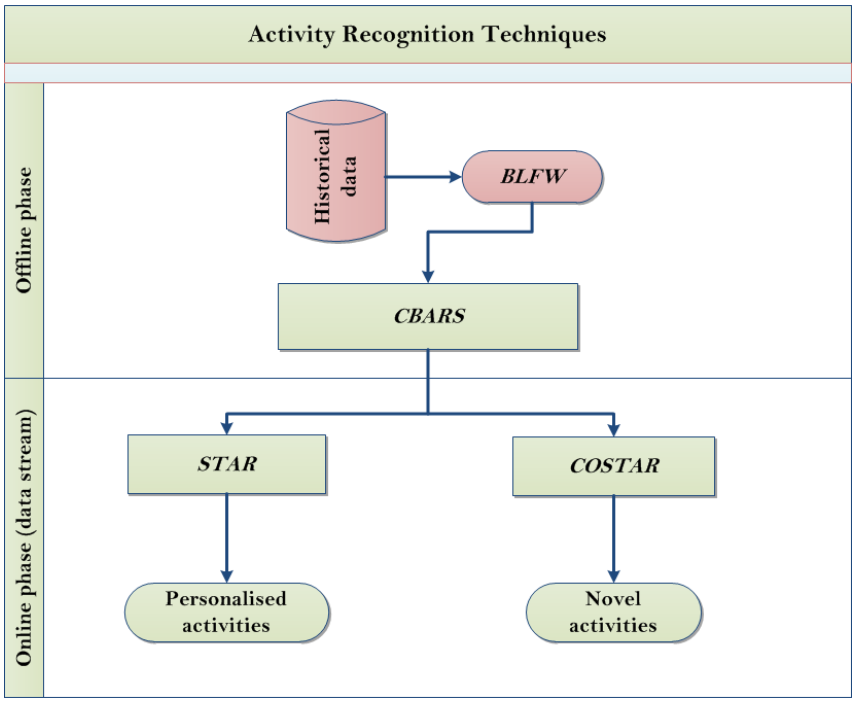


Figure 1.2: Proposed Activity Recognition Techniques

1.6 Structure of the Dissertation

The thesis is organised into six chapters. Figure 1.3 depicts the roadmap for the thesis against research objectives.

Chapter 2 gives an overview of related work in the areas of both activity recognition and stream mining. First, we review different approaches in the context of activity recognition in general. This is followed by an overview of data mining and activity recognition. Then, we review different learning techniques such as supervised, unsupervised, active, and incremental learning for activity recognition. The chapter also reviews techniques for stream learning in general and for activity recognition in particular. We conclude this chapter with a systematic comparison of the state of the art systems in activity recognition with a discussion of the research gaps and challenges.

Chapter 3 introduces the novel baseline framework for activity recognition that is built from the historical data. The baseline framework supports adaptation and personalisation by enabling incremental and continuous learning with

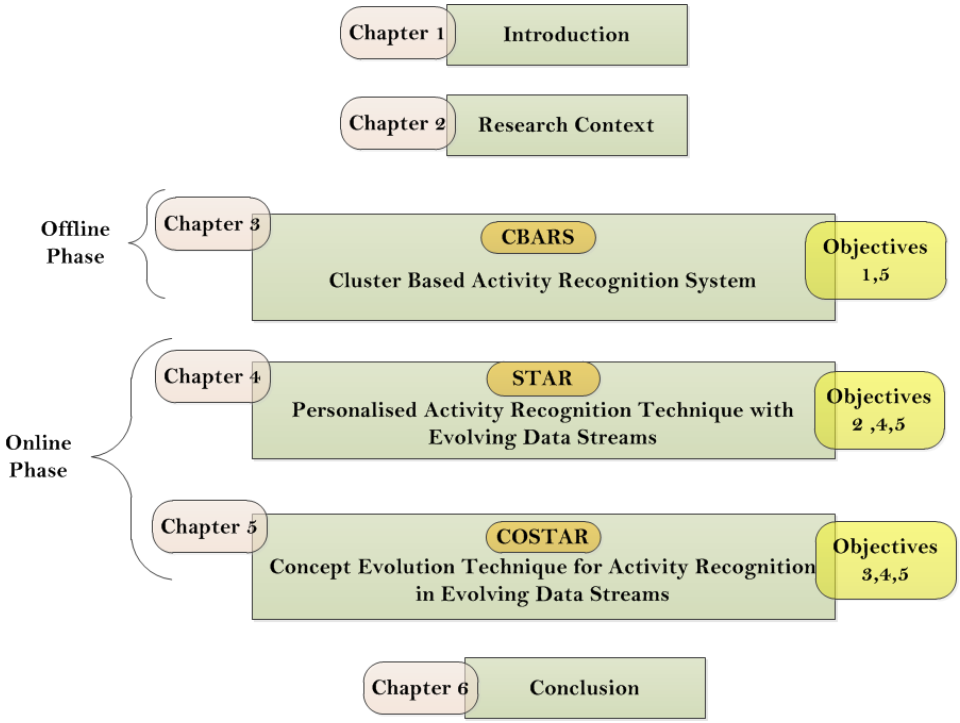


Figure 1.3: Thesis Chapters Flow-graph with the Corresponding Thesis Objectives

a robust, accurate, efficient, and flexible model. In this chapter, we provide a detailed description of the approaches for building the baseline framework. We also introduce in this chapter our novel ensemble recognition technique for classifying various activities based on the developed baseline framework: CBARS. We discuss the details of the ensemble classifier that is based on a hybrid similarity measure method. We then present an extensive evaluation for CBARS with various benchmarked and publicly-available datasets. We compare our technique performance with other state of the art techniques in activity recognition. CBARS is evaluated in terms of accuracy and robustness for activity recognition.

Chapters 4 and 5 present our proposed techniques for personalisation and adaptation respectively. First, **Chapter 4** presents our approach that applies incremental learning for personalised recognition of activities in data streams; STAR. In this chapter, we take steps towards extending CBARS for recognising personalised user activities in streaming environments. The novel approach implements a model refinement to enhance the recognition accuracy by personalising the model for a particular user. It also enables the handling of the evolutionary nature of streaming data. The model is dynamically refined and personalised to cope with the recent changes in activity data. The chapter

is concluded with an evaluation of the performance of STAR for personalisation and recognition of activities in an incoming stream in terms of accuracy, flexibility and computational efficiency.

Chapter 5 continues on the path towards dynamic learning and adaptation of the recognition component in the online phase. In this chapter, we propose a strategy for capturing the evolution of activities in a data stream and detect novel activities, COSTAR. We introduce in this chapter our novel technique for novelty detection and concept evolution for detecting novel activities in data streams. The novel technique explores to which extent we can track the evolution of different activities to be able to recognise novel activities or remove abandoned or outdated activities. We test the performance of the extended system for recognising novel activities. We provide descriptions of the approaches compared, datasets, experimental configuration, and performance metrics to evaluate the accuracy, flexibility, and computational efficiency of the developed technique.

Chapter 6 summarizes the work of the thesis, draws conclusions, and gives an outlook to possible future work.

Chapter 2

Research Context

2.1 Introduction

Activity recognition has been widely studied in the literature from many perspectives. Typical activity recognition techniques deal with data that is continuously streaming from various sensors. Sensory data is a data stream that contains unbounded data which arrives at high speed. The focus of this dissertation is on developing adaptive techniques for personalisation and adaptation form evolving data streams. Therefore, the area of this research concerns not only activity recognition but also the streaming data that evolves over time. Thus, we focus in reviewing literature of state-of-the-art approaches for both activity recognition and stream mining areas of research. We also discuss research gaps and challenges that are faced by current activity recognition approaches with data streams. Our research contribution for addressing these challenges are also summarised.

We start this chapter by introducing an overview of the general concepts of activity recognition in Section 2.2. In this section, we discuss the different types of sensors employed for recognising performed activities. The scope of our research across activity recognition and stream mining disciplines is explained in Section 2.3. Within this scope, we first discuss different approaches for activity recognition in Section 2.4. This section investigate techniques that are applied for recognising activities. Then, section 2.5 reviews state-of-the-art techniques for stream mining from sensory data. Recent research work focuses on activity recognition in streaming environments. Section 2.6 represents the research area comprising the intersection of stream mining and activity recognition approaches. Section 2.7 surveys key activity recognition systems. We further present in Section 2.8 current research challenges and gaps, along with

a comparison of key systems. The chapter is summarised and concluded in Section 2.9.

2.2 Overview on Activity Recognition

Activity recognition is one of the emerging applications in the area of ubiquitous computing. Systems that can recognise human activities opened the door to many important applications in the fields of healthcare [TF08, DLL13, WCZZ08, MMEJ08, ZHM08, STF08], social networks [MLF+08], environmental monitoring [MRS+09], surveillance, and emergency response [ZHM08, NCM04]. Human activity recognition has been a fast growing research area. Monitoring and recognising activities are performed using two main approaches: vision based and sensor based. Early work in activity recognition focused on analysing visual information such as images and videos [Pen00, Gav99]. Since low cost sensors are embedded pervasively in the environment, the sensor based approach provides an effective alternative for activity recognition. Thus, recent research has moved to the use of the rich sensory data with the sensor based approach. As early as 1999, the TEA project [SAT+99] implemented a prototype that demonstrated the feasibility of recognising activities with low-level sensors.

2.2.1 Sensing platform

A wide range of sensors are employed for activity recognition. A sensing platform includes either one sensor or a combination of sensors. We discuss the sensing platform from two perspectives: the kinds and the placement of the deployed sensors. Several kinds of sensors are used for collecting data for activity recognition. That includes accelerometer sensors [BI04, RDML05], RFID tags [PFP+04, KRT+05], state change sensors [TIL04], microphones [WLTS06], location based sensors such as GPS data [LFK05, RB11], and light and temperature sensors [KZX+11]. Much existing research has explored inferring a user's activities from accelerometer sensors only, such as in [BI04, LCK+05, RDML05]. Figure 2.1 shows a subset of the sensors that are deployed for activity recognition.

Sensor placement categorises sensing approaches as wearable sensors, mobile sensors or ambient sensors. Wearable sensors allow data collection over a long period of time while being attached to the human body. Many activity recognition studies focus on the use of accelerometer sensors attached to a user's body to recognise different activities as in [MS10, hLKK+09, FF00,

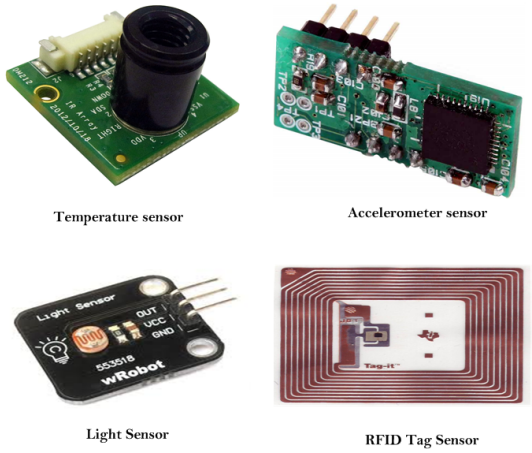


Figure 2.1: Example of Sensors for Activity Recognition

LYA09]. Bao and Intille [BI04] deployed five biaxial accelerometers worn on the user's body positions in order to distinguish not only the whole body movement like walking or running, but also those activities involving partial body movement such as standing still, folding laundry, brushing teeth, watching TV, or reading. Some other studies combined multiple types of sensors in addition to accelerometers for activity recognition. The SATIRE system [GJAS06] used a combination of accelerometer and GPS sensing data embedded into "smart clothing" to identify different user's activities. Maurer et al. [MSSD06] implemented the eWatch sensing platform to detect activities from a dual axes accelerometer, a light sensor, a temperature sensor and a microphone. Other studies as in [LM02, SRBF06, CCH+08, CNCC08] and [LCB06] deployed different combinations of sensors including velocity, compass, image and temperature sensors in additional to accelerometers.

Although the wearable sensor systems have been deployed widely in the field of activity recognition, mobile phone sensors have recently gained popularity in this field for many reasons. Today's smart phones come with a rich set of embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone, and camera. The merit of using mobile embedded sensors is that we do not need to deploy additional devices. Figure 2.2 shows an example of embedded sensors in today's smart phone. Since people carry their personal companion devices all the time and have full control of their own devices, those devices will not make the users feel intrusive. Furthermore, the mobile phones are becoming increasingly intelligent and powerful. Thus they appear to be the ideal platforms for detecting people's activities. Importantly, the popularity of mobile phones is allowing researchers to deploy and test the developed models on a large scale of users, which consequently enables the collection and analysis of data far beyond what was previously possible [LML+10].

Figure 2.2: Example of Embedded Sensors into Mobile Phone

Researchers have considered the use of widely available mobile phones to address the activity recognition problem. However the earlier approaches did not leverage the sensors incorporated into the mobile devices themselves. For example, Győrbíró, Fábián, and Hományi [GFH09] used a band ("Motion Band") that was attached to the user's body. The Motion Band contains an accelerometer, a magnetometer, and a gyroscope. The data collected by Motion Band was transmitted to a smart phone carried by the user and stored for further analysis. Ravi et al. [RDML05] collected data from users wearing a single accelerometer based device and then transmitted this data to the HP iPAQ mobile device carried by the user. These studies have not considered the use of mobile phones as a sensing platform. Instead mobile phones have been used for storage and data analysis purposes.

Few studies employed the actual commercial mobile device to sense and process data for activity recognition. Such systems offer an advantage over other systems because they are unobtrusive and do not require additional equipment for data collection and recognition. Miluzzo et al. [MLF+08] explored the use of various sensors (such as a microphone, accelerometer, GPS, and camera) available on the commercial smart phones for social network applications. The Nokia N95 phone as used in [Yan09] constructs physical activity diaries for the users. While Kwapisz, Weiss, and Moore [KWM11] collected accelerometer data on android based mobile phone to identify activities.

Ambient sensors are neither wearable nor embedded on a portable device. These include sensors that are deployed to track motion, location, or object interaction in a smart home setting [TIL04,NCM04]. Examples of ambient sensors that are deployed for activity recognition in smart environments include infra-red motion sensors, temperature sensors, switches, smart power meters, and pressure sensors. The major limitation in applying the ambient sensing platform is the fixed laboratory settings required to fulfil the recognition of

activities. Smart environments limit the recognition of activities to only indoor settings. Furthermore, the cost of setting up sensors and the annotation involved are mostly impractical.

The data collected form various sensing platforms are processed by different techniques for recognising activities. The recognised activities vary from atomic basic activities to complex activities.

2.2.2 Kind of activities

Activity recognition systems aim to understand the set of activities that users perform from low level sensory data. Various activities could be predicted by the recognition system. *Atomic activities* are the basic activities that can not be broken down to any simpler activities. Examples of atomic activities are walking, standing, and running. The recognition of atomic activities gained an immense focus in the literature [CK11]. Other approaches aim to recognise *fine grained activities* such as gestures by attaching sensors to specific human parts as in [WLT05, ZLS+08]. Researchers have applied sensors to detect gesture activities such as finger movement, leg movement, or upper body movement. Whereas *complex activities* integrate atomic activities into contexts of activity such as in [SZC13, HFS08]. For instance, the activity of 'sitting in the train' is different from 'sitting on the couch at home'. 'Sitting' is the atomic activity while the complex activity includes the context which offers a deeper understanding of the activity. Tapia, Intille, and Larson [TIL04], for instance, aimed to recognise complex activities such as preparing lunch, dressing, and bathing from sensors installed in smart home settings. While Yang [Yan09] recognised atomic activities such as sitting, standing, and walking from accelerometer sensors on the mobile device.

In a nutshell, activity recognition approaches are either sensor based or vision based. The sensor based approach for activity recognition aims at predicting various types of activities from low level sensors. These activities include atomic, fine-grained, or complex activities. A wide range of sensors are deployed for data collection in activity recognition. These sensors are wearable sensors, mobile sensors and/or ambient sensors.

2.3 Research Scope

Activity recognition is a very wide research area that has been investigated from many perspectives. A well investigated research perspective focuses on

managing the process of data collection. This perspective concerns issues related to the kinds of sensors and sensing platforms. Yet, immense research has been directed towards learning methods for recognising activities from sensory data. The literature covers a wide variety of learning techniques applied for activity recognition. This research focuses on learning yet in streaming environments with incremental and continuous learning approaches. The focus of this research is on building dynamic techniques that can cope with the evolving nature of the sensory data streams and achieve recognition in real time. As we focus on the methodology of analysing sensory data in streaming environments, we aim also to develop sensor and platform independent techniques that can leverage the wide range of sensors deployed for activity recognition. The developed techniques have to be computationally efficient to be deployed on limited resources devices for real time recognition. Although the literature represents subsets of our research theme across activity recognition and stream mining disciples, this dissertation primarily focuses on the holistic approach that merges both activity recognition and stream mining for building adaptive techniques.

In the light of our research scope, we introduce different learning techniques for activity recognition with data streams. To explain the scope of our research, we first introduce the known term of *i.i.d.* in statistical and probability theory, which refers to data that is both independent and identically distributed. A key challenge with learning from sensory data in a streaming environment is that it requires learning beyond identically distributed and independent conditions. In machine learning, a collection of data D is defined as independent and identically distributed if all samples in $D : X_1, X_2, ...X_n$ follow the same distribution function f and are independent of each other.

Many state-of-the-art statistical and machine learning approaches rely on processing the data that follows the *i.i.d.* conditions. These approaches assume that data instances are independent and follow the same distribution. Thus, the prediction of new data relies strongly on prior knowledge. In the ubiquitous environment, either of the *i.i.d.* conditions or even both can be violated. The distribution condition is clearly challenged with the basic concepts of data streaming. In a streaming dynamic environment, we can not assume the fulfilment of the identical distribution condition, as a typical data stream evolves over time either for small or abrupt changes. There is an increasing interest in the domain of learning from data that is not identically distributed. Research into non-identically distributed data streams addresses slow changes in a data stream which is known as concept drift. Sudden change is addressed with novelty and outlier detection techniques. Indeed, these methods still

preserve data independence. Other areas of research focus on the violation of the independent data condition. These approaches deal with data that is dependent while the distribution is fixed. Stationary time series and Markov Chains are examples of approaches that challenge the dependency assumption.

Activity recognition data that represents a sequence of performed activities is intuitively dependent. Therefore, an efficient activity recognition system focuses on dealing with the dependency among data for predicting performed activities. Furthermore, in a streaming environment, activity recognition violates not only the assumption of independence but also identical distribution. Changes in activity recognition data include either small or abrupt changes. Sensory data deployed for activity recognition has the same characteristic of non-identical distribution that is known in data streaming. Examples of small changes are a change of "walking" or "running" pattern for one user, or changes in ways of performing activities across users. Abrupt changes in activity recognition include the detection of novel activity such as "exercise at the gym", or detecting sudden activities such as "fall detection" and malicious behaviour. Recognition of changes in activities is crucial especially with applications in healthcare, surveillance, and security. Figure 2.3 depicts the intersection of the the two research areas of stream mining and activity recognition. The intersection between the two overlapped research areas is where the underlying scope of our contribution in this thesis. In the following sections, we survey methods for both activity recognition and data stream mining as well as the intersection between them.

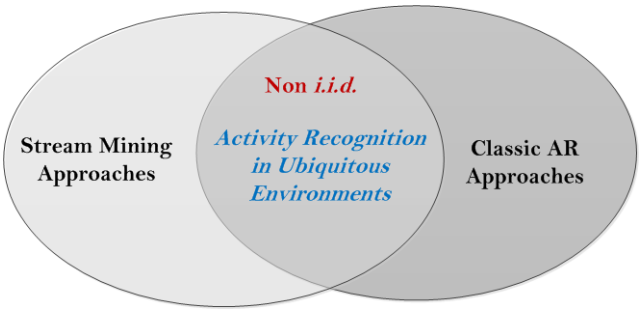


Figure 2.3: Scope of Research

2.4 Activity Recognition Techniques

We survey in this section the main body of research in activity recognition. Different from traditional probabilistic and logical approaches, other approaches that consider activity recognition from the machine learning perspective have

been studied. In general, approaches in machine learning for activity recognition can broadly divided into two major strands. The first strand concerns the underlying learning approach. That includes supervised, unsupervised, and semi-supervised learning. The second strand focuses on the dynamic capabilities of the recognition system beyond the learning phase. This includes personalisation and adaptation of a learning model and the concept of transfer learning.

2.4.1 Learning approaches

Most of the studies considered deploying supervised machine learning algorithms such as Naive Bayes, Decision Trees, Hidden Markov Models, Nearest Neighbour, Support Vector Machines, and different Boosting techniques. Moving beyond fully supervised settings, researchers have started studying the feasibility of other machine learning techniques for activity recognition: unsupervised and semi-supervised learning.

Supervised learning is applied pervasively for activity recognition. Basically, labelled data is collected to train a static classificatory model for recognising a set of activities. The classificatory model built from labelled data is used to recognise the incoming unlabelled data. Figure 2.4 explains the supervised learning process. Supervised learning is categorised as either generative, discriminative or hybrid approach [RH+97]. Algorithms that follow the generative approach models the class conditional distribution. Naive Bayes is an effective generative approach that is applied pervasively for activity recognition such as in [RDML05]. Hidden Markov Models are also generative methods that have been successfully applied for recognising activities [PFKP05, WLTS06]. Discriminative models on the other hand learn the boundaries between classes. Decision trees [BI04,LHP+07] and nearest neighbour [MSSD06,LM02] are well-studied examples of the discriminative approach for activity recognition. Moreover, a hybrid approach is the one that combines the two approaches into a single classifier. Viola and Jones [VJ01] applied a modified version of AdaBoost that combines a set of static classifiers for recognising activities. While Lester et al. [LCK+05] demonstrated the efficiency of combining discriminative and generative classifiers for smooth recognition of activities. The hybrid approach discriminatingly selects useful features and learns an ensemble of static classifier to recognise different activities. In recent research, a hybrid classifier was developed that combines both threshold based methods and machine learning methods to select the most suitable classifier dynamically on the cloud [YH14]. Artificial Neural Networks rely on a hybrid generative-discriminative approach.

Do, Loke, and Liu [DLL13] developed a system based on stream reasoning and Artificial Neural Networks for recognising activities from mobile phone sensors. Given the promising performance of the hybrid methods, our research in this dissertation is also based on the same approach but in a streaming environment that supports incremental and continuous learning.

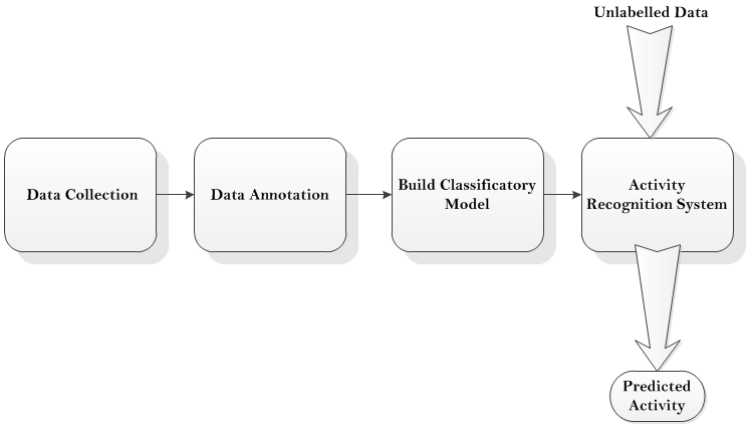


Figure 2.4: Supervised Learning in Activity Recognition

A wide range of supervised methods commonly used for activity classification was reviewed in [PGK+09, PPGD14]. One main characteristic of these methods is the necessity of a significant amount of labelled data to build the classificatory model. The assumption that labelled data is consistently available is unrealistic. Due to many reasons, activities may vary while time evolves for the same person or across different individuals. For each activity, data is required to be collected for each user to attain an accurate recognition. However, the annotation process is a time consuming,error prone, and mostly tedious process. Therefore, researchers investigated an unsupervised learning approach for activity recognition to overcome the limitations raised from the supervised approaches.

The main goal of ***unsupervised learning*** techniques in activity recognition is to discover variation and likelihood among data. There are a few researchers who studied unsupervised learning for activity recognition such as [LLPK09, WPC05, LD11, HS06]. Lee et al. [LLPK09] used unsupervised learning for abnormality detection. To detect whether a pattern is registered or not, a probability model based on the past activity pattern is created. The Expectation-Maximisation (EM) algorithm is applied with the feature vectors to decide whether the activity has abnormal behaviour. In another study, Li and Dustdar [LD11] studied the feasibility of applying a specific type of unsupervised learning to high-dimensional, heterogeneous sensory input. The correspondence between clustering output and classification input is proposed

as well. Although clustering is promising approach in discovering data structure and patterns, at least a few labels have to be provided for performing the actual recognition of activity. Also, the feasibility of traditional clustering is questionable for high dimensional streaming data [LD11]. Furthermore, methods for unsupervised learning require a large pool of unlabelled data in order to find interesting patterns. Figure 2.5 depicts an overview of the unsupervised learning process.

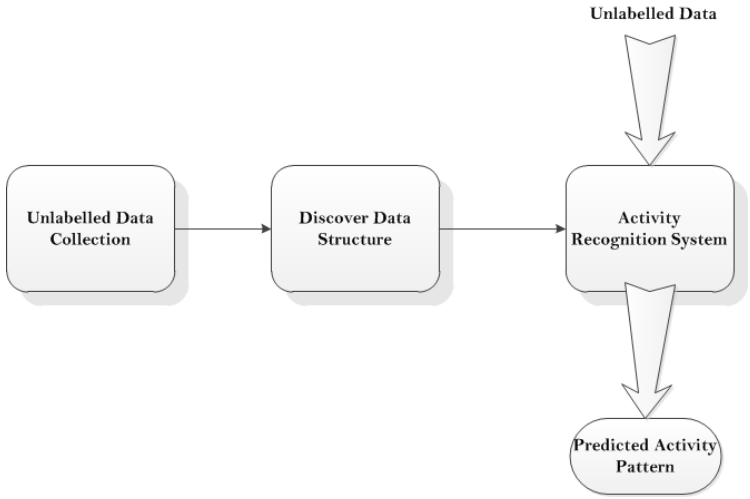


Figure 2.5: Unsupervised Learning in Activity Recognition

In realistic conditions, large amounts of unlabelled data can be easily collected while small set of labelled training data is available. In order to integrate the advantages of both supervised and unsupervised learning; the concept of ***semi-supervised learning*** is considered for activity recognition. A semi-supervised approach requires less labelled data for recognising a substantial amount of unlabelled data. Figure 2.6 represents an outline of the semi-supervised learning approach. The ability to use unlabelled data for enhancing the recognition system became an interesting topic for many researchers. Self-learning and co-training are attractive paradigms in semi-supervised learning. In the self learning paradigm, a small amount of annotated data is used to build the classificatory model for later prediction of the unlabelled data. The predicted label with highest confidence is added to the training seed for rebuilding the classificatory model. Self learning has been successfully applied in many applications such as text analysis [Yar95] and image processing [LWFF07]. Continuing research is being conducted to study self-learning for activity recognition. Longstaff, Reddy, and Estrin [LRE10] investigated methods of further training classifiers after a user begins to use them using semi-supervised learning techniques.

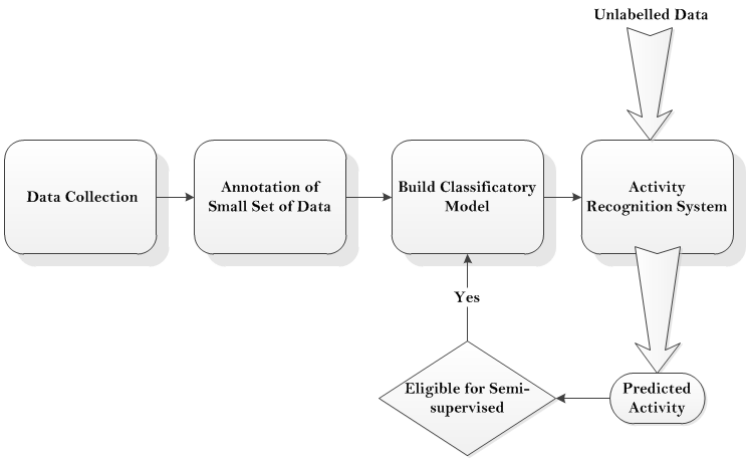


Figure 2.6: Semi-supervised Learning in Activity Recognition

Co-learning was first developed by Blum and Mitchell [BM98]. It uses two classifiers, each trained on different perspective of the data. Each classifier adds its most confidently predicted label to the training set and rebuilds the model. Guan et al. [GYL+07] applied co-learning with three classifiers from the same view of data. They also used majority voting to choose the most confident label to augment the training data. Stikic, Laerhoven, and Schiele [SVLS08] applied various semi-supervised techniques for enhancing accuracy of recognising users' activities.

Despite the ongoing research on semi-supervised learning, most of the developed techniques rebuild the whole classificatory model upon predicting the most confident label. Yet, this is impractical for real time recognition in streaming settings. Also, the focus of aforementioned methods was mainly to minimise the labelled data required for building the initial classificatory model rather than improving the classifier itself, especially with confusing data [LRE10].

Active learning is another approach in the semi-supervised learning category. Unlike self learning and co-learning, active learning requires user input to label data with the true label. According to Muslea, Minton, and Knoblock [MMK00], the main goal of active learning algorithm is to find the more profitable and less costly data to label. Kapoor and Horvitz [KH08] compared different methods to decide upon the selection of profitable data. While Stikic et al. [SVLS08] employed a multi-sensor approach to choose important data to be labelled. The data is selected for active learning based on two approaches. One approach chooses the data with the lowest classification confidence. The other approach chooses the data that causes a high degree of disagreement between two classifiers. Results showed improved performance when active learning is applied. The ongoing research in active learning still

requires further improvement. State-of-the-art active learning techniques in activity recognition assume the retraining of models. This is not applicable in streaming settings. Alternatively, dynamic update to tune the classificatory model in real time are crucial in a streaming environment. Improving active learning methods to consider the value of label compared to the cost of interrupting the user is essential in the context of activity recognition. Also, estimating the time taken to provide true labels and their effect on the prediction accuracy in real time requires more investigations.

In this section, we have reviewed the three main approaches of activity recognition according to the underlying learning approaches. Traditional activity recognition systems are based on fully supervised learning approaches. However, these techniques require all data to be annotated which is impractical, especially when applied in a streaming environment. On the other hand, unsupervised learning finds pattens in unlabelled data. Unsupervised learning is not capable of finding the actual predicted label of activity without at least some labelled data presenting the ground truth. Therefore, semi-supervised learning is a recent trend of activity recognition that employs only a small set of labelled data for training. In semi-supervised learning, the recognition system continuously learns from unlabelled data either automatically with self learning or similar approaches or interactively with user input via an active learning approach.

The research work represented in this dissertation is uses active and semi-supervised learning in a streaming environment. The three developed techniques in this dissertation build an initial classificatory model from a small set of annotated data. The primitive model is incrementally updated while sensory data evolves. We extend the literature in activity recognition by introducing adaptive techniques that address the evolving nature of activities and scarcity of labelled data. In the following, we discuss the dynamic capabilities of the learning model beyond learning. That includes personalisation and adaptation of the learning model.

2.4.2 Dynamic capabilities

Moving forward from learning approaches in activity recognition, we review in this section post learning dynamic capabilities. In this section we aim to review the dynamic capabilities of the recognition system according to the objective of a system update. In the literature, two reasons urge model update beyond the learning phase: model *personalisation* to best fit a specific user; model *adaptation* by adding new activities or deleting abandoned ones.

The wide concept of *transfer learning* is also presented to explain the different kinds of anticipated changes in activity recognition and the proposed techniques to handle these changes. We represent, in the following, the literature contribution to each category.

Model personalisation

It is hard to generate one learning model that fits all users in activity recognition. Due to many reasons, different individuals might perform the same activity but in different ways. A "walking" activity for one person, for instance, might seem to be "running" for another. The most accurate recognition results can be obtained if we train the learning model with the annotated data for a specific user. This assumption is invalid in a ubiquitous environment when labelled data is scarce. Therefore, continuous learning approach for tailoring the model to best fit a specific user is crucial for improving recognition accuracy. We define model personalisation as *the process of tuning the general model to represent a user's personalised way of performing different activities.*

The vast majority of activity recognition research did not consider the personalisation issue. Only few studies investigated the impact of training model on personalised data and compared it to training the model on general data collected from different users. The researchers proved the improved accuracy when deploying subject-specific data for training instead of the general model [KWM11, WL12]. Weiss and Lockhart [WL12] demonstrated the improved accuracy if a personalised model is deployed even using only a small amount of user-specific training data. Whereas Kwapisz et al. [KWM11] created models individually for each user, then deployed a personalised model for recognition.

Due to the sparsity of labelled data for a specific user, a more practical approach is investigated to tailor a general model to best fit a specific user. In [ZCL+11], the authors developed an algorithm that learns a binary decision tree model for one person from his labelled data, transfers its structure to another person, and automatically adapts its non-determinate nodes with the unlabelled samples of the new person. This accomplishes the cross-people knowledge transfer task. Pärkkä, Cluitmans, and Ermes [PCE10] proposed a similar approach based on a binary decision tree. In this method, the user's input is required to tune tree thresholds for a specific user. Moreover, it takes 3–10 minutes of new data with annotation and uses that for updating the thresholds in each node. The problem of activity recognition is more challenging with multi-dimensional data in a streaming environment. Therefore, the binary decision trees applied in the aforementioned studies are not efficient

with large scale data and complicated scenarios. Gomes et al. [GKG+12a] constructed a personalised activity recognition system that is deployed in a streaming environment. Despite the efficiency of the developed system, their model still required user specific annotated data to achieve personalisation. Similarly, Reiss and Stricker [RS13] used a set of classifiers as a general model which is later updated with new labelled data from a specific user. Recent research in [VHC13] adjusted the learning model from person A with a selected confident sample for another person B. The proposed algorithm is an integration of an SVM classifier and clustering approach for updating a model automatically. However, the proposed system has not been evaluated in a streaming setting. The deployment of activity recognition system in streaming environment imposes more challenges as the change of data distribution while a stream evolves may cause the model to drift away from the actual data distribution.

Model adaptation

The other dynamic feature of the recognition system concerns its ability to capture significant data changes. It is impractical to assume that there is always a static set of activities along evolving data streams. In the recognition system, the initial data represents a set of activities that is collected to train the primitive model. Then, the learning model is deployed for the actual recognition of incoming unlabelled sensory data. However, state-of-the-art approaches in activity recognition do not consider the appearance of new activities that did not exist in the initial training data [YLLB10]. New activities appear because of two common reasons. First, it is unrealistic to collect annotated sensory data for all kind of activities that exist in a domain. A typical activity recognition system contains only a few activities that are annotated with experts or with a means for assistance such as videos. Other activities may appear beyond the learning phase. Therefore, the set of activities may need to be extended later after deployment because of the scarcity of annotated data in the learning phase. Second, novel activities include also sudden activities. It is impractical to collect data represents sudden activities for training the model. An example of this kind of sudden activity is a sudden fall for an elderly person in the application of health care. Due to the difficulties of collecting annotated data for such sudden activities, other approaches have to be considered for detecting novel activities. Traditional methods solve this problem by rebuilding the entire model based on the training data of the new set of activities. This is impractical for real time activity recognition and especially in a streaming environment.

Developing a recognition system that can recognise new activities and assimilate it with existing model for further recognition is essential for real life recognition. The same concept applies for the removal of abandoned activities that are no longer relevant to a particular user. Model adaptation is a key criterion for the flexibility and accuracy of any activity recognition system with an evolving data stream that changes over time. There is no notion of adaptation/refinement of the learning models in the literature. Models do not detect activities that may emerge over a period of time (post the data collection) or changes in a user's patterns, which are both completely realistic in the context of a mobile user. The adaptation process needs to update the recognition model to recent changes in a real life user's activities in real time. Different from model personalisation that only tunes existing models, model adaptation enforces changes to core activities with incremental assimilation of new discovered activities or elimination of abandoned activities.

Transfer Learning

A major assumption in activity recognition is that the underlying concepts of both training and target (deployment) are the same. This assumption is invalid in many cases. Data collected for training in activity recognition may be different from data received in actual recognition in terms of distribution, domain and tasks. Transfer learning concerns "*developing systems that can leverage experience from previous tasks into improved performance in a new task which has not been encountered before*" [CFK13]. The power of transfer learning is in the flexibility in mapping between training and deployment. Thus, we can reuse knowledge learned previously to solve a new problem faster or in a more efficient way. Figure 2.7 represents an illustrative figure of transfer learning.

Figure 2.7: Illustration of Transfer Learning

The concept of transfer learning has attracted more attention recently in machine learning and data mining. Pan and Yang [PY10] categorised transfer learning techniques into four approaches based on what is transferred. The approaches are instance transfer, feature representation transfer, parameters transfer and relational knowledge transfer. In instance transfer, a subset of training data can be re-weighted to be used in the target domain [Zad04,

HSG+07]. While feature representation transfer aims to adapt the feature space of the training domain to attain the best performance in the target domain [RBL+07]. The third approach assumes common parameters across training and target domain [LP04, GFJH08]. Therefore, prior knowledge of parameters would be transferred between training and target domains. Lastly, the relational knowledge transfer approach discovers relationships among the training data and transfer this knowledge to the target domain [MM08]. In the context of activity recognition, the meaning of the term *transfer based activity recognition* is based on traditional transfer learning concepts presented by Cook, Feuz, and Krishnan [CFK13]. Traditional approaches for activity recognition assumes that training and target domains are typically identical. However, this assumption is not valid due to many differences that might exist between training and target domains which include the following:

- Sensors applied for data collection in the training domain are different from sensors applied in deployment. Few studies have addressed knowledge transfer between different sensor modalities such as [KHF+11, RFCT13].

- Domain of training is different from target domain. This difference can be in the form of a feature space [ZPYP08], data distribution [HSU12, RC09], label space [HY11, WP11, ZHY09] and/ or predictive function [VKP10, ZHY09].

- The availability of data is different between both training and target domains. The availability of data in the training domain categorises the learning approaches to supervised, unsupervised, and semi-supervised learning, while availability in the target domain represents the informed and uninformed terms introduced by [CFK13].

In our research, we focus on the latter two differences: domain and data availability. The research scope in this dissertation covers techniques for transfer learning between static environment for data collection in the training domain and streaming environment in the deployment domain. This includes changes of environment distribution as well as change of data availability between training and target domains. The essential model adaptation and personalisation to cope with the domain change is the focus of our research. The capability to learn incrementally with the scarcity of labelled data in the deployment domain is also one of our key objectives.

In the next section, basic concepts of the deployment domain of stream environment are presented to introduce the developed challenges. These challenges of activity recognition that arise from the streaming environment are described in Section 2.8.

2.5 Stream Mining Techniques

Traditionally, data mining and machine learning techniques are used to analyse sensory data. Thus, data processing in machine learning is applied to the collected data for prediction. Data is initially collected to train the offline model which is deployed later for predicting unseen data. Unlike traditional machine learning techniques, streaming data is a high speed, unbounded, and fast changing flow of data. In order to discover knowledge from data streams, methods and approaches that are tailored to the nature of streaming data have been developed.

A common assumption applied in most machine learning techniques is that data is independent and identically distributed. Machine learning techniques applied for activity recognition are presented in Section 2.4. However, these techniques do not consider the invalidation of the identically distributed features. In a dynamic streaming environment, data distribution might change over time while a stream evolves. The change in data stream distribution is known as *concept drift* that is first introduced by Schlimmer and Granger [SGJ86]. Learning in a data stream imposes more challenges than a static machine learning environment. Several data stream studies outlined these challenges such as [GvB+14, GGK+14, GRSdC10, ZBG+12]. We summarise some of these key challenges as follows:

- *High speed of data streams:* A very significant challenge in a data stream is how to deal with its high speed and high volume in real time [GR09]. A non stopping sensory data that evolves from various sensing sources at high speed mostly requires one pass to scan data for real time prediction. The time taken to process new streaming data is constrained to both data sampling rate and available memory to store data until reaching a decision.

- *Unbounded memory requirement:* Due to the unlimited nature of a data stream, it is infeasible to accommodate such unbounded data in memory for later processing. It is also impossible to scan through a data stream multiple times due to its tremendous volume. Many approaches have been developed to address this issue. The most common approaches are

sampling, load shedding, sliding window, and data synopsis [BDM03, ZS03, WK96].

- *Real time and accurate prediction:* The crucial challenge in stream mining is how to maintain the tradeoff between real time prediction and accuracy. In many approaches, algorithms that attain a reasonable level of accuracy are those of high space and time complexity. Whereas approximation algorithms mostly guarantee quick convergence, yet with lower performance efficiency. A key success in data stream mining is to keep the balance between the algorithm efficiency in terms of accuracy and real time prediction.

- *Concept drift:* This key term refers to the continuous change in the relation between the training and the target data over time [GvB+14]. Concept drift includes changes that occur gradually over time or abrupt changes. To cope with these changes, many approaches that apply incremental and continuous learning have been developed. The aim of these techniques is to adapt the model to accommodate for the most recent changes in the data stream.

- *Concept evolution:* The essential issue of "concept evolution" in data streams refers to the appearance of novel classes while streams evolve. Traditional machine learning techniques are unable to detect these novel classes and consequently misclassify all instances representing the novel concept. In order to cope with concept evolution in data streams, the mining technique has to be able to detect a novel concept as soon as it arrives without being trained with labelled data. That also followed with assimilating the novel concept into the underlying concept for further detection of recurring novel class instances.

- *Concept forgetting* [GvB+14]: Learning from data streams with concept drift requires not only the detection of novel concepts but also forgetting outdated and abandoned concepts. Close monitoring of the evolution of data over the time facilitates the detection of concepts that became irrelevant by the time. Existing concepts could disappear either gradually or suddenly. Moreover, it is hard to distinguish between totally abandoned concepts or less frequently detected/occurred.

The aforementioned challenges concern data stream mining in general. In the following subsections, we outline various techniques proposed in the literature to handle different challenges in data stream. Two categories are illustrated. The first category focuses on a mining data stream in terms of state-of-the-art techniques for clustering, classification, etc. The other category is for mining changes in data streams which includes techniques for detecting changes, novel concepts, and outliers. Section 2.6 represents stream mining techniques that are applied for activity recognition.

2.5.1 Techniques for handling data streams

In order to process high speed data streams, new adapted approaches have been developed that are capable of handling streaming data for real time prediction. *Sampling* is a traditional statistical approach that uses a probabilistic choice for selecting a subset of data to be processed. Despite the simplicity and the bounded error rate in sampling approach, it is very sensitive to outliers. Another approach for scanning data stream is a *sliding window*. This window could be either fixed size or variable size. Fixed size contains a fixed number of the most recent data instances. Whereas a variable size window holds varying numbers of instances depending on other parameters such as change detection. As early as 1996, Wimber and Kubat [WK96] developed the *FLORA* algorithm for stream mining that applies a fixed size sliding window. The window contains the most recent data with a first in first out approach (*FIFO*). The problem with a fixed size window is the decision made upon window size. A small sized window would allow more frequent monitoring of data stream, yet it might negatively affect the algorithm performance with unnecessary processing. On the other hand, a long window preserves the limited resources with less frequent processing. However, it takes longer to show the reflection of changes detected when applied. Thus, that leads to poor performance because of slow adaptation. A variable sized window adjusts window size based on a criterion. For example, Klinkenberg and Joachims [KJ00] built various Support Vector Machine models with different window sizes to select the window size with minimum generalised error dynamically. A sliding window technique assumes that recent data is more important than any other data seen in the stream. Yet, this assumption is not always valid, especially when dealing with noisy data that contains outliers.

Another technique that has been applied pervasively for querying data streams is load shedding [BDM03, TÇZ+03]. The basic idea in load shedding is to drop chunks of data to reduce the system load. For efficient deployment

of this technique, various methods have been developed for determining the magnitude and location of drops. Although this technique showed a good performance in querying data streams, the drawback of dropping important parts of the data might negatively affect the performance. Another efficient technique for handling data streams is the use of a synopsis data structure. In this approach, summarisation of incoming stream is computed for further analysis. Histograms [GGI+02] and wavelets [ZS03] are two approaches for synopsis structure. Since a synopsis represents only a summarisation of a data stream, the expected output when applying this technique is approximate. Also, it lacks efficiency when dealing with a very fast stream.

2.5.2 Mining data streams

Data stream mining is conceptually an extension for static data mining, but in a streaming environment. New approaches have been developed to extract meaningful information from streaming data. In this section, we review clustering, classification and time series analysis and sequential data stream mining techniques.

Clustering techniques

The basis of clustering technique is to process unlabelled data in order to categorise similar data into groups. Each group contains a partition of data that attains the maximum into-cluster "similarity". Under some definition of similarity, similar data are gathered into the same cluster. Clustering approaches for data streams have been reviewed by many researchers. Guha et al. [GMMO00, GMM+03] proposed a clustering technique based on K-median. The approach is multi-level with divide-and-conquer that aims to achieve a constant factor approximation in small space. Many techniques have been developed to enhance the Guha et al. algorithm such as [BDMO03] that used exponential histogram to aggregate similar clusters. STREAM [OMG+02] is another technique that is based on K-median. A STREAM algorithm processes data in batches to generate k clusters in each batch. It then summarises information about the k clusters and their weights. The collected weighted centres are then re-clustered to produce another set of clusters that fits into memory. Merging online and offline approaches has been studied for clustering. Aggrawal et al. [AHWY03] proposed CluStream algorithm that divided the clustering process into two components: online and offline. It applied a synopsis approach to store summary statistics about the data stream in the online components. The summary of statistics is stored in microclusters. The

offline component clusters the summarised information based on the user's queries. Microclusters are continuously updated with the recent changes in the data stream. Gaber, Krishnaswamy, and Zaslavsky [GKZ04] developed a light weight clustering technique that adjusts to the available resources by monitoring input and output rate. When the upper memory bound is reached, existing clusters are merged to generate another set of clusters.

Classification techniques

Data stream classification has been an interesting research topic for years with many approaches developed in this area of research [GZK07]. Classification is a supervised learning technique that aims to predict unlabelled data based on a set of previously collected labelled data. Wang et al. [WFYH03] developed an ensemble classifier for data stream. The ensemble classifier calculates a weight for each classifier that controls the classifier contribution to the final decision. Very Fast Decision tree [DH00, HSD01] is a technique based on Hoffding trees that splits a tree according to the decision of the best current attribute. The algorithm periodically drops non-potential leaves and attributes in order to bound memory requirements. Another iconic classification technique in stream mining is On demand classification [AHWY04]. On demand classification follows the same approach of two components proposed in ClusStream [AHWY03]. One component continuously stores summarised information about evolving streams. While the other classifies incoming data based on the summarised information. An online information network (OLIN) is proposed in [Las02] to classify data streams with concept drift. OLIN uses a tree model built with a fuzzy technique. The model is rebuilt to represent the most recent changes in the data stream. The algorithm adjusts window size based on the error rate. Gaber, Krishnaswamy, and Zaslavsky [GKZ05] developed a LWClass algorithm that applies k nearest neighbours technique for updating the frequency of class occurrences given the data stream features. LWClass introduced a resource aware data analysis technique to adjust the frequency based on the available resources. A rule based classifier for data stream is developed by Ferrer-troyano, Aguilar-Ruiz and Riquelme [FTARR04]. The classifier coined SCALLOP stands for Scalable Classification Algorithm by Learning decisiOn Patterns. Rules are initially constructed from labelled data. New data is either positively covered, negatively covered, or possibly extended. The confidence level for each rule is recalculated based on new data cases. Finally, unlabelled data is classified by a voting based classification technique. In the following, we briefly overview two special categories in data stream mining: time series analysis and sequential data stream mining.

Time series analysis

Time series analysis is a well-studied topic in signal processing and statistics. In data stream mining, time series data is a specialised type of sequenced data stream. Whereas a sequenced data stream consists of a sequence of chunks of ordered data, with or without concrete notions of time. Time series data consists of a sequence of values or patterns obtained over repeated measurements of time [Han05]. Examples of time series data are stock market data, economic data analysis, and observation of natural phenomena (such as temperature, wind, earthquake). Research on mining time series focuses on the representation of time series trends or search for similarities. StatStream [ZS02] has been developed to calculate statistical measures over time series. The developed system applies discrete Fourier Transformation and calculates the error bound correlations and inner products. The Symbolic Aggregation approximation (SAX) technique was proposed to represent time series trends with reduced dimensionality [LKLC03]. The SAX technique applies clustering, classification, indexing, and outlier detection in mining time series data. Similarity matching in time series includes subsequence matching and whole sequence matching. Subsequence matching finds sub-sequences that are similar to a query sequence; whole sequence matching finds a set of sequences that are similar to each other (as a whole) [Han05]. Himberg et al. [HKM$^+$01] built a system onboard mobile phone to analyse time series data. This application used a clustering approach on time series data to adapt a system based on a user's context. A more recent approach tends to detect changes in time series. Online Divisive-Agglomerative Clustering (ODAC) [RGP08] is a clustering technique for time series that continuously maintains a hierarchical cluster structure to monitor the evolution of clusters and detect changes. Xie, Huang, and Willet [XHW13] presented a technique to detect change points in noisy time series data.

Sequential data stream mining

Unlike time series, a sequential data stream is comprised of ordered chunks of data with or without notion of time. Activity recognition data is categorised as sequential data as it contains a set of ordered examples representing ordered activities. The order of activity recognition data is specified by its time stamp. Sequence mining in static data mining has been researched with many approaches such as [AS95, MCP98, Zak01, PHW07]. However, these methods are developed for static datasets which allows several scans for data in order to

extract meaningful information. The luxury of multiple scanning is not applicable in data stream environments. Approaches for mining data streams aim to find frequent patterns or mine sequential patterns. An example of approaches that aim to identify data items occurring frequently in sequential data stream is presented in [MM02]. Manku and Motwani proposed an algorithm for an approximate frequency count that uses an incremental approach to approximate the occurrences of frequent patterns in data streams. Another approach is FP-stream [GHP+03] which also is an incremental approach based on the tree structure that uses titled windows to find frequent patterns in the most recent data in the stream. Teng, Chen, and Yu [TCY03] developed a regression based approach that considers different time granularity for counting frequent patterns. The time granularity approach is also presented in [GHP+03]. Chen and Mei [CM14] presented an algorithm for computing frequency counts over a user specified threshold on a data stream based on the time fading model.

Unlike the aforementioned approach, other techniques have been developed to mine sequential patterns in a steam. SPEED [RPT08] mines frequent patterns and stores them in a tilted-time based structure. Old or less frequent patterns are pruned to maintain only the recent/most frequent patterns. Marascu and Masseglia [MM06] developed a clustering stream technique to build a summary from which sequential patterns can be extracted for web usage. A recent approach proposed in [GQ12] is based on incremental learning on a tree structure that counts minimal occurrences of a sequence in a window. A summary of approaches for sequential data analysis is represented in Table 2.1.

Table 2.1: Approaches for Sequential Data Processing

Domain	Task	Methods
Static data	Mine sequential patterns	[AS95], [Zak01], [PHW07]
Stream data	Find interesting/frequent patterns	[MM02], [GHP+03], [TCY03], [CM14]
Stream data	Mine sequential patterns	[RPT08], [MM06], [GQ12]

State-of-the-art approaches in sequential data stream mining deal with data streams as sets of transactional data with boundaries. Whereas typical data streams in activity recognition consist of sequential chunks that represent activities with no in-between boundaries. Thus, sensory data in activity recognition is a stream of continuous flows of sequential chunks of activities. Stream data for activity recognition is non transactional data with no boundaries. Furthermore, the goal of in activity recognition system is different from sequence stream mining approaches. In activity recognition, the main target is to mine

sequential sequence of activities in order to predict occurrences of activities in incoming data streams.

2.5.3 Mining changes in data streams

This section discusses various methods proposed to capture changes in data streams. We first differentiate between input and target domains in stream mining. The input data is received before time point t_0; while the target data is received between t_0 and t_1. Thus, the change is monitored between the two time points t_0 and t_1. In a streaming dynamic environment, changes are expected to occur between the input and target data. These changes might occur once or many times, gradually or suddenly [GvB+14]. Figure 2.8 represents an example of the change in data distribution between the input domain and target domain in 1-D data.

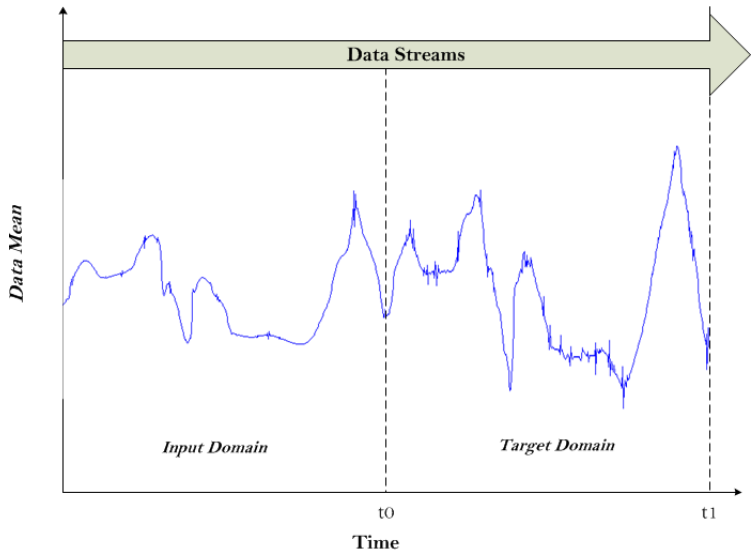


Figure 2.8: Example of Change in Data Streams

Various types of change could be detected in the stream. Concept drift is one of them that refers to the change in the distribution that occurs while the stream evolves. Another type is concept evolution which refers to the appearance of a new concept in the target domain that did not exist in the input domain. An extension of concept evolution is concept forgetting. Opposite to detecting novel concepts, concepts that are no longer relevant in the target domain require a forgetting technique for adapting the model to the recent changes in the stream. Outliers are also considered as a sudden change of the stream. Unlike concept evolution, outliers are not incorporated into the system or added to the model for enhancement. The goal of detecting outliers is to isolate irrelevant data and filter it out from real data. Noise can also be

considered as a special case of outliers. For various kinds of change, adaptive learning techniques are applied in order to update the learner and cope with the evolution of data. Figure 2.9 illustrates categories of changes.

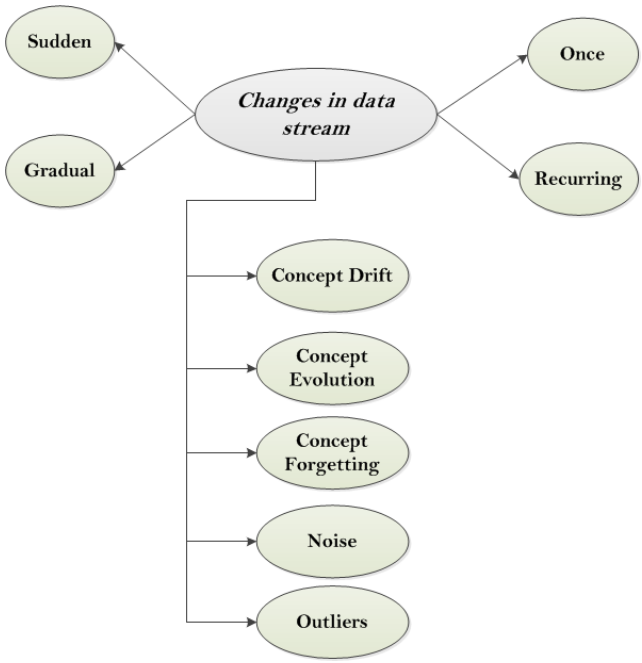


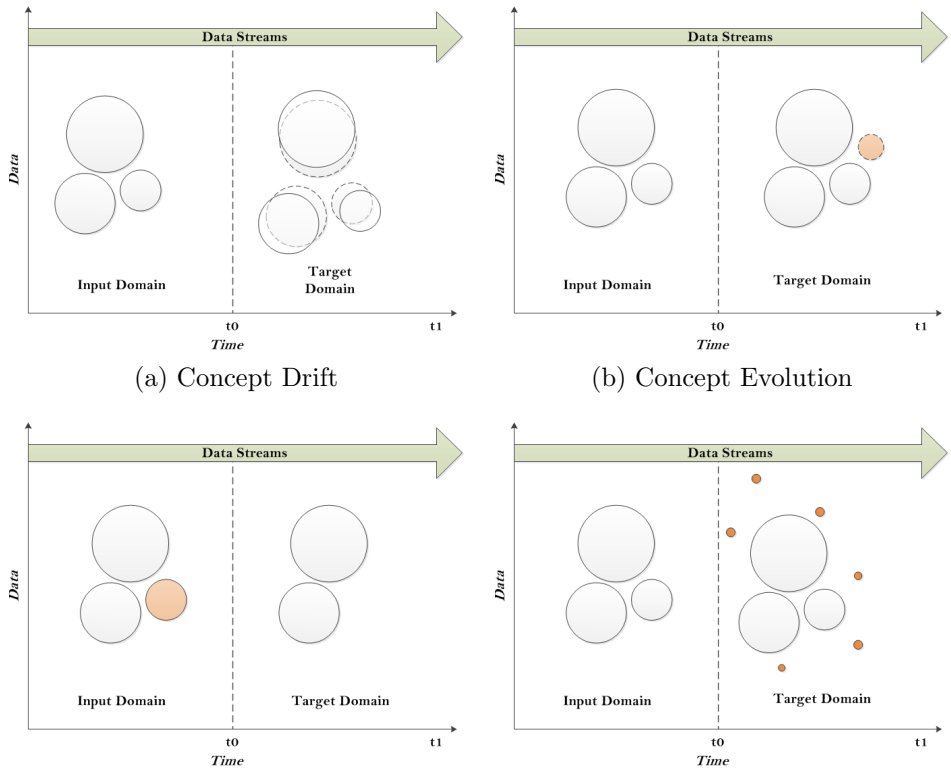
Figure 2.9: Categories of Changes in Data Streams

Adaptive learning in stream mining can be categorised into two approaches. The first approach aims to update underlying concepts to cope with the most recent changes in data streams. In this approach, the change is not explicitly detected, yet the learner is updated periodically to accommodate for the expected changes in the data streams. While the other approach aims to observe data streams in order to detect the changes and then adapt the model upon the identified changes. In the following, we illustrate different types of change in data streams. Then, we discuss adaptive learning techniques that are applied for model update.

Tracking changes

Concept drift. Figure 2.10 illustrates different types of changes in data streams. Figure 2.10(a) represents the type of change known as *concept drift* which is first identified in [SGJ86]. It refers to the gradual change in distribution between input and target domains. Klinkenberg and Renz [KR98] specified three indicators for concept drift. The first one is based on the classifier performance metrics such as the accuracy of the classifier. While the second indicator is based on model properties such as model complexity. The

last one concerns the change of data properties, i.e. data distribution. Examples of techniques that detect concept drift based on performance indicators are the FLORA family of algorithms [WK96]. A FLORA algorithm monitors the accuracy and the coverage of the model of a rule based classifier. The algorithm adapts the window size dynamically according to the measured performance metric. This approach requires true labels provided by the user in order to measure the accuracy. Indeed, this input is impractical in streaming settings when data arrives at high speed and requires real time adaptation. Other relevant techniques in [GMCR04, Bou11] have applied statistical evaluation to monitor the performance and adapt accordingly. Hulten, Spencer, and Domingos [HSD01] presented a system for concept adapting very fast decision trees - CVFDT. The adaptive Hoeffding tree monitors the quality of the previous model and adapts the model in terms of the slitting features in the tree. While Gaber and Yu [GY06] presents a STREAM-DETECT technique that capture change in data streams by monitoring data distribution using an online clustering deviation method.



(a) Concept Drift (b) Concept Evolution

(c) Concept Forgetting (d) Outliers

Figure 2.10: Changes in Data Streams

The change is monitored between two time points t_0 and t_1. Detecting concept drift occurs over either a fixed window size, an adaptive window or combination of both. One major problem with the fixed window is the selection

of the window size. VFDT [DH00] and CVFDT [HSD01], for instance, applied a sliding window of fixed size to handle data streams. A small size of window in stable data causes exhaustion of resources and increases the probability of false alarms. While a long time window in fluctuating data might miss a crucial information and result in slow reaction to the encountered changes. Unlike the fixed window approach, an adaptive window is adjusted according to the detected changes. Thus, window size either expands or shrinks upon the change indicator. Widmer and Kubat [WK96], for instance, applied a sliding window of adaptive size to monitor the incoming data. The assumption in this approach is that the most recent data is the most important data. Therefore, the model is updated based on the most recent data appearing in the most recent window. Klinkenberg and Joachims [KJ00] proposed an SVM model that compares the error on various window sizes, then selects the size with minimum error. The third approach is based on the combination of two different window sizes; one is fixed and the other is adaptive. The fixed window stores the baseline historical information, while the sliding window captures the incoming data streams. Where statistical measures change between the two windows, the change is detected [DR09, AB13]. Bifet and Gavalda [BG06, BG07] proposed an adaptive sliding window technique - ADWIN and ADWIN2 - that maintains the size of windows according to the rate of change. The main drawback of applying the combination of both windows is the resource constrains. Figure 2.11 presents the taxonomy of concept drift approaches.

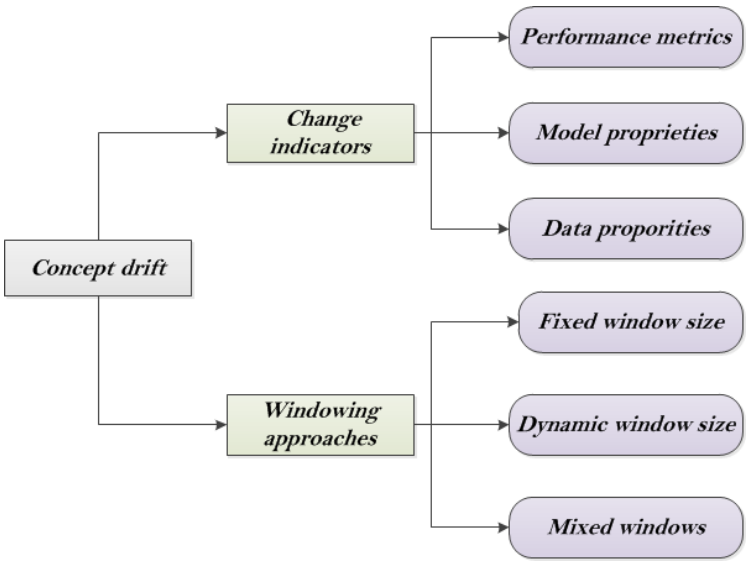
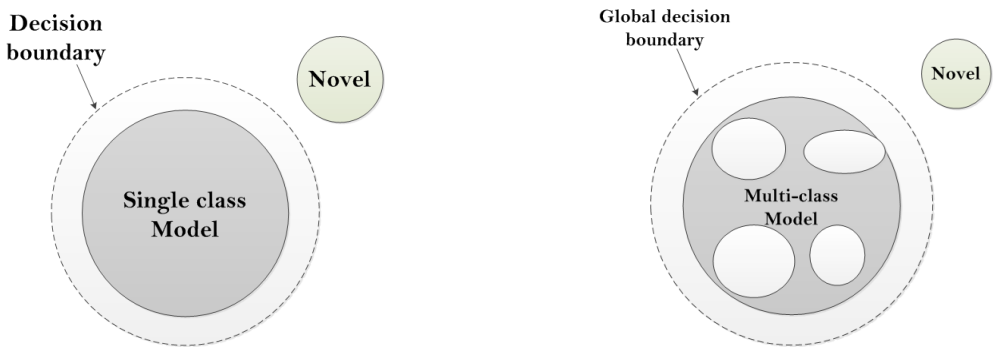


Figure 2.11: Summary of Concept Drift Approaches

Concept evolution. The other type of change is *concept evolution*, which refers to the appearance of novel concept in the stream. Figure 2.10(b) represents an illustration of concept evolution. Detecting a novel concept is a

challenging task in data streams, especially when dealing with data streams with concept drift. An efficient approach has to be able to distinguish between the drifting of an existing concept and the appearance of an entirely new concept. The appearance of a new concept is followed later in the stream by detection of the recurring data instances that belong to the novel concept. Concept evolution techniques aim to capture the arrival of novel concepts and incorporate the detected novel concept into the existing underlying concept. This integration allows further detection of recurring instances that belong to the novel concept. Different approaches have been developed to distinguish between existing and novel concepts. In terms of the underlying concept, some approaches identified the underlying existing concept as a single model that represents only a single class, while others considered a multi-class underlying concept. Spinosa, Carvalho, and Gama [SCG07] represented the underlying model by a single concept with all incoming data either part of the underlying model or novel concept, as explained in Figure 2.12(a). This approach assumes that there is only one "normal" class and any other classes are novel. The cluster based system, named OLINDDA, is based on three models: normal profile model, concepts that extend the normal profile, and novel concepts. Learning phases in OLINDDA are offline and online. The normal model is built in the offline phase, while the extension in the online phase detects minor changes in the normal model. A novel concept is detected when incoming data is located away from the normal model and also satisfies a specific validation criterion. A single model approach limits the capabilities of a concept evolution algorithm to differentiate only between existing and novel concepts without considering the presence of multiple existing concepts. Thus, other approaches have developed to address the multi-class structure of an underlying concept, illustrated in Figure 2.12(b). The techniques developed in [MGK+11, FGC13, HH10] are examples of stream learning approaches for novelty detection with multi-class underlying concept. ECSMiner [MGK+11] applies an ensemble of classifiers to an incoming stream of equally sized data chunks for prediction. The global decision boundary is defined as the union of local decision boundaries for existing classes in the underlying model. The classifier model in the ensemble classifier is dynamically updated to detect instances that are outside the global boundary. If no classifier is able to predict the incoming data, then data is stored in short memory for further processing. The novel concepts are detected when data in a buffer maintains cohesion with other buffer data and separation from existing underlying concepts. This approach addresses the novelty detection in multi-class underlying concepts, yet it requires all data chunks to be labelled to define the new concept. Faria, Gama, and Carvalho [FGC13] proposed the

MINAS system for concept evolution which applies unsupervised learning approaches. MINAS classifies new incoming instances as known or unknown. Unknown instances are the ones located outside the decision boundaries. The declared unknown instances are stored in short time memory. Then, data in short memory is clustered in order to discover new concepts. Hayat and Hashemi [HH10] proposed an approach that is based on the discrete cosine transform to build normal concepts of multi-classes with sub-clusters. The distance measure is also applied to distinguish between existing and novel concepts. The aforementioned approaches rely mainly on the distance measures that predict novel concept based its location from the decision boundaries. The new concept is declared as novel if it is outside the global decision boundary which is the union of local boundaries of clusters. Although the underlying concept contains multi classes, creating a global decision boundary results in a similarity between multi-class models and single class models. It combines all existing concepts in one concept and defines the global boundary for the combined concepts. Thus, these approaches also did not address the appearance of novel concepts that might exist outside the local boundaries, yet inside the global boundary. Moreover, most of theses approaches did not consider the labelling cost of data streams when identifying novel concepts.



(a) Single Class Model (b) Multi-class Model

Figure 2.12: Underlying Concept Approaches for Concept Evolution

Concept evolution approaches are also categorised based on the action taken upon novelty detection. Yeung and Chow [YC02] and Yang et al. [YZCJ02] presented instance based algorithms that consider whether the incoming data is sufficiently close or far from the underlying concepts based on some appropriate metric. Novelty detection in these techniques detect "filtered out" instances without tracking data for the detection of normal concepts. While approaches in [FGC13, HH10, MGK+11, SCG07] extend the detection of "filtered out" instances by detecting the level of cohesion among these instances to form a novel concept. Studies in [MAKK+11, AKMK+12] further

integrated the novel concept with underlying ones in order to detect recurring instances that belong to novel concept. Figure 2.13 summarises various approaches in concept evolution. The idea of concept evolution is also attached to concept forgetting. The system performance relies on its capabilities to learn changes and new concepts appear in the stream as well as forgetting outdated concepts that became a burden on the system [KBDG04]. Whereas incremental learning manages the continuous learning of new concepts, decremental learning focuses on ***forgetting abandoned concepts*** [CP01]. Figure 2.10(c) explains the concept forgetting in data streams.

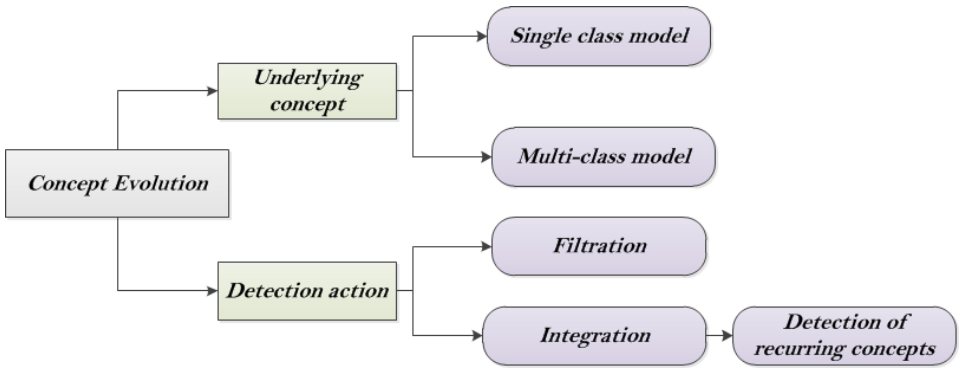


Figure 2.13: Summary of Concept Evolution Approaches

Outliers. The other category of change that is explained in Figure 2.10(d) is *outliers/anomalies*. Many of the outliers are considered as noise, while others are of paramount importance such as credit card frauds. Concept evolution in data streams is closely related to the outlier detection. Outliers are defined as data instances which deviate from underlying concepts. However, this definition also applies for novel concepts. The main characteristic that differentiates between outliers and concept evolution is the cohesiveness among "filtered out" instances. The novel concept has to satisfy a validation criterion that concerns mainly the separation of short memory instances from underlying concepts as well as cohesion among these instances to form a novel concept. In streaming settings, outlier detection under realistic assumptions is an unsupervised learning approach because they are mostly rare to occur. Therefore it is not possible to train the model on them beforehand. Different metrics have been developed in literature to measure the deviation of incoming data from underlying concept that define anomalies. The deviation measure such as distance and distribution have been applied pervasively for outlier detection. There are many techniques that studied outlier detections in data streams such as [AF07, AKBS12, NKR10, PLL07]. Few studies have combined the detection of outliers with concept evolution such as in [MAKK+11, MCK+13, AKMK+12]. A detailed survey of outlier detection approaches is presented in [Agg13]

In a nutshell, evolving data streams may have encountered many kinds of change. The change could be gradual such as some kinds of concept drift or concept evolution. Sudden change is also expected such as anomalies and outliers. A particular change might occur once or recur along the stream: i.e, detecting recurring novel concepts. In this section, we discussed various kinds of change and different approaches to detect each. The main target of these approaches is to spot and identify the change. However, in the following section, we review techniques to learn from rapidly changing data streams. Thus, we study the problem of learning in an evolving streaming environment when changes occur.

Learning from changes

Adaptive learning. The evolving nature of data streams emerges the need of a learning approach that is capable of accommodating the anticipated changes. Continuous learning in evolving data streams refers to the well-known term of *incremental learning* or *adaptive learning*. Three are two categories of adaptive learning; blind/implicit learning or informed/explicit learning [GRSdC10]. Techniques for blind learning update the learner periodically without prior knowledge of the encountered changes. The adaptation occurs at fixed time intervals independent of the kind of change. Unlike the blind approach, explicit/informed learning techniques are triggered when change is detected. Therefore, informative learning based on the kind of change is performed whenever change occurs.

An adaptation process in the blind approach is incremental without prior explicit knowledge of the change itself. VFDT [DH00] is a typical example of blind learning where the decision tree leaves are updated periodically according to a loss function. In contrast, informed learning requires explicit knowledge about change to trigger the adaptation of the learner. Masud et al. [MGK+11], for instance, applied an ensemble classifier for detecting concept evolution in data streams with concept drift. The proposed system incorporates a new learnt model into a learner for future prediction of recurring instances.

Blind learning has the advantage of the periodic update without relying on the detection of change and its corresponding performance. However, there are also some limitations with this kind of implicit learning. The adaptation may take different forms based on the kind of change. For example, the action required for tuning the learner in case of concept drift is different from the action required when a new concept has emerged. Moreover, performing updates based on fixed time intervals strongly relies on both the interval length and the rate of change. The response to change might be slow if the interval size is

big especially in data streams with high rates of change. On the other hand, unnecessary updates might cause over use of resources in short time intervals with stable data streams. Thus, the trade-off between the cost of updates and the gain in performance has to considered for choosing an adequate interval length [Gam12]. On the other hand, informed/explicit learning only responds to the detected changes. Based on the change detected, different actions are taken to respond to the specified changes. The main limitation of this approach is its dependency on the change detection. Learning only occurs when change is detected and is tightly related to the detection performance. Either an incapability of detecting changes or a high rate of false alarm in the detection technique might mislead the learning process. For example, whenever the change detection fails to detect the appearance of new concept, the new concept and its recurring instances will be either misclassified or classified as *unknown*.

Adaptive learning is also categorised based on the adaptation response. The adaptation reaction is either revolutionary or progressive. In the revolutionary approach, the learner is retrained and reconstructed with the most recent/most important data. The existing model is discarded and totally replaced with a new model from new data [GvB+14, SK01, GMCR04]. Alternatively, new data adapts existing models incrementally in the progressive approach in order to react to the changes in the stream. An example is the OLINDDA system [SCG07] that applies online and incremental learning for extensions of existing concepts or discovered novel concepts.

Adaptive learning is also influenced by the way of managing new data applied for learning. That includes learning from data either in batches or responding to each single instance. Batch incremental learning stores data in buffer, then all data is used for incremental learning in batch when it reaches a pre-specified threshold, i.e., buffer size, change detected. Guralnik and Srivastava [GS99], for instance, have developed an approach for batch incremental learning in time series data streams. Figure 2.14 summarises different approaches for adaptive learning.

A special category of incremental learning is active learning. Active learning focuses on labelling only few data instances order to enhance the learner accuracy. In data stream settings, it is impractical to assume the availability of labels while a stream evolves. Many approaches have been proposed for efficient active learning techniques in data streams, such as in [WY05, MWG+12, HD07]. Studies in [HD07, FHWP04, LMND13] developed an active learning approach based on the change detection method. When the system detects

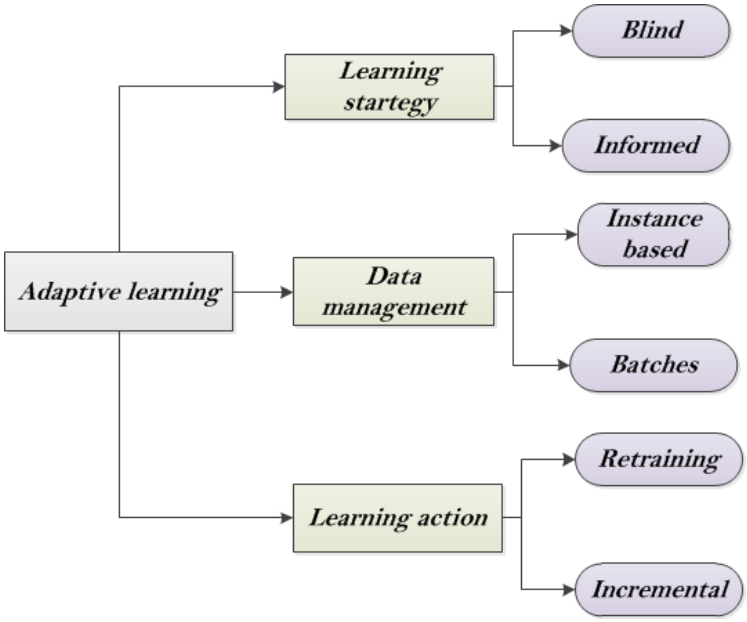


Figure 2.14: Summary of Adaptive Learning Approaches

change in a data streams, that triggers active learning to inquire for true labels. Other approaches built on an assumption that part of the data streams is labelled while the other part is unlabelled [LWH12]. Recent work in [ZBPH14] presents a framework for active learning in data streams with concept drift. The developed system learns in batches while saving the labelling cost. The main target of active learning is to keep the trade-off between minimising the cost of labelling while maximising the gain in performance.

All in all, we reviewed state-of-the-art-techniques in data stream mining. Approaches for data stream mining focuses mainly on the non-identical distribution feature of data. Techniques for data stream mining include handling data streams, learning from data streams and capturing change in evolving data streams. Incoming data in activity recognition typically streams from sensors. In the following section, we discuss the intersection between stream mining and activity recognition areas of research. That also includes the mapping of terminologies between the two fields of study.

2.6 Activity Recognition Techniques using Data Stream Mining

This section focuses on the intersection of the two fields of research: stream mining and activity recognition. State of the art activity recognition techniques address the dependency nature of activity data, while approaches in

stream mining focus on the non-identically distributed streaming data. Traditional activity recognition approaches assume the stationary of data. This assumption is violated when dealing with real time sensory data that is typically evolving over time. Data streams approaches deal with non-identically distributed data and also constraints imposed in the streaming nature, e.g. infinite number, high speed, concept drift etc. However, most of the stream mining approaches have not applied for activity recognition. Activity recognition in data streams approach is an overlapping between stream mining and activity recognition when both *i.i.d.* conditions of independent and identically distributed are violated. The developed techniques for activity recognition in data streams have to be adaptable and flexible, to accommodate for the evolving nature of activities.

Some approaches in data stream mining might be seen as applicable for activity recognition. Particularly, mining sequential data streams could be considered for activity recognition. This approach deals with sequential data streams as ordered chunks with or without a notion of time. Although techniques in this approach consider the dependency in data, two major core differences make these techniques incompatible with activity recognition. According to the previously explained approaches in Section 2.5 and Table 2.1, most techniques in sequential data processing aim to find interesting/frequent patterns instead of mine the sequential patterns in order to predict classes of incoming data. This is different from the main target of activity recognition which is the prediction of incoming activities with sub-targets that might also be included in some cases for finding the interesting patterns. In data stream mining, few studies have addressed the actual mining of sequential patterns in data streams. Yet, these techniques assume that the sequential data is in a transactional form that contains boundaries separating different transactions. This assumption is not valid for activity recognition when a stream of data representing a sequence of performed activities arrives with no boundaries in-between. Although sequential stream mining is conceptually related to activity recognition, its techniques are not applicable for activity recognition because of the aforementioned core differences in terms of the learning target and domain definition.

Real time activity recognition might also be related to activity recognition in data streams. Techniques for real time recognition focus mainly on enhancing the response time by simplifying both data processing and model structure. In an early study, Karantonis et al. [KNM+06b] implemented a real time classification to distinguish between activity and rest, before classifying further lower level activities. The classification technique is based on a hierarchical binary

structure for broad classification at the top level. Sub-classification occurs at the lower levels. The resource aware techniques are performed onboard the sensor and target real time recognition. CeneMe [MLF+08] represents a system for real time recognition of contextual activity using off-the-shelf, sensor-enabled mobile phones. The algorithms used by the CenceMe classifiers run on the phone and the backend server according to the split-level classification design. The backend server classification presents a higher level of contextual recognition of activities. The activity-recognition algorithm presented in [PCE10] is based on a binary decision tree classifier to automatically recognise physical activities on a portable device. The labelled data is provided to evaluate the system performance and update the decision tree threshold values with the user's own data.

Although the response time of these techniques seems to resemble the one for stream mining techniques, none of the other characteristics of data streams have been addressed in these techniques. Particularly, they have no notion for handling the infinite, high speed, and mostly unlabelled data streams. Moreover, these approaches lack the consideration of concept drift, outliers, or concept evolution that are known phenomena in data streams. These approaches are not flexible for personalisation and adaptation with the evolving activities.

Some terms in activity recognition have their corresponding meanings in stream mining, yet in different settings and with different contexts. A typical example of this is *outlier detection*. Hawkins [Haw80] defined an outlier in general as: "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". Outliers in data stream mining correspond to abrupt changes in data streams from the underlying concepts. Outliers can be referred to as noise, anomalies, and abnormalities. The term of outlier detection in activity recognition refers to *sudden activity* such as *fall detection*. Detection of sudden activities resemble outlier detection as they both aim to find the unusual events in data. The term of "sudden activity" or "fall detection" is pervasively discussed in activity recognition especially for elderly people aids [LK09, MSS13].

Novelties are considered as a special category of outliers, yet with aggregated data points [Agg13]. However, in this dissection we suggest a separation between the two concepts, outliers and novelties. By definition, outlier is an unusual event that is completely different from the underlying/known concepts. It is mostly expressed as a set of individual data points that correspond to noise, sensor failure, etc. This data cannot be aggregated under a single concept as they have different natures and different causes. On the other hand,

novelty data represent novel concept which still have some commonality with existing underlying concepts. Though the novel concept is different from the underlying concept, they still preserve some common similarities. Examples of outliers are noise, credit card fraud, or sudden falls in activity recognition which are entirely different from the underlying concepts. While novelties are about discovering new "normal" concepts such as discovering a new category of credit card transaction or a new activity that the user started performing recently. There are two main differences between outliers and novelties. First, novelties represent novel concepts that have not been seen by the system before. Yet, novel concepts are neither completely different nor abnormal from the underlying concepts. Thus, new concepts might appear in the middle of existing ones. Moreover, novel concepts are represented by a set of aggregated instances, while outliers instances are mostly irregular that appear separately and most of the time with no relation to other detected outliers.

The term "novelties" is also referred to as concept evolution and novelty detection. In stream mining approaches, *concept evolution* is the process of monitoring the data streams in order to discover the appearance of novel concepts. The term of concept evolution is analogous to *detecting novel activity* in activity recognition. In contrast to concept evolution, some concepts became outdated and no longer relevant to the target domain. These concepts require an adaptive mechanism to forget the abandoned concepts. The term of *concept forgetting* has been presented for data streams in [GRSdC10]. Concept forgetting is also relevant to activity recognition whereas activities are no longer performed by users. *Forgetting abandoned activities* aims to update the model continuously to reflect the most recent changes in data and remove outdated/abandoned activities.

Another term that is well-known in stream mining is *concept drift*. It primarily refers to the change in data distribution while a stream evolves. This change could be gradual or sudden that happens once or recurring. A typical data stream evolves over time. Thus it requires an effective approach to handle the drift and accommodate for the most recent changes in the stream. In activity recognition, the definition of concept drift is relevant, yet in a different context. The deviation between input and target domains in an activity stream occurs as activities are performed in a different way from one user to another. When the recognition model (target) is different from the incoming data (input), this deviation resembles concept drift for activity recognition. More precisely, approaches addressing concept drift in activity recognition are termed *model personalisation*. The personalisation process targets the adaptation of the recognition model to fit data for a particular user.

The process of learning from aforementioned changes is done with *adaptive learning* in data streams which include incremental and active learning. The anatomy of adaptive learning in data streams is *model adaptation* in activity recognition. The goal of adaptive learning in data streams is the same of model adaptation in activity recognition, which is continuous learning to adapt to the most recent changes in the evolving data. The concept of model adaptation in activity recognition applies also to incremental and active learning.

Some of the issues that have been addressed in both stream mining and activity recognition, yet under different settings and with resembling meanings, are summarised in Table 2.2. The table shows a subset of stream mining terminologies and its corresponding related terms in activity recognition.

Table 2.2: Terminology Mapping between Stream Mining and Activity Recognition

Stream mining	**Meaning**	**Activity recognition**	**Meaning**
Outlier detection	*The detection of bursts in data streams*	Sudden activity detection	*The detection of sudden changes in activity data*
Learning from concept drift	*The detection and response of a gradual change in a data stream*	Model personalisation	*The tuning of the model to suit a personal way of performing activities*
Concept evolution	*The discovery of new concepts*	Detecting novel activity	*The discovery of new activities*
Concept forgetting	*The decremental learning of outdated concepts*	Forgetting abandoned activity	*The decremental learning of abandoned activities*
Adaptive learning	*The continuous and incremental learning with evolving data*	Model adaptation	*The continuous adaptation of the recognition model*

Few studies have considered activity recognition in streaming environments. Krishnan and Cook [KC14] developed an efficient technique for handling streaming data based on windowing technique. This system is based on the fact that different activities can be characterised by different window lengths. Sensors deployed for this study are binary motion sensors installed

in a smart home environment. Another study that also deploys binary sensors in a smart home environment is presented in [RC10]. In this study, authors applied a tilted time window to find sequential patterns from streaming data using multiple time granularity. The technique adapts window size, not the classifier model, for boosting the recognition accuracy. Unlike the aforementioned techniques, Gomes et al. [GKG+12a, GKG+12b] have developed an on-board data stream mining technique for mobile activity recognition. The developed system predicts activities in the stream and adapts the model to fit a user's profile. Do et al. [DLL13] built a logic based framework for recognising basic and complex activities from mobile sensors. Recently, Lockhart and Weiss [LW14] have presented the Actitracker system for mobile activity recognition. Actitracker builds a general/universal classifier that could be replaced by a personalised model for a particular user. The system collects data with fixed time windows and transmits data for backend server for processing. More details of these techniques and other state-of-the-art activity recognition techniques along with the research gaps are discussed in Sections 2.7 and 2.8.

2.7 Key Techniques

In this section, we review key techniques in activity recognition. The first two systems are deployed in smart home settings, while the others are on the mobile phone. Krishnan and Cook [KC14] proposed a sliding window based approach for recognising streaming activities from motion sensors events in a smart home environment. The learning classifier applies Support Vector Machine (SVM) for modelling activities. The main focus of this study is on the techniques for handling data streams. The study evaluated different windowing techniques for analysing a stream of activities. The performance is evaluated on a fixed size window with both time based weighting and mutual information weighting. It also includes the classification probabilities of activities that are previously recognised in the preceding window. The developed technique handles the 'other' class activity that does not correspond to any known activity classes. It explicitly trains the model offline for transitional activities and other unknown activities. Then, it gathers them all in one 'other' activity class and incorporates it with the model. While this study has been added to the field of activity recognition, research is still requires to address other research gaps.

The approach addressed the important research of activity recognition in streaming settings. However, the developed approach did not allow personalisation or adaptation with the evolving streams. Thus, the classifier is built

with training data, with no flexibility to be adapted and personalised post the deployment. The recognition of 'other activity' is not for detecting the emergence of a novel activity in the stream. The 'other activity' category adds more activities corresponding to transitional and unknown activities to the offline model during the training. Moreover, incorporating the 'other' activity into the model causes more confusion in recognising both known and unknown activities. Also, the system takes an average of 4 days to learn the different activity models for each technique. Moreover, a smart home environment requires installation of sensors in fixed laboratory settings which limits the application of activity recognition.

Another activity recognition system in smart home settings is presented in [RC10]. Rashidi and Cook developed a system that recognises activities from unbounded input data: a stream rather than a transactional format. The aim of this study is to find the sequential patterns in the stream of data of a smart home environment. Sensors deployed for this study are also binary motion sensors. The proposed system applies a tilted window technique for handling data collection. The tilted window with an approximation approach applies a relaxed threshold in order to find the sequential patterns in the stream. The tilted window stores the records in order of time; the oldest records are kept at the highest granularity. For mining patterns in the stream, an extension of DVSM [RCHSE11] method has been proposed. After mining the sequenced patterns, the system updates the tilted time window with the most recent patterns. The system combines sequential processing and data stream mining for activity recognition. Nevertheless, the proposed technique adapts only the tilted window rather than updating the recognition model. The main focus of this system is handling the stream with the adequate tilted window for recognising interesting patterns. Explicit recognition of sequenced activities in a data stream is not considered in this study. Furthermore, no techniques have adapted for dealing with concept drift, concept evolution, and concept forgetting in data streams. Stream mining from smart home binary sensors with less constrainted devices is less challenging than mobile devices with limited resources that require online learning and recognition in real time.

MARS [GKG+12a, GKG+12b] stands for Mobile Activity Recognition System. It is an incremental system for predicting activities in data streams on the mobile device. The learning process in MARS is divided into two phases: training and recognition. In the training phase, a user performs activities and annotates them interactively while data is collected from mobile sensors. The collected annotated data is saved for building the model offline. In the recognition phase, the incoming unlabelled data is then classified based on the offline

built model. The study compared the results of both static decision tree and incremental Naive Bayes for evaluating system performance. The proposed algorithms are light-weight thus can be deployed on mobile devices. Although, MARS has presented an early system that combines activity recognition with stream mining, some challenges still need to be addressed. The system assumes the availability of labelled data for each user. Each individual has to collect and annotate data representing the personalised activities for building the model. When new subject uses the system, the model has to be retrained with the data collected and annotated for this particular user. This assumption is impractical in streaming settings where the majority of data is unlabelled. Moreover, retraining the entire model for recognising the user specific activities is a time consuming process that may not be applicable in streaming settings. The developed system also lacks the description of techniques for handling the streaming nature of data. Online and incremental adaptation of the model to include entirely new activities or forget abandoned activities has not been considered in this study.

The system presented in [DLL13] applies logic based stream reasoning combined with an Artificial Neural Network (ANN) for recognising both complex and basic activities. The system comprises four components: Client, WebServer, data stream manager system (DSMS), and reasoning server. The client collects data from mobile sensors and trains the Artificial Neural Network for recognising basic activities. The collected data from a user's mobile phone (client) is transmitted to the WebServer which uses the GPS data to recognise the user's location. The WebServer sends all data to the server. It also facilitates the connection between the client and backend server. DSMS collects and stores data for the reasoning server component. The recognition of complex activities and ambiguity reasoning occur at the reasoning server component. The system tracks complex activities for suggesting a healthier user lifestyle. Despite the efficiency of the developed system, personalisation and adaptation are not considered in this study. Also, the reasoning on a backend server requires additional connectivity requirements and does not preserve a user's data privacy.

Actitracker [LW14] is a recent system that is being developed for activity recognition application in the health domain. A universal model is built initially from general and impersonal data. Then, the user has the option to either train the model with the personalised data or instead deploy the universal model. For model personalisation, the system retrains the model with a sufficient amount of personalised annotated data collected for a particular user. Streaming data is collected from mobile sensors with a fixed size window of 10

seconds. The collected data is sent to a backend server for processing and classification. Various classification algorithms are applied: decision trees (C4.5 and Random Forest), instance-based learning (knn), neural networks, Naive Bayes and Logistic Regression (LR). The Actitracker system only uses Random Forest models (which were shown to perform very well) [WLP+14, act]. Data is classified with a personalised random forest if available; with a universal random forest model otherwise. The developed system is available on the application store and ready to download for mobile devices. The application has already 250 registered users. The data is shared with activity recognition researchers via publicly available datasets [LW14]. This system used a fixed window size technique to handle data streams. However, the personalised model is neither automated nor incremental. A collection of personalised activities requires not only annotation for each activity but also retraining the entire model to suit the personalised user. Training a model in such way is impractical in streaming settings that require automated and incremental approaches for 'adjusting' the model to fit a particular user. Moreover, the annotation process is time consuming, erroneous, and not applicable for a streaming environment. Therefore, selecting only the most profitable data with the minimum effect on performance is crucial for effective recognition. The system also lacks the consideration of discovering new activities or deleting irrelevant activities.

Zhao et al. [ZCL+11] developed a cross-people motion activity recognition system. This developed smart phone based activity reporting system can accurately recognise the daily activities of stationary, walking, running, upstairs, and downstairs, and report the accumulated time of each activity. More significantly, the system is concerned about the calibration free and personalised problem. The algorithm learns a binary decision tree model for one person from his labelled samples, transfers its structure to another person and automatically adapts its non-determinate nodes with the unlabelled samples of the new person, thus accomplishing the cross-people knowledge transfer task. The system consists of two components, the first one is the TransDT layer (Transfer Decision Tree), and the other one is the EM Algorithm layer (Expectation Maximization Algorithm). TransDT is a binary decision tree learnt off-line and built on a well-labelled training set. In this layer, the system constructs a classifier as well as finds the attributes that can distinguish one class from others. The EM layer corresponds to transferring and adapting the TransDT model to a personalised one. When having collected sufficient unlabelled samples from a new user, the algorithm classifies them with the TransDT model and then uses the result as the initial condition of the EM algorithm. After the EM algorithm, the unlabelled samples are well labelled. So the system

can update the non-terminal nodes of the TransDT model and thus form the personalised model.

Pärkkä et al. [PCE10] developed an activity recognition system based on a decision tree classifier that automatically recognises physical activities on a portable device, online. The model is personalised by updating the decision tree threshold values with a user's own data. The central device in data collection and activity recognition is a personal digital assistant (PDA). The application receives data over a Bluetooth connection, computes feature signals from raw data online, classifies the data in a second-by-second basis online, and stores the data on a memory card. The human movements are quantified with Nokia wireless motion bands using the 3-D accelerometer signal and Bluetooth connection for data transfer to the PDA. Only ankle sensor data are used for computing the feature signals. The time-domain features were computed from the most recent samples, and the frequency-domain features were computed from the same samples and one added zero for an efficient fast Fourier transform (FFT) implementation. The structure of the binary decision tree includes four nodes with one threshold value in each node. The personalised algorithm searches for an optimum decision boundary between the activities falling left and right in each node. It takes 3-10 minutes of new data with annotation and uses that for updating the thresholds in each node. The algorithm requires only a few comparisons and thus consumes very little battery power. Personalising the training model by adjusting the thresholds is a key contribution, however a long delay up to 10 seconds is required for updating the model.

The last two techniques present key systems applied for personalised activity recognition approaches in static environments. Although having an improved accuracy with personalised models, these techniques are incapable of handling the evolving nature of the incoming data streams. Moreover, the prediction relies on the static model built offline. This model can not be pruned or expanded after deployment.

2.8 Research Gaps

Upon discussing the wide range of approaches proposed for activity recognition, we present the research gaps that have not been addressed or partially addressed in the literature. One of the main limitations that encounters the recognition system is the *static nature of the classificatory model*. Studies demonstrated that deployment of a personalised model for activity recognition outperforms the system of applying a general model [WL12]. It can easily

be seen that models that are built for general activity recognition would need to be tuned and adapted to suit the evolving activities data. Each individual has his/her own personalised way of performing activities. Thus, the significance of personalising the training model in real time became crucial in activity recognition in order to improve the recognition accuracy for a specific user. Personalisation is the process of tuning the general model to represent a user's particular way of performing various activities. Personalisation updates the current existing model without changing the core activity types. Therefore, the personalisation process only 'tunes' current activities for a particular user in an incremental manner. Few studies in activity recognition considered the personalisation issue. Most of these studies addressed the personalisation by retraining of the entire model with a particular user's annotated data. Few studies considered the incremental and automated approach for personalisation. Moreover, a data stream is an unbounded flow of data that is mostly unlabelled. Personalisation methods in activity recognition require annotation of recent data to update the model. However, *the scarcity of labelled data* as well as the *time consuming process of labelling each data instance* are in conflict with the streaming nature of activity recognition of sensory data. Alternatively, the recognition of activities needs to consider batch labelling for only the most profitable data a in batch active learning approach. Thus, the time and resource consumptions are only limited for the most profitable data. Handling the streaming nature of data is crucial for processing and responding to a streaming activity in almost real time.

Moreover, in a real life application, activities that a user performs evolve over time. Therefore, the set of activities represented in the model has to be updated to reflect the change in the performed activities. That includes adding new recognised activities and also understanding activities that are no longer relevant to that particular user. Model adaptation is a key criterion of the robustness of any activity recognition system. There is no notion of adaptation/refinement of the classifier models in the literature. Models do not include activities that may emerge over a period of time (post the data collection) or changes in a user's patterns, which are both completely realistic in the context of a mobile user. The adaptation process needs to update the recognition model over time to reflect changes in a real life user's activities in real time.

The deployment of inexpensive sensors for collecting data for activity recognition coupled with stream mining techniques has resulted in the development of the wide variety of applications. Sensors could be either onboard the mobile device itself, body worn or embedded in a smart environment. Using mobile

devices with limited resources for recognition of streaming activities became a hot topic of research in pervasive and ubiquitous computing. Two approaches have been applied for mobile based recognition systems. Firstly, the phone transmits the data to the backend sever with the server applying the activity recognition model and transmitting the results back to the phone. The second method involves implementing the activity recognition model directly on the mobile device. Given the computational power of these everyday devices, this is certainly a feasible option. One key advantage of this method is that it removes the need for a server and therefore save transmission time and allows real time prediction. Today's mobile devices make the solution perfectly scalable, and ensures the user's privacy, since the sensory data is kept locally on the device.

Time and accuracy are well-known critical factors used in judging the performance of any real-time activity recognition technique. Building a classifier that accurately recognises physical activities on a mobile device in real time is a key research issue.

2.8.1 Comparison of the state-of-the-art activity recognition techniques

In Table 2.3, we present a comparison of the different activity recognition systems addressing the aforementioned research issues. It is clearly demonstrated that there are many significant missing research gaps that need to be addressed for the development of an efficient activity recognition system. Although the pervasive techniques in activity recognition are deployed in a static environment, we focus in this comparison mostly on the missing gap in recent approaches for streaming environment. We also included both techniques in [ZCL+11] and [PCE10] for their highlighted contribution in model personalisation.

Data collection platforms vary from one system to another. Data is collected from either mobile, body worn, or smart environment sensors. The processing platform is related to the recognition application and response time. Processing data on a backend server or on a PC indicates two factors. The model is not light-weight. Therefore it has to be implemented on a high performing device, such as a PC. The other factor is the real time recognition as the implementation on a backend server or PC causes a delay in transmitting data back to the user and thus does not maintain a real time recognition.

None of these systems considered real time adaptation for activity recognition. Recognition systems use a static training model which is built offline

to recognise incoming data. There is no notion yet to expand the model after it is already deployed. When new activity was urged or any current activity abandoned, current techniques could not be adapted accordingly. Reflecting the change in activities in real time is also a crucial research issue. Only an approach in [KC14] considered the discovery of 'other' activity. The recognition of other activity is different from recognising novel activities from an evolving stream. The technique has to build a static model offline that represents other transitional and unknown activities. It explicitly trains the model offline for transitional activities and other unknown activities. Then, it gathers them all in other activity class and incorporates it with the model. Therefore, this approach is not detecting the emerging of novel activities.

Two approaches for model personalisation are presented in Table 2.3. The first is a complete reconstruction of the learning model with the personalised data, while the other aims only on tuning the model to suit a particular user. There is a little focus on the personalisation of the learning models in activity recognition. Retraining the model is not applicable for streaming environments. In the other incremental learning approach, offline models in [PCE10] and [ZCL+11] are refined by tuning the model with incoming data. However, they both deployed in non-streaming environments. Moreover, the Pärkkä et al. technique performs the personalisation offline with long delays. Therefore, none of these systems is representing a real time approach to resolve the personalisation issue in a streaming environment. The scarcity of labelled data requires incorporating active learning with recognition for labelling only a small amount of data that is most informative. Relying on less labelled data has not been addressed by any of these key systems.

We focus in this dissertation on filling in the missing gaps for building an efficient activity recognition technique as per objectives discussed in Chapter 1. We propose, develop and evaluate techniques that address the personalisation (Objective 2) and adaptation (Objective 3) of the recognition model in a streaming environment (Objective 4). The new techniques are flexible (Objective 1), to learn incrementally and contentiously from the evolving activities in data streams and adapt the recognition model accordingly. The core of our focus is to target the challenge of the scarcity of labelled data by incorporating an active learning approach (objective 5) with all our techniques. The merit behind active learning is to label only a small amount of data that is most informative and with lowest cost.

We present our contribution in this dissertation in the next three chapters. Chapter 3 represents the *modelling component* that builds the baseline framework which is essentially flexible, robust, and computationally efficient. The

baseline framework is the basis of our developed techniques for recognition. We also represent in this chapter the *recognition component* that uses a developed ensemble classifier based on the hybrid similarity measure approach for recognition. We move to a streaming environment in Chapter 4 by developing our technique for detecting personalised activities from data streams. This technique adds another component of *personalisation* that applies incremental and active learning for continuous personalisation of the recognition model with the evolving streams. Finally, we target the recognition of an entirely novel activity with continuous learning from evolving data streams in Chapter 5. The approach aims not only to discover the arrival of the novel activities but also incorporate the novel activities into the recognition model for future recognition of the recurring activities. The technique aims also to forget abandoned activities that became irrelevant over the time period. Therefore, the techniques developed in this dissertation conceptually address the current research challenges in activity recognition with streaming data.

2.9 Summary and Conclusion

In this chapter, we reviewed state-of-the-art approaches for the two overlapped research areas concerning activity recognition in stream mining. We first introduced the area of activity recognition in general. We reviewed the variety of sensors applied for activity recognition along with the different kinds of activities to be predicted. Then, the known term of *i.i.d.* in statistical and probability theory that refers to independent and identically distributed- ness is introduced. A key challenge with learning from sensors in a streaming environment is that it requires learning beyond identically distributed and independent sensory data. We then illustrated the position of our research scope from the two active areas of research in techniques concerning the distribution of data (stream mining) and techniques concerning data dependency (activity recognition).

Activity recognition techniques along with different approaches are depicted in detail. We survey two main categories of activity recognition. The first category focuses on the learning approaches; the other one concerns the dynamic capabilities beyond the learning process. Most of the developed techniques in activity recognition are based on supervised learning, however few studies considered unsupervised and semi-supervised learning considering the limited availability of labelled data. Beyond learning, different approaches for adapting the initial model are discussed. This includes adaptation for model personalisation or for adding/removing activities. The dynamic capabilities

are related to the concept of transfer learning in terms of transfer between the static domain of modelling to the dynamic streaming deployment domain.

Deployment of activity recognition in streaming settings imposes many challenges. In order to understand the limitation and constraints imposed with stream mining for activity recognition, we survey state-of-the-art techniques for stream mining. Stream mining techniques correspond to the distribution oriented approach. Data that evolves in the stream is not identically distributed. In the literature, many approaches have addressed the handling of the high speed and infinite stream of data. We reviewed in this chapter different techniques developed for handling streaming data. Unlike traditional machine learning approaches, new approaches have developed to address the streaming nature of data. Various kinds of changes may encounter data streams over time. We presented in this chapter the methods of both tracking these changes and learning from these changes.

Upon surveying the main approaches for both activity recognition and stream mining, we presented the overlap between the two research areas. We first presented a conceptual mapping between the two areas to demonstrate the connection. We then discussed techniques that considered activity recognition with data streams. The research gaps based on the discussed state-of-the-art approaches were illustrated. Then, the key techniques in the literature that target subset of the research gaps were presented in detail along with a comparison between them. Finally, the chapter concluded with a description of the contribution of this dissertation in addressing the discussed research gaps with linkage to the research objectives.

Table 2.3: Comparison of Key Techniques for Activity Recognition

Criterion	Krishnan and Cook [KC14]	Rashidi and Cook [RC10]	Gomes et al. [GKG+12a]	Do et al. [DLL13]	Lockhart and Weiss [LW14]	Zhao et al. [ZCL+11]	Pärkkä et al. [PCE10]
Aim	Evaluate sliding window approaches for handling streaming data in activity recognition	Find sequential activity patterns in data streams using multiple time granularities	Personalise activity recognition model in data streams	Integrate stream reasoning with activity recognition techniques for recognising basic and complex activities	Build a personalised activity recognition system for real time recognition	Build a robust activity recognition system across people on smart phone	Apply adaptable tree based model for real time activity recognition
Learning approach	Stream	Stream	Stream	Stream	Stream	Static (semi-supervised)	Static (supervised)
Data collection platform	Smart home (binary motion sensors)	Smart home (binary motion sensors)	Environmental and on body sensors	Mobile	Mobile	Mobile	Motion Band
Processing platform	PC	PC	Mobile phone	Mobile & Backend server	Backend server	Mobile	PDA
Handling sensory data	Sliding window	Tilted window	Not provided	Sliding window	Fixed size window	No	No
Classification technique	SVM	DVSM	Incremental NB, DT	ANN	Random Forest	DT& EM	Binary DT
Personalisation approach	No	No	Retrain	No	Retrain	Incremental	Incremental
Discovering new activities	Partially	No	No	No	No	No	No
Forgetting abandoned activities	No	No	No	No	No	No	No
Incorporate active learning	No	No	No	No	No	No	No

Chapter 3

Cluster-based Activity Recognition System

3.1 Introduction

After reviewing state-of-the-art methods developed for activity recognition (AR) in Chapter 2, it is clear that AR still needs attention to address important challenges. This dissertation particularly addresses both the personalisation and adaptation challenges in activity recognition. The solutions developed in this dissertation have to be accurate, efficient, robust, and flexible.

In this chapter, we present our baseline framework for activity recognition. The framework is the foundation for the three techniques that are developed in this dissertation. The key purpose of developing the baseline framework is that it enables personalisation and adaptation through an incremental and continuous learning. Thus, in designing this framework, we focus on developing it in a way that our objectives are met. However, it is equally important to evaluate the performance of the baseline framework in terms of its accuracy, efficiency, robustness, and flexibility. Therefore, that demonstrates the validity of applying the baseline framework for recognition in streaming environments for personalisation and adaptation, which will be discussed in Chapters 4 and 5.

This chapter also sketches the first recognition technique based on the baseline framework. The developed recognition technique is an ensemble classifier based on a hybrid similarity measure approach that is integrated with the baseline framework for recognising activities. In order to test the validity of our ensemble classifier for recognition, we compare it with other standard techniques which have been applied pervasively in activity recognition. The proposed technique has the features of being extendable, to achieve our aims of

personalisation and adaptation by incorporating incremental and continuous stream learning.

The chapter starts with a conceptual description of the baseline framework. This is then followed by a discussion of the desired features of the baseline framework that enable our techniques for personalisation and adaptation. Our first technique that integrates the ensemble classifier for recognition with the baseline framework is also discussed in this chapter. Finally, we evaluate both the baseline framework and the developed technique, in terms of accuracy, efficiency, robustness, and flexibility by conducting extensive analysis with benchmarked datasets in activity recognition.

3.2 Baseline Framework Overview

Three techniques are developed in this dissertation for activity recognition. Each has its own perspective and corresponding goals and objectives. The baseline framework (BLFW) is the centre for all techniques as shown in Figure 3.1. It models the ground truth of historical annotated data collected from different data sources while training the learning model. Our aim in this dissertation is to achieve adaptation and personalisation for activity recognition that naturally deals with data streams. However, a streaming environment presents always-challenging scenarios to handle and process multi-dimensional data that arrives at high speed. Therefore, we initially develop an effective technique for activity recognition that would be the base for activity recognition systems in streaming settings. The early activity recognition technique is coined CBARS, standing for Cluster-based Activity Recognition System. CBARS deploys a novel ensemble classifier in a non-streaming environment. This technique aims to demonstrate the BLFW efficiency independently before deployment in a streaming environment. Also, it is crucial to evaluate the proposed recognition technique in terms of accuracy, efficiency, robustness, and flexibility in a standard environment before imposing more challenges and constraints in the streaming environment. In this way, it is feasible to compare CBARS performance with standard methods in activity recognition that are mostly deployed in non-streaming environments. Based on a CBARS foundation, we develop our STAR technique that extends CBARS by enabling personalisation through incremental learning in a streaming environment. STAR applies a sliding window to process data streams continuously while incrementally personalising the BLFW, to cope with recent changes in the data stream and personalise the model for a specific user. The adaptation process to detect an entirely new activity is performed by our third technique coined COSTAR

which stands for Concept Evolution in data STream for Activity Recognition. COSTAR monitors the evolution of concepts in data streams for continuous adaptation of the recognition model. The aim of COSTAR is to detect the novel activities or eliminate the abandoned ones through continuous learning while streams of activities evolve. COSTAR also assimilates detected changes for detecting future activities. Building the baseline framework that is accu-

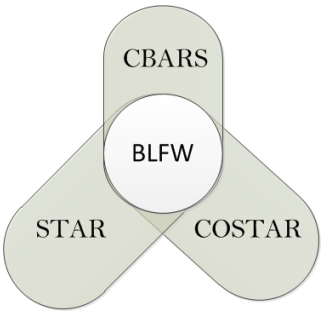


Figure 3.1: BLFW Position of Prospective Techniques

rate, efficient, robust, and flexible is a crucial step in all proposed techniques. We discuss first the desired features of the baseline framework, and then we discuss the approaches and details of building the framework.

3.3 Desired Features for the Baseline Framework

The process of building the BLFW occurs offline on annotated sensory data that is already collected from various sensors. The BLFW is basically a representation of the historical sensory data that contains different activities. The BLFW aims essentially to satisfy a set of key features that we summarise as follows:

- *Accuracy:* Poor, generalised, or over-fitted learning models are the key reasons for an inaccurate recognition of activities. Labelled data is mostly scarce and the annotation process is costly. Activities are poorly modelled when a limited amount of annotated data is applied for training the learning model. On the other hand, a learning model is overfitted when a large mount of data from few users is applied for training. Thus, generalisation can lead to an inaccurate model when applied for a specific user. An accurate modelling of activities requires an adequate understanding of the nature of data in activity recognition. Sensory data that represents users' activities intuitively varies across people. For instance, "walking" from one person could be" jogging" for another. Some

activities contain numerous patterns. For example, a "walking" activity contains patterns such as "strolling", "jogging", or "normal pace walking" patterns. Furthermore, some activities are highly correlated and interleaved with others such as "walking" and "standing". Recognition of a "walking" activity could be challenging because of interleaving "standing" activities. The duration of each activity is also essential information for efficient recognition in real time. Thus, it is essential to build an accurate model that is tailored especially for nature of AR data. In our proposed BLFW, we build a cluster-based model that is fine-grained for accurate representation of the activities. Each cluster is represented by a set of features that best describes the cluster and distinguishes it from other clusters. The quality of these features is the key aspect that influences the model accuracy.

- *Flexibility:* Another significant factor of the BLFW is its ability to cope with data change and perform model personalisation and adaptation with incremental and continuous learning. Model flexibility is of paramount importance in activity recognition. The accuracy is tightly related to the model flexibility. The model flexibility is measured by the ability of the model to adjust for a specific user and adapt for detecting new activities or forgotten abandoned ones. Different users perform the same activity differently. Even for the same user level, changes in the way of performing activities may also occur because of many reasons such as injury and age. New activities may emerge; abandoned activities might gradually disappear. The model that relies only on historical data and is not flexible to change over the time becomes outdated and inaccurate. We aim with building the BLFW to enable incremental and continuous learning from incoming data, therefore the model achieves personalisation and adaptation and thus satisfies the flexibility criterion.

- *Efficiency:* One key challenge in the modelling process is to keep the balance between model accuracy and efficiency. An efficient learning model has to minimise the computational and time complexity. On the other hand, accurate modelling requires a detailed representation of the historical data, to be able to distinguish between different activities. An efficient learning model enables the deployment of the recognition technique on devices with limited resources and attains an accurate recognition in real time. However, a tradeoff between accuracy and efficiency in building the learning model is a crucial challenge in activity recognition. Thus, we develop the BLFW in a way that keeps the balance between the

accurate representation of the activities and the efficiency of the model. To validate the efficiency of the BLFW, we deploy it on mobile phones for achieving accurate recognition in real time.

- *Robustness:* Dealing with sensory data for activity recognition increases the possibility of incoming noise or missing data during transition. In the BLFW, we represent historical data in a robust way to eliminate noise and therefore enhance the recognition accuracy.

All in all, accuracy, efficiency, flexibility, and robustness are four desirable features that we aim to achieve in the proposed BLFW for activity recognition.

3.4 Baseline Framework Description

In this section, we introduce our BLFW for activity recognition that addresses the aforementioned key features. We introduce in this section the developed learning model for AR that is extendible to supporting incremental and continuous learning. Therefore, we can achieve our objectives of personalisation and adaptation with evolving streams. Basically, our BLFW is represented by a set of clusters that correspond to a set of activities. Each activity/cluster is represented by a group of characteristics that best explain each activity and separate it from other activities. Annotated sensory data are used to create clusters, however only summarised characteristics of data are maintained in memory to represent various clusters. We refer to activities in terms of the BLFW as clusters.

3.4.1 Approaches for building BLFW

The lightweight nature of the BLFW demonstrates its efficiency. All raw data is dismissed beyond the model training and leaves only the summarised characteristics of the clusters. The BLFW is also flexible as the summarised characteristics that represent the BLFW enable personalisation and adaptation through incremental and continuous learning. Thus, incremental and continuous learning ensure low computational and time costs while streams evolve with no need of re-processing all the training instances. Moreover, the BLFW deals with noisy data, with either the reduction or elimination approach presenting a robust model. We investigate two approaches for building BLFW clusters from annotated sensory data namely clustering with dominant pattern selection and clustering with sub-clusters.

Clustering with dominant pattern selection approach (DPS)

This approach creates k clusters that represent k activities in the training data. We label each cluster with the majority label among all cluster instances. Then, we extract characteristics of each cluster from its assigned instances.

Training instances typically contain outliers and noisy data that might affect directly the quality of the model representation. Therefore, we add a filtering step that aims to purify clusters and therefore boost the accuracy. We coin the filtering step the **D**ominant **P**attern **S**election process (**DPS**). DPS performs a detailed scan inside each cluster to pick instances that form the dominant pattern inside the examined cluster. Only selected instances are used to build BLFW characteristics. Correspondingly, DPS helps eliminate misclassified, outliers, and noisy data from initially formed clusters. However, building a learning model from only a subset of the data that represents the dominant pattern may result in an over-fitted model and thus cause poor performance at deployment.

Clustering with sub-clusters approach (SC)

We propose another method for building more precise and detailed learning model with sub-clusters. This approach aims to build a fine grained BLFW that not only represents activity but also represents various patterns inside each activity. The fine-grained model allows the separation of the cluster into subsets that have distinguishable characteristics. We refer to these subsets as sub-clusters of the cluster. Sub-clusters of a particular cluster are analogous to different patterns within this particular activity. For example, the activity of "using the stairs" contains patterns for either "going downstairs" or "upstairs". Also, the "running" activity spans patterns such as "serious running for exercising", "relaxed jogging at the park" or "sprinting to catch the train". Moreover, one person's "sprint" could be "jogging" for another. Building a fine-grained model that contains different patterns/sub-clusters within each activity/cluster is essential to enhance the recognition accuracy. Since sub-clusters provide microscopic information about each cluster, a fine-grained model is expected to be more accurate than using the clusters themselves directly.

Moreover, the SC approach reduces the effect of noise on the BLFW representation. This approach tends to aggregate similar data together in the same sub-cluster. Thus, noisy data would be either distributed among sub-clusters or separated in different sub-clusters. This way, the effect of noisy data on the sub-cluster level is lowered. The reduction approach is different from the

elimination approach applied with DPS as it only limits the effect of noise without totally ignoring it.

Besides the fine granularity and noise reduction advantages of the sub-cluster approach, the characteristics that represent the sub-cluster model are also more explanatory. For instance, to measure a cluster boundary in the clustering approach, we compute the maximum distance between the cluster centroid and any instance that belongs to this cluster. Thus, the cluster boundary is highly affected by noise, outliers, or misclassified data. On the other hand, with the sub-cluster approach, boundaries are controlled by a new measure that we term $DMax$, which is defined as the maximum distance between any pair of sub-clusters within the cluster. The framework ensures that sub-clusters represent stable patterns inside the cluster, and therefore $DMax$ provides a more precise representation that is also less sensitive to expected data impurities.

The methodology for building the BLFW starts with applying a supervised learning technique to the annotated sensory data collected for training. Thus, the initial learning model that is composed of a number of clusters is created. Each cluster corresponds to one of the labelled activities that exist in the training data. After creating K clusters, we further split each cluster into smaller sub-clusters. We apply a clustering technique to each cluster to form sub-clusters representing different patterns inside the cluster. The process of building the BLFW with the sub-clusters approach is illustrated in Figure 3.2.

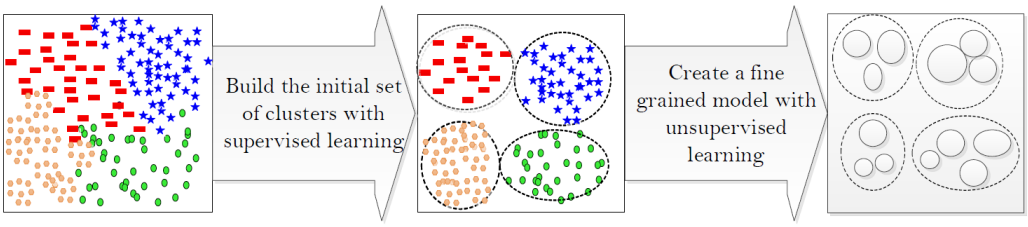


Figure 3.2: Clustering with Sub-clusters Approach for Building the BLFW

Information is extracted in this approach based on clusters and their corresponding sub-clusters. Because of the aforementioned advantages of the sub-cluster approach, we used this approach for the proposed techniques for personalisation and adaptation in this dissertation. We justify our selection by evaluating both approaches in the following sections in this dissertation.

3.4.2 BLFW characteristics

The BLFW is represented by a set of informative characteristics that best describe activities in the training data. We aim in this component at building a robust, flexible, and efficient BLFW that is the basis for successful recognition. To achieve this, we aim to extract a summary of the statistics of the each cluster, before we dismiss all raw data instances at the end of the modelling process. Characteristics of the BLFW span across two levels: cluster characteristics and holistic characteristics. Cluster characteristics are the extracted information that describes each cluster/activity and distinguishes it from others. Holistic characteristics describe the training data as a whole with global features that explain the entire data. We also extend the explanatory characteristics by adding a new level of description for sub-clusters, which is used for the sub-cluster approach in building the BLFW. Sub-cluster characteristics provide an abstract description of each pattern/sub-cluster that distinguishes it from other sub-clusters belonging to the same cluster. Sub-cluster characteristics include the following:

- $Weight_{sc}$ is the total number of data instances that belong to the sub-cluster (sc).

- $Centroid_{sc}$ is one of the most important extracted characteristics. It locates the centre of each sub-cluster in the data domain. For n-dimensional data instances, $Centroid_{sc}$ is an n-dimensional vector of the mean value of the n-dimensional instances inside the sub-cluster as per Equations 3.1 and 3.2.

$$Centroid_{sc} = \{d_1, d_2, d_3, ...d_n\} \tag{3.1}$$

  Where,

$$d_j = \frac{\sum_{i=1}^{Weight_{sc}} P_{i,j}}{Weight_{sc}} \tag{3.2}$$

  Where d_j is the centroid of the jth feature, and $P_{i,j}$ is the jth feature of the ith data instance within the sub-cluster (sc).

- $WISCSD_{sc}$ (WIthin Sub-Cluster Standard Deviation) measures the dispersion of instances within the sub-cluster. Each activity pattern which is represented by a sub-cluster in the BLFW has its own standard deviation. Standard deviation of an n-dimensional instances within the sub-cluster is calculated as shown in Equation 3.3.

$$WISCSD_{sc} = \sqrt{\frac{\sum_{i=1}^{m}(EDistance(P_i, Centroid_{sc}))^2}{Weight_{sc}}} \tag{3.3}$$

Where $EDistance$ is the Euclidean Distance and P_i is an n-dimensional data instance inside the sub-cluster (sc).

- $Radius_{sc}$ is the maximum distance between the sub-cluster centroid and any data instance belonging to the sub-cluster as explained in Equation 3.4.

$$Radius_{sc} = \max\{EDistance(P_i, Centroid_{sc}) \forall P_i \in sc\} \tag{3.4}$$

Where $EDistance$ is the Euclidean Distance, P_i is an n-dimensional data instance inside the sub-cluster (sc), and $Centroid_{sc}$ is the sub-cluster centroid.

- $AvDist_{sc}$ is the average distance between instances and the centroid within the sub-cluster. It is defined as the sum of the distances between sub-cluster data instances and their respective centroid divided by the number of instances (weight). The average distance is calculated as per Equation 3.5.

$$AvDist_{sc} = \frac{\sum_{i=1}^{m} EDistance(P_i, Centroid_{sc})}{Weight_{sc}} \tag{3.5}$$

Where $EDistance$ is the Euclidean Distance and P_i is an n-dimensional data instance inside the sub-cluster.

- $Density_{sc}$ (density of sub-cluster) is defined according to the density function in Equations 3.6 and 3.7.

$$Density_{sc} = \frac{Weight_{sc}}{Volume_{sc}} \tag{3.6}$$

Where,

$$Volume_{sc} = \frac{4}{3}\pi {Radius_{sc}}^n \tag{3.7}$$

Where $Volume_{sc}$ is the volume of the sub-cluster as a hypersphere in an n-dimensional space. With high dimensional datasets,we deal with data as in 3-D hypersphere (n=3) to avoid a possible curse of dimensionality.

The other level of characteristics is the one that represents clusters. **Cluster characteristics** represent a higher level description of the extracted features corresponding to clusters/activities. The description of cluster characteristics is as follows:

- $Nsub_c$ is the number of sub-clusters within the cluster that corresponds to the number of patterns inside an activity cluster. It has a value of 0 when applying a DPS approach to building the BLFW.

- $Centroid_c$ is the mean of all sub-clusters centroids that belong to the cluster when applying the SC approach to building the BLFW. The centroid is defined in Equation 3.8.

$$Centroid_c = \frac{\sum_{i=1}^{Nsub_c} Centroid_{sc}}{Nsub_c} \tag{3.8}$$

  In the case of applying the DPS approach to building the BLFW, Equations 3.1 and 3.2 are applied for calculating the centroid from all data instances in the cluster.

- $Weight_c$ is the total number of data instances within the cluster.

- $Radius_c$ measures and controls the decision boundary of each cluster in the domain space. It is defined as the maximum distance between the cluster centroid and all data instances inside the cluster. $Radius_c$ is calculated as per Equation 3.9.

$$Radius_c = \max\{EDistance(P_i, Centroid_c) \forall P_i \in c\} \tag{3.9}$$

  Where $EDistance$ is the Euclidean Distance and P_i is an n-dimensional data instance inside the cluster.

- $WICSD_c$ measures the dispersion of data instances within the cluster. Standard deviation is calculated according to Equation 3.3, with cluster centroid and weight measures.

- $Density_c$ is calculated according to Equation 3.6, with cluster $Weight_c$ and $Radius_c$ replacing $Weigh_{sc}$ and $Radius_{sc}$ respectively.

  The following two measures of GF and $DMax$ are only extracted for the BLFW with sub-clusters.

- GF is the gravitational force among sub-clusters of the cluster. Inspired by physical laws, there exists a natural attraction force between any two objects in the universe and this force is called the gravitational force. According to Newton's universal law of gravity, the strength of gravitation between two objects is directly proportional to the product of the masses of the two objects, but inversely proportional to the square of distance between them. The gravitational force among sub-clusters is

defined in Equation 3.10. GF characteristic, as per Equation 3.10, is a two dimensional array of the gravitation force generated among each pair of sub-clusters within the cluster.

$$GF_{i,j} = G\frac{Weight_{sc_i}Weight_{sc_j}}{r^2} \tag{3.10}$$

Where G is the constant of universal gravitation; $Weight_{sc_i}$ is the weight of sub-cluster i; $Weight_{sc_j}$ is the weight of sub-cluster j; r is the Euclidean distance between sc_i and sc_j centroids.

- $DMax_c$ is the maximum distance between any pair of sub-clusters centroids within the cluster.

Finally, the most abstract level represents the **holistic characteristics** for the entire training data. Thus, the holistic characteristics represent a collective view on the data. The characteristics are as follows:

- $Nclus$ is the number of clusters/activities in the training data.
- $Centroid_{glob}$ is the global centroid of all instances in the training data, which is analogous to the centre of all clusters' centroids.
- $Capacity$ is total number of processed training instances.

All of the extracted characteristics facilitate incremental and continuous learning due to their simplicity of calculation. After extracting all characteristics in the modelling process, a cluster-based, fine-grained and lightweight BLFW becomes ready for recognition. All raw data is dismissed at the end of this modelling process.

3.5 CBARS: Cluster-Based Activity Recognition System

In this section, we present our first technique for activity recognition based on the BLFW. We coined our system CBARS, which stands for **C**luster **B**ased **A**ctivity **R**ecognition **S**ystem [AGSK12a]. CBARS contains two components, as shown in Figure 3.3, a modelling component and a recognition component.

In the modelling component, we deploy the BLFW that is discussed earlier in this chapter. The BLFW has the flexibility feature that enables incremental and continuous learning for personalisation and adaptation in Chapters 4 and 5. For the recognition component, we develop a novel ensemble classifier

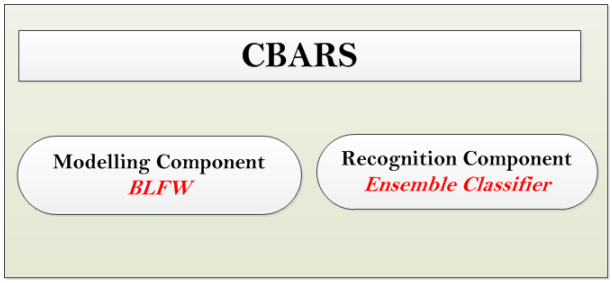


Figure 3.3: CBARS Components

for efficient recognition based on a hybrid similarity measure approach. Although personalisation with incremental and continuous learning is not the aim of CBARS, the ensemble classifier attains an improved accuracy across different users because of the hybrid similarity measure approach. CBARS also introduces an active learning approach to deal with the challenge of the scarcity of the labelled data in activity recognition. The focus in this section is to present the recognition component in CBARS that integrates with the modelling component for an efficient recognition system. In the following, we first discuss the essential features of the proposed ensemble classifier in the recognition component. Then, we present the methodology of the ensemble classifier in detail.

3.5.1 Features of the ensemble classifier

This section answers the question of "what technique can be applied to the BLFW for an accurate and efficient recognition of activities?". This technique has to essentially consider both the learning model structure and the activity recognition domain for an efficient recognition. Thus, we propose a novel ensemble classifier that applies a hybrid similarity measure approach for this task. The ensemble classifier is a cluster-based approach that deploys multiple measures to match similarities between incoming data and the BLFW. The proposed ensemble classifier is customised to suit both the BLFW and activity recognition, by providing the following advantages:

- *Cluster-based approach:* Activity recognition data is noted for its sequential presentation of activities. Thus, data representing different activities arrives in batches. In this case, processing each individual instance causes unnecessary processing and time overhead. Furthermore, the cluster-based approach is compatible with the nature of the cluster-based framework (BLFW). Thus matching the similarities is based on the same characteristics. Also, some activities are unavoidably interrupted by the individual instances of other activities. For instance, the atomic

activity of "running" may be interleaved with short periods of "walking", yet the major activity for the entire time elapsed is "'running" despite the existence of few "walking" instances. The cluster-based approach effectively handles this common case by considering data collectively and decides based on the majority in the cluster instead of responding to each instance.

- *Collective measure perspectives:* The ensemble classifier studies various perspectives with a hybrid similarity measure approach. It presents insights with four measures corresponding to internal cluster cohesiveness and closeness as well as interactions among clusters in the entire domain distribution. Combining various perspectives yields benefits from each perspective and minimises the anticipated drawbacks of applying only a single perspective. Activity recognition leverages a variety of perspectives in understanding data insightfully. Moreover, applying the hybrid similarity measure approach enhances the recognition accuracy across different users. When we use a general model for recognising activities of a particular user, activities in the general model may be differentially presented. For instance, data that represents a "walking" activity for a specific user may have been presented as "jogging" in the general model. The hybrid similarity measure approach brings different perspectives together for more accurate recognition across users. In this example, the two representations of the same activity are spatially far apart, unlikely to be recognised as the same activity. However, the decision that is based not only on one measure but using multiple is instead expected to improve the accuracy. Thus, the hybrid similarity measure approach tends to reduce the problems related to poor or over-fitting of the learning model.

- *Lightweight measures to match similarities:* Another advantage of the ensemble classifier for activity recognition is its lightweight measures that are simple to compute and update. The measures are distance, density, gravitational force and deviation. The four measures are based on simple calculations using only elementary BLFW characteristics. The ensemble classifier matches the similarities between incoming data and BLFW characteristics. To apply the similarity test for each measure, fast computations using simple characteristics such as distance between centroids and number of instances inside each cluster are performed. The easy to compute, easy to update measures facilitate BLFW being computationally efficient.

- *Effective active learning:* The ensemble classifier enables active learning that addresses the scarcity of labelled data, which is a well-known issue when dealing with sensory data. Annotating data on a big scale is a costly, erroneous, and time consuming process. Therefore, selecting a small subset of data to annotate is crucial in any learning technique. Active learning is applied as an extension for our ensemble classifier. As the novel ensemble classifier is based on a hybrid similarity measure approach, decision confusion among measures triggers automatically the need for active learning. We select the data automatically with no computational overhead, during the recognition process. Data that causes confusion among measures is expected to be the most uncertain and indeed requires user input for ground truths. Active learning enhances overall system performance by feeding true labels into the system for future improved recognition of similar data.

To introduce the details of the ensemble classifier, we first discuss the measures themselves. Then we present how these measures are applied in the ensemble classifier for activity recognition.

3.5.2 Hybrid similarity measure approach

A key success of any machine learning technique is the similarity measure applied for learning and recognition. Each single measure has its own strengths and weaknesses. Thus, combining and voting among various measures would have the benefit of bringing together different perspectives of the data. For example, Figure 3.4 shows that, point A would be seen initially as a member of the small cluster from the closeness perspective. However, taking into consideration other measures perspectives, such as the density and size of the small and big clusters, will lead us to better understanding and more accurate classification decision-making.

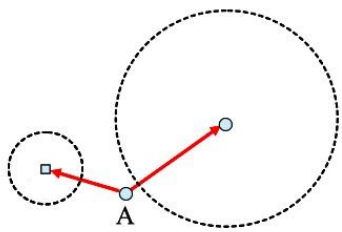


Figure 3.4: Example of Misclassification Using a Single Similarity Measure

Basically, assigning a data point to a cluster does not depend only on how close it is from its centroid, but also other parameters associated with the whole

data distribution are of paramount importance. Therefore, we propose our ensemble classifier that applies four different measures for accurate recognition. Applied measures are *distance*, *density*, *gravity*, and *deviation*. While the distance focuses on the closeness metric amongst data, the density measure concerns the impact of data assignment on the cohesiveness of data. The relationship between the closeness metric and cluster weight is the focus of the gravitational force measure. Each cluster has its own gravitational force that is generated from its weight. The heavier the cluster, the more data instances that it can attract due to its gravitational force. Unlike density that focuses on the global cohesiveness among clusters, deviation concerns the cohesiveness of each cluster individually. Details of the four measures are as follows:

- *Distance:* Many well-known learning techniques such as K Nearest Neighbor (KNN) rely on the distance measure for finding similarities among data [YJ06]. The distance-based learning classifies similar data as the closest in a given collection of similar/dissimilar data instances. The distance measure focuses primitively on the closeness and the separation of incoming data from training data. In activity recognition, the distance measure has been applied pervasively, such as in [RDML05, BI04, BGC09]. We also apply the distance measure for activity recognition, yet we combine it with other measures for a comprehensive view of the data from different perspectives. We implement the distance metric using the Euclidean distance. The distance between any two data instances with n features, $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$, is defined as in Equation 3.11

$$EDistance(P, Q) = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2}\,. \tag{3.11}$$

- *Gravity:* This measure on the other hand concerns not only the closeness but also the weight of the clusters. Each cluster has its own gravitational force. Thus, this measure focuses more on the attraction among clusters caused by their gravitational force. The gravitational field is limited to the slack that surrounds each cluster and proportionally corresponds to the cluster weight. When incoming data is located inside the gravitational field of any existing cluster, it is more likely to be attracted by its gravitational force. Figure 3.5 shows an example of the gravitational field (dotted lined circles) of each cluster generated from its weight. Although the Cluster 3 radius is smaller than the Cluster 2 radius, the weight of Cluster 3 causes a larger surrounding gravitational field than Cluster 2.

It has to be noted that cluster weight is different from cluster density. For example, in the explanatory figure, Cluster 1 is of heavy weight and thus its gravitational field has a large slack though the cluster is of low density.

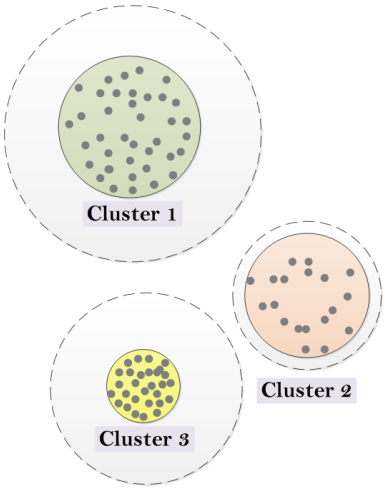


Figure 3.5: Illustration of Clusters Gravitational Force

The gravitational search algorithm (GSA) is an efficient algorithm based on the law of gravity and mass interactions as in Equation 3.10 [RNpS09]. The concept of the gravity has been applied in machine learning for both clustering [OCY01, EFWD88] and classification [BNpBS12]. Other techniques combine the gravitational concept with the distance measure as in [HANp12, YHY+11, RNp14]. In our ensemble classifier, we also combine gravity and distance in addition to two other measures for an efficient recognition of activity.

- *Density:* Unlike the distance and gravity measures, the density measure focuses on data cohesiveness and dispersion for both the cluster level and collective level of all clusters. Density-based algorithms have been applied for different purposes in machine learning. One of the most popular techniques is DBSCAN [EKSX96] that implements data clustering based on a "density reachability" concept which connects points based on a distance threshold. OPTICS [ABKS99] is a generalised form of DBSCAN yet with different parameters. Other studies have considered the density measure for clustering in data streams such as in [CEQZ06], while Hempstalk, Frank and Witten [HFW08] presented a new technique for novelty detection and outlier detection based on the density measure. Due to the successful implementation of the density concept for machine learning, the density measure is among the collection of measures that is used by our ensemble classifier for activity recognition.

- *Deviation:* Besides density, deviation focuses on a cluster's internal cohesiveness around the centroid. Thus, this measure considers the cohesion and dispersion internally in each cluster. K-medians clustering [JD88], for instance, is different from K-means as it calculates the median (based on the deviation) instead of the mean (centroid). Deviation has also been used for outlier and change detection. Aggrawal [Agg13] defined a deviation-based method as the one that measures the impact of new data on the data variance. For example, the method proposed in [AAR96] measures the impact on the cluster variance when a particular data point is removed/assigned. Gaber and Yu [GY06] presented an algorithm that captures changes in data stream distribution or domain by using clustering result deviation. The standard deviation of different activities is used pervasively in activity recognition to find similarities or distinguish between different activities [LW14, KWM11].

In general, ensemble classifiers apply multiple learning algorithms for enhancing the classification performance obtained when applying a single algorithm. There are different approaches of ensemble classification. Bagging is a well-known approach that trains multiple models on random subsets of the training data. Then it applies voting among the models for prediction. Random forest [Bre01] is an example of an ensemble classifier that uses the bagging approach for combining multiple decision trees. Boosting is another type of ensemble classification that involves building an ensemble by training each model with misclassified instances from other models to boost the performance. Adaboost [FS97] is one of the most common classification techniques that combines weak learning for producing a boosted classifier.

Our ensemble classifier, on the other hand, classifies data based on the same training data and using the four measures. It applies an equal weighted voting among measures for predicting the classification output. The more effective measures we include, the more accurate learning we achieve. At this stage, we apply the four measures that assess both closeness and cohesiveness for each cluster and also among clusters collectively. For each measure, a set of candidates are chosen and ranked, and then we apply voting to choose the best candidate amongst all ranked candidates.

We applied the hybrid similarity measure approach for machine learning techniques in [AG11, AG09, AG15]. In the next section, we explain the methodology of applying the proposed ensemble classifier based on the hybrid similarity measure approach for activity recognition.

3.5.3 Methodology

In this section, we explain the details of CBARS along with its two components: modelling and recognition. We start the process in CBARS by building the BLFW with one of the two approaches discussed earlier in this chapter. In the recognition component, the ensemble classifier is applied to match similarities between the BLFW and new data for recognising activities. Active learning is triggered in the event of confusion in the recognition decision among different measures. Figure 3.6 illustrates the recognition component in CBARS.

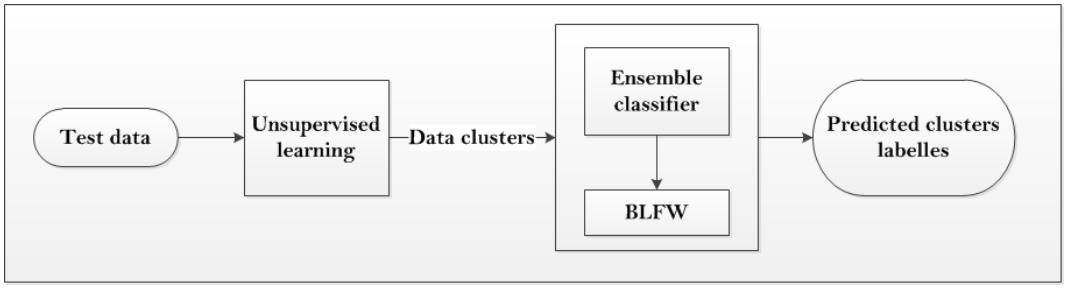


Figure 3.6: Recognition Component in CBARS

From the learning perspective, CBARS contains three consecutive phases. The first phase corresponds to the modelling component, while the other two phases are implemented in the recognition component. The three phases are described as follows:

- *Building the BLFW (phase 1):* The process of any recognition technique starts with building the BLFW from historical data for training (the modelling component in CBARS). The generated BLFW consists of a set of clusters. Each cluster represents one of the labelled activities that exists in training instances. BLFW extracts features from each cluster and dismisses all raw data at the end of this phase.

- *Unsupervised learning for incoming data (phase 2):* This step occurs only once and aims to create clusters of various activities existing in incoming test data. When unlabelled data emerges, we apply clustering on data to generate clusters of the performed activities. Various clustering techniques such as K-means [Har75], Expectation Maximisation [DND77] and DBScan [EKSX96] have been used. The more pure clusters we produce, the more accurate recognition we attain. In other words, the accuracy of unlabelled data clustering method impacts directly the decision of the hybrid similarity measure method and thus the recognition accuracy. The characteristics of the generated clusters are also extracted and all raw data is dismissed similar to the BLFW building procedure. The

output of this phase is a set of cluster/activity characteristics ready for the following recognition phase.

- *Recognition phase (phase 3):* We assess incoming data clusters with the ensemble classifier to predict the clusters' labels. The ensemble classifier matches the similarities between incoming data and BLFW clusters, for recognising the correct label of the cluster. Clusters are similar if they match based on the hybrid similarity measure approach. The activity label is selected based on an equal weighted voting procedure among measures.

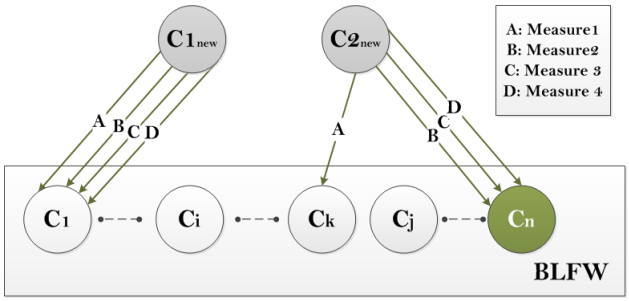
  For each incoming data cluster, the algorithm checks how similar it is to other clusters existing in the BLFW. As we apply various measures to test similarities among clusters, each measure votes for its own "candidate" from the measure perspective. The predicted label is the candidate cluster with the majority of votes among all measures, while the true label of a cluster is the majority true label among all cluster instances. There are three cases expected from the voting procedure explained in Figure 3.7. They are illustrated as follows:

  i *Correct recognition:* This case occurs when the majority of votes have chosen the correct candidate cluster/activity. That means the predicted label of the cluster, with the majority votes, matches its true label. Figure 3.7 (a) shows the two scenarios of correct prediction. The new clusters ($C1_{new}$ and $C2_{new}$) are correctly classified if at least three measures vote for the correct candidate cluster.

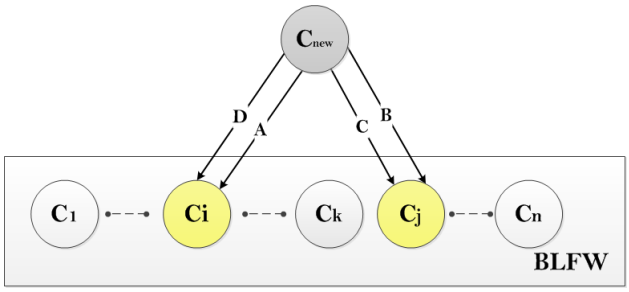
  ii *Active learning:* In this case, equal votes are assigned to exactly two candidate clusters. For example, when both distance and density vote for the candidate cluster that presents the "walking" activity, deviation and gravity vote for "jogging". The algorithm inquires about the correct (true) label from the user in an active learning mode with either of the labels of the two nominated candidates. In Figure 3.7 (b), C_{new} has two nominated clusters (C_i and C_j) with equal votes assigned to each.

  iii *Incorrect recognition:* The cluster is incorrectly classified in one of two scenarios. The first one occurs when there is a total confusion with the lowest confidence among all measures. The other scenario occurs when the majority vote goes for an incorrect cluster, where the predicted label does not match the true label of the cluster. $C1_{new}$

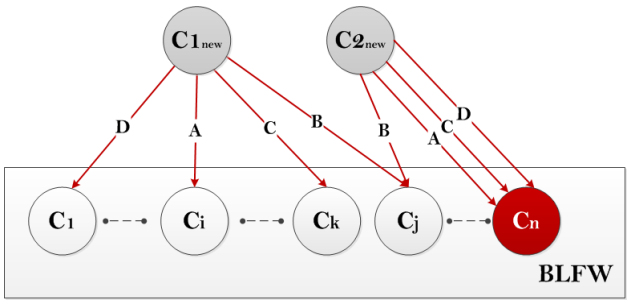
and $C2_{new}$ explain the two scenarios for incorrect recognition in Figure 3.7 (c). The total confusion occurs with $C1_{new}$ when each measure votes for a different candidate cluster. The other scenario, with $C2_{new}$, indicates consensus among measures however on the incorrect candidate, C_n in the explanatory figure.



(a) Correct Recognition



(b) Active Learning



(c) Incorrect Recognition

Figure 3.7: Illustration of the Three Recognition Cases

The ensemble classifier based on a hybrid similarity measure approach implements the aforementioned four measures namely distance, density, gravity, and deviation. The BLFW is either built with the clustering approach with

dominant pattern selection (DPS) or with the clustering with sub-clusters approach (SC). BLFW with DPS consists of n clusters/activities as per Equation 3.12.

$$BLFW_c = \{C_1, C_2, C_3, ...C_n\} \tag{3.12}$$

The BLFW with SC approach that consists of n clusters and m sub-clusters across all n clusters is defined as in Equation 3.13:

$$BLFW_{sc} = \{C_1\{sc_{11}, ...sc_{1P_1}\}, C_2\{sc_{21}, ...sc_{2P_2}\}, ...C_n\{sc_{n1}, ...sc_{nP_n}\}\} \tag{3.13}$$

Where m, the total number of sub-clusters is $\sum_{i=1}^{n} P_i$. The number of sub-clusters varies from one cluster to another based on the diversity of patterns within the cluster. When a new cluster of incoming data C_{new} emerges, the ensemble classifier applies the four measures to choose the best candidate cluster among n clusters, with m sub-clusters in the case of the sub-cluster type of BLFW. The four assessing measures are as follows:

Distance measure is the basic measure to check the closeness of the incoming data cluster from BLFW clusters. C_j is the candidate cluster from the distance perspective, if the distance between centroids of C_{new} and $C_j \in BLFW_c$ or $sc_i \in C_j$ for $BLFW_{sc}$ is the shortest among the n clusters (with m sub-clusters in $BLFW_{sc}$). Algorithm 3.1 illustrates procedure of selecting the distance candidate.

Algorithm 3.1 FindDistanceCandidate

Input: $BLFW$: the baseline framework is either $BLFW_c$ or $BLFW_{sc}$
C_{new}: incoming data cluster
Output: $DisCand$: best candidate from the distance perspective

```
1: if BLFW is BLFW_c then
2:     for all C_j ∈ BLFW_c do
3:         temp[j] ← EuclideanDistance (Centroid_Cj, Centroid_Cnew)
4:     end for
5: else
6:     for all C_j ∈ BLFW_sc do
7:         for all sc_i ∈ C_j do
8:             temp[j][i] ← EuclideanDistance (Centroid_sci, Centroid_Cnew)
9:         end for
10:    end for
11: end if
12: Choose j with shortest distance in temp
13: DisCand= C_j
```

Therefore, the procedure chooses the candidate cluster by calculating the distance between centroids. The algorithm searches the shortest distance among clusters with the $BLFW_c$ that is built by clustering with the DPS approach (lines 1–4). Whereas it searches the candidate on the sub-cluster

level with the $BLFW_{sc}$ with sub-clusters as in (lines 5–11). The cluster with the shortest distance is the candidate cluster from the distance perspective.

Among the valid measures is the **density** which measures the degree of cohesiveness between the new cluster and existing BLFW clusters. In order to choose the best candidate cluster from the density perspective, we check the virtual density difference (VDD) for both gain or loss. The VDD computes the virtual value of gain/loss when the new cluster is merged with the existing BLFW cluster and compared to the original cluster density ($Density_c$). The density of the cluster/sub-cluster is calculated as per Equation 3.6. C_j is the best candidate from the density perspective if the VDD when C_{new} merged with C_j, or with sc_i where $sc_i \in C_j$ (in case of $BLFW_{sc}$), attains the minimum loss or maximum gain. Procedure of *FindDensityCandidate* is explained in Algorithm 3.2. The virtual density difference (VDD) is the difference between the cluster/sub-cluster density before and after merging the incoming cluster data (lines 3 and 8). The cluster that achieves the maximum gain or minimum loss among all clusters is selected as the candidate from the density perspective.

Algorithm 3.2 FindDensityCandidate

Input: $BLFW$: the baseline framework is either $BLFW_c$ or $BLFW_{sc}$
C_{new}: incoming data cluster
Output: $DenCand$: best candidate from the density perspective

```
1: if BLFW is BLFW_c then
2:     for all C_j ∈ BLFW_c do
3:         temp[j] ← calculateVDD (C_j, C_new)
4:     end for
5: else
6:     for all C_j ∈ BLFW_sc do
7:         for all sc_i ∈ C_j do
8:             temp[j][i] ← calculateVDD (sc_i, C_new)
9:         end for
10:    end for
11: end if
12: Choose j with maximum gain/minimum loss in temp
13: DensCand= C_j
```

The third measure we apply in the ensemble classifier is the **gravity**. According to Equation 3.10, each cluster generates its own gravitational force created from its weight. The bigger the weight of the cluster the stronger the gravitational force produced around it. Therefore, the probability it could attract more data instances is increased. When the gravitational force between C_{new} and C_j (or $sc_i \in C_j$) is the biggest among all n clusters (with m sub-clusters in $BLFW_{sc}$), then C_j is the best candidate from the gravitational force perspective. The gravitational check procedure is explained as in Algorithm 3.3.

Algorithm 3.3 FindGravityCandidate

Input: $BLFW$: the baseline framework is either $BLFW_c$ or $BLFW_{sc}$
C_{new}: incoming data cluster
Output: $GravCand$: best candidate from the gravity perspective

```
if BLFW is BLFW_c then
    for all C_j ∈ BLFW_c do
        temp[j] ← calculateGravitationalForce (C_j, C_new)
    end for
else
    for all C_j ∈ BLFW_sc do
        for all sc_i ∈ C_j do
            temp[j][i] ← calculateGravitationalForce (sc_i, C_new)
        end for
    end for
end if
Choose j with maximum gravitational force in temp
DensCand= C_j
```

The **deviation** measure shows the cohesiveness or dispersion of data inside the cluster. Clusters that have similar standard deviation are more likely to present the same activity/cluster. The standard deviation of n-dimensional points inside the cluster is calculated as shown in Equation 3.3. Clusters that have similar standard deviation are more likely to present the same activity. We investigate the impact of a new cluster, assumed to join one of existing BLFW clusters, by using the cluster standard deviation. The selected candidates from the standard deviation perspective are the ones with the lowest impact on standard deviation within the BLFW.

Therefore, this measure shows how much variation or dispersion between the C_{new} and BLFW clusters/sub-clusters. Each cluster/sub-cluster has its own standard deviation. C_j is the candidate cluster from the deviation perspective if it has the smallest difference in the deviation measure between C_{new} and C_j, if $BLFW_c$, or $sc_i \in C_j$ with $BLFW_{sc}$. Algorithm 3.4 describes the procedure of selecting the deviation candidate.

The overall process of CBARS is described in Algorithm 3.5. The process starts with phase 1 (line 1) for building the BLFW from training data. When new data arrives, unsupervised learning is applied to cluster incoming data in phase 2 (line 2). The ensemble classifier applies the four measures to obtain the candidate from each perspective (lines 4–7). The hybrid similarity measure approach chooses the candidate cluster with the majority vote among the four measures (line 8). We consider all measures equally weighted. Therefore, the confidence level of choosing C_j has to be more than or equal to 50%, i.e. at least 2 measures vote for the same cluster would be sufficient to choose the cluster as the best candidate. The two cases of active learning occur when a confusion

Algorithm 3.4 FindSDCandidate

Input: $BLFW$: the baseline framework is either $BLFW_c$ or $BLFW_{sc}$
C_{new}: incoming data cluster
Output: $SDCand$: best candidate from the deviation perspective

```
if BLFW is BLFW_c then
    for all C_j ∈ BLFW_c do
        temp[j] ← SDDifference (C_j, C_new)
    end for
else
    for all C_j ∈ BLFW_sc do
        for all sc_i ∈ C_j do
            temp[j][i] ← SDDifference (sc_i, C_new)
        end for
    end for
end if
Choose j with the minimum difference in temp
SDCand= C_j
```

among measures happens or when equal votes are given to exactly two candidates (groups of two measures nominated the two candidates). Therefore, CBARS inquires user automatically based on the learning algorithm for the true label of the cluster.

Algorithm 3.5 CBARS

Input: $Data_{historical}$: historical training data
$Data_{new}$: incoming test data
Output: $y'[]$: activity labels

```
BLFW ← ModellingComponent (Data_historical)
Clusters_new[] ← unsupervisedLearning (Data_new)
for all C_new ∈ Clusters_new[] do
    DisCand ← FindDistanceCandidate (C_new, BLFW)
    DensCand ← FindDensityCandidate (C_new, BLFW)
    GravCand ← FindGravityCandidate (C_new, BLFW)
    SDCand ← FindSDCandidate (C_new, BLFW)
    y'_new ← Vote (DisCand, DensCand, GravCand, SDCand)
end for
```

In summary, CBARS consists of two components: modelling and recognition. The modelling component in CBARS builds the BLFW that could be either clusters with DPS or clusters with sub-clusters. While the recognition component applies an ensemble classifier based on a hybrid similarity measure approach for recognising activities from incoming data. In the following sections, we discuss the CBARS contribution and then evaluate and discuss the performance of CBARS components with benchmarked datasets for activity recognition.

3.6 Contributions of CBARS

This chapter presented the implementation of CBARS and its two components: modelling and recognition. The proposed and developed components in CBARS are then extended for personalisation and adaptation in Chapters 4 and 5. The contributions of CBARS are described across its two components as follows:

- *Building flexible, robust, and efficient learning models:* The BLFW that is built in the modelling component in CBARS enables an incremental and continuous learning for personalisation and adaptation, which will be discussed later in this dissertation. Both approaches for building the BLFW aim to construct a computationally efficient framework where all training data is dismissed after extracting a summary of characteristics that best present the historical data. Both approaches present solution for dealing with noisy data by either elimination or reduction. The BLFW either eliminates noisy data by including only the data that represents the dominant pattern inside the clusters and ignoring noise, or reducing the effect of noisy data. Thus, the CBARS modelling component builds a learning model that is flexible, efficient, and robust.

- *Accurate recognition across users:* The ensemble classifier applied in the recognition component of CBARS uses a hybrid similarity measure approach for classification. Applying the hybrid similarity measure approach for activity recognition shows superiority over the use of individual measures by bringing various perspectives together, and therefore enhances recognition accuracy especially when used across users.

- *Automated active learning that is triggered based on the recognition decision:* Active learning in CBARS is part of the recognition component. The trigger of active learning happens automatically based on the recognition decision when the measures are confused between candidate clusters. Therefore, active learning is implemented efficiently as it does not require any learning overhead.

In addition to aforementioned advantages, CBARS is adaptable to the nature of activity recognition data with the cluster-based approach. Moreover, it is a sensor type and location independent technique for activity recognition. In the following, we carry out an extensive analysis to validate and demonstrate the efficiency of CBARS.

3.7 Experimental Study

This section reports the experiments conducted, to study how CBARS performs in practice. We evaluate various aspects of CBARS performance with different benchmarked datasets for activity recognition. CBARS deployment, analysis, and evaluation are performed in the standard non-streaming environment. Precisely, 'standard' in this context refers to the traditional machine learning approach which eliminates many of the constraints that occur in a streaming environment. Intuitively, the streaming environment imposes more constraints for data processing and prediction. A data stream is typically an unlimited sequence of data that arrives at high speed and requires a mostly real time response. Streaming settings allow only linear/sub-linear data processing. Moreover, the data storage capability is very limited due to the resource constraints required for processing non-stop streaming data. Chapter 4 presents an extension of CBARS in a streaming environment. However, in this chapter, we discuss the deployment of CBARS in the standard non-streaming environment. To analyse the performance of CBARS, we aim in this section to evaluate the following:

- *Aim 1: investigate the efficiency of modelling component approaches:* The evaluation aims at testing and analysing the two approaches of building the BLFW. Testing in a standard non-streaming environment keeps an adequate separation between the framework performance, which is very crucial, and the streaming constraints, with their impact on performance. The solo evaluation of the framework in a static environment, without posing additional factors that appear in the streaming environment, allows more accurate evaluation of the framework approaches. The two approaches are evaluated in terms of their robustness to noise and impact on the ensemble classifier performance.

- *Aim 2: evaluate the measures performance in the ensemble classifier:* As we hypothesised that various measures cover various perspectives, applying collective measures performs better than applying only a single measure. We focus in CBARS on evaluating the hybrid similarity measure approach as well as assessing each measure individually. Measures evaluation is more applicable in a static environment, as more parameters might cause ambiguity in system performance when moving to the streaming settings.

- *Aim 3: demonstrate CBARS overall accuracy for recognising activities:* To be able to evaluate our technique with the two components of modelling and recognition, we test our technique on benchmarked AR datasets. We focus in the experiments on evaluating CBARS when applied across users. Thus, we compare CBARS accuracy with other classification techniques that are applied pervasively for activity recognition and discuss its performance.

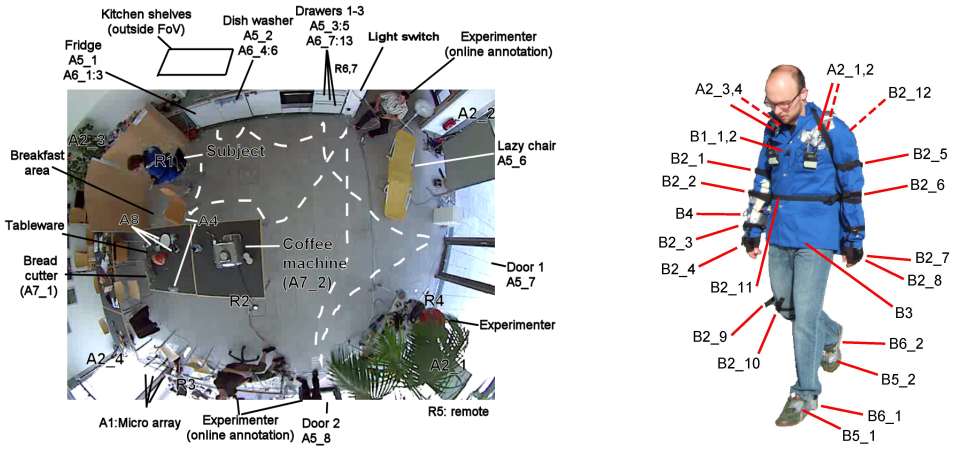
In this section, we start by presenting the datasets that are applied for the evaluation. Then, we explain the general setup and performance metrics. Finally, we demonstrate the main evaluation according to our aims from different performance perspectives.

3.7.1 Datasets

We conducted the evaluation on three real life activity recognition datasets. Details of the datasets are as follows:

- *OPPORTUNITY* [RFC+09]: This dataset provides realistic scenarios of activities with a variety of sensing modalities. The number of instances for different activities is uneven. The data is collected by four different users performing daily morning activities. The total amount of 72 sensors in 10 modalities are deployed for data collection. The sensors are located on objects in the environment and on subject's body. Environmental sensors are attached to fixed places in a studio flat with kitchen, deckchair, and outdoor access where subjects performed activities. The full setup of sensors is illustrated in Figure 3.8. Body worn sensors include 19 sensors distributed as: 5 IMU (inertial measurement units) sensors in the motion jacket, 12 bluetooth body worn accelerometers, and 2 inertial sensors placed on the feet. The collected data consists of annotated complex, interleaved, and hierarchical naturalistic activities, with a particularly large number of atomic activities (around 30,000). The dataset contains four types of atomic activities, namely *Sitting*, *Standing*, *Walking*, and *Lying*. For each subject, data is collected for 5 unsegmented sessions. Data for Subject 4 contains rational and additive noise to evaluate the robustness of AR techniques to noise. The annotation process is performed during collection; verified with video recording. The sampling rate for the OPPORTUNITY dataset is 30 Hz (approximately 1 instance every 33 millisecond). Table 3.1 [RCR+10] represents statistics on the four labels of the mode of locomotion in the OPPORTUNITY dataset.

Subset of the OPPORTUNITY dataset has been released for the opportunity challenge [CSC+13]. The main objective for this challenge is to establish publicly available benchmarked data for evaluating different methods of activity recognition. We perform our analysis on the same subset of data that is used for the challenge. This subset contains data collected by body worn sensors. From Table 3.1, we notice that the total time for the two activities of "Walking" and "Standing" is considerably longer than the one for the other two activities of "Sitting" and "Lying". However, the mean length for 'Sitting' and 'Lying' is longer that the one for "Walking" and "Standing". Thus, the overall occurrences of 'sit' and 'lie' activities are less frequent, yet the length/duration for each occurrence is long. On the other hand, "Walking" and "Standing" activities occur very frequently, with a short duration for each occurrence.



(a) Environmental sensors setup (b) Body worn sensors setup

Figure 3.8: Sensors Setup for the OPPORTUNITY Dataset [CSC+13]

Table 3.1: OPPORTUNITY Activity Labels (Locomotion)

Activity	Min Length	Max Length	Mean Length	Total Time (sec)
Walking	0.3	242.6	5.6	7900.5
Standing	0.2	171.2	7.5	7770.9
Lying	0.9	166.6	21.8	1219.5
Sitting	0.8	274.9	26.4	3349.5

- *Smart Phone Accelerometer Data (SPAD)* [DLL13]: In this dataset, we use the data collected from a mobile accelerometer sensor only. The data is manually annotated with four different activities of "Walk", "Stay still", "Run", and "drive". The sampling rate for this dataset is 5 Hz. The data for training and evaluating the system was collected from 8 users. Users had no restriction on how to carry the phone while collecting this data. Thus, the phone was carried in different ways such as in

tight/loose pants pocket, high/low jacket pocket, hand-held bag, and shoulder bag. When the data was collected, each user was encouraged to do different activities in different styles like walk with/without shoes, run slowly/quickly, run with high/low steps. The data is manually annotated as users were required to take note in their diaries for each performed activity.

- *Physical Activity Monitoring* [RS12b, RS12a] *(PAMAP2 dataset)*: The data is collected with 3 IMU body worn sensors. The positions of the sensors are over the wrist on the dominant arm, on the chest, and on the dominant side's ankle. The sampling rate for this dataset is 100 Hz. The dataset also contains heart rate sensor readings with a sampling rate of 9 Hz. The total of 9 subjects participated in the data collection (both females and males). Each of the subjects had to follow a protocol, containing 12 different activities. More detailed information about the PAMAP2 dataset can be found in Appendix A.

3.7.2 Experimental setup

The aim of the experiment section in this chapter is to evaluate both the BLFW and the ensemble classifier in CBARS. The evaluation for BLFW focuses on the two approaches that are proposed to build the baseline framework. The approaches are clusters with sub-clusters (SC) and clustering with dominant pattern selection (DPS). Therefore, we can conclude which approach is more efficient in modelling activities from historical data.

CBARS is a cluster-based technique that applies the ensemble classifier for recognition. The classification unit for the ensemble classifier is a cluster instead of an individual instance. Thus, unlabelled data instances are categorised into a set of clusters. Then the label of each cluster is predicted with the ensemble classifier. To measure the performance of the cluster-based technique, we assume that unlabelled data contains m different clusters $clus1$,$clus2$,... $clus_m$. S is the total number of unlabelled instances to be classified. The weight of clusters varies from one cluster to another. The weight is the ratio of the number of instances within this cluster (N_{clus_i}) to the total number of unlabelled instances (S) where i= 1,2,...m. The weight of any cluster i is defined as follows:

$$w_i = \frac{N_{Clus_i}}{S}$$

The true label of $Clus_i$, with weight w_i, is the majority label of instances within $Clus_i$. The cluster is classified with the ensemble classifier for either

one of three recognition outcomes: correct, incorrect or active. The class is correctly classified when the predicted label matches the majority label inside $Clus_i$. On the other hand, it is incorrectly classified when the predicted label is not the same as the true majority label of instances within the cluster. Lastly, the prediction is active when the ensemble classifier could not reach a decision about the label of the cluster.

A cluster-based technique is more suitable for handling activity recognition data. A single activity is represented by a sequence of instances. Thus, responding to each individual instance for activity recognition is inefficient and unnecessary. Our technique categorises data in clusters for batch processing for the entire cluster. Thus, the clustering technique accuracy to cluster similar data together affects the performance of the ensemble classifier. We define a new metric of cluster purity that represents the ratio of instances with the majority label within the cluster to the total number of instances in the cluster. Consider a cluster i with N_{clus_i} instances, 'A' is the major label of N_{maj} instances inside the cluster. The purity of $cluster_i$ is defined as follows:

$$CPur_{clus_i} = \frac{N_{maj}}{N_{clus_i}} \ \%$$

The purity of the cluster influences the decision of the ensemble classifier. The cluster that has a majority of label 'A' yet with low purity causes confusion to the decision of the classifier. In order to maintain a reasonable high purity among unlabelled clusters, we try to minimise the number of points assigned to each cluster (N_{clus_i}) by increasing the number of generated clusters (m). This way, we decrease the value of the dominator in the purity equation and thus expect the purity percentage to increase.

One advantage of CBARS is its flexibility in choosing the number of clusters. The choice has to keep the tradeoff between generating pure clusters, while maintaining the main characteristics of the cluster. In the following discussions and analysis, we will present an evaluation for the sensitivity of CBARS to the number of clusters. The analysis shows the tradeoff between the number of clusters, purity, and the classification decision.

Two sets of performance metrics are used to evaluate CBARS. The first one concerns the cluster-based performance metrics, while the other is for instance-based metrics. Cluster-based metrics focus on the evaluation of the classification of the clusters in the unlabelled data. The accuracy according to the cluster-based metrics is defined as follows: in S unlabelled data instances, CBARS uses the clustering technique to categorise unlabelled data into m clusters, each with weight w_i where i $\in$ 1,2,...m. Let $Correct_{CB}$ be the set of clusters that are correctly classified by length $Ncor_{CB}$. $Active_{CB}$ is the

set of $Nact_{CB}$ clusters that trigger active learning. $Incorrect_{CB}$ is the list of $Nmis_{CB}$ misclassified clusters. $m = Ncor_{CB} + Nact_{CB} + Nmis_{CB}$. Thus, the cluster-based metrics are defined as follows:

- TP_{CB} (%): Accuracy rate as the percentage of weighted clusters that are correctly classified = $\sum_{Ncor_{CB}} w_i$ where $i \in Correct_{CB}$
- ACT_{CB} (%): The percentage of weighted clusters that require the user's input via active learning= $\sum_{Nact_{CB}} w_i$ where $i \in Active_{CB}$
- ERR_{CB} (%): The rate of error as the percentage of weighted clusters that are misclassified = $\sum_{Nmis_{CB}} w_i$ where $i \in Incorrect_{CB}$
- CAA_{CB}: We also define a metric for combining both Correct And Active metrics (CAA) for cluster-based metrics. CAA_{CB} (%) = TP_{CB}+ACT_{CB}. This value measures the accuracy of the classifier as the percentage of weighted clusters with any classification decision except incorrect.

The cluster-based metrics measure the performance, based on the entire cluster classification decision. Depending on the purity of the cluster, there might be instances inside the cluster that have labels different from the majority label. Therefore, we introduce the instance-based metrics that calculate the performance metrics based on the instances instead of the clusters. The instance-based metrics checks the accuracy of each individual instance inside the cluster. Therefore, with impure clusters, the difference increases between cluster-based and instant-based accuracy. An instance with true label T is assigned to cluster i with majority label A. The ensemble classifier predicts the label of the entire cluster to be P. In cluster-based perspective, cluster i is correctly classified when $A = P$. However, with an instance-based perspective, an instance is correctly classified only when T= P regardless of A. To define the instance-based metrics, let $Ncor_{inst}$ be the number of instances that are correctly classified, $Nact_{inst}$ be the number of instances that trigger active learning, and $Nmis_{inst}$ be the number of instances that are misclassified. Where $S = Ncor_{inst}+Nact_{inst}+Nmis_{inst}$.We apply the following performance metrics to evaluate our technique with instance-based metrics.

- TP_{inst} (%): Accuracy rate as the percentage of instances correctly classified = $\dfrac{Ncor_{inst}}{S}$
- ACT_{inst} (%): The percentage of class instances requiring the user's input via active learning = $\dfrac{Nact_{inst}}{S}$

- ERR_{inst} (%): The rate of error as the percentage of misclassification instances $= \frac{Nmis_{inst}}{S}$
- CAA_{inst}: We also define a metric for combined both Correct And Active metrics (CAA) for instance-based. CAA_{inst} (%) $= \frac{Ncor_{inst} + Nact_{inst}}{S}$

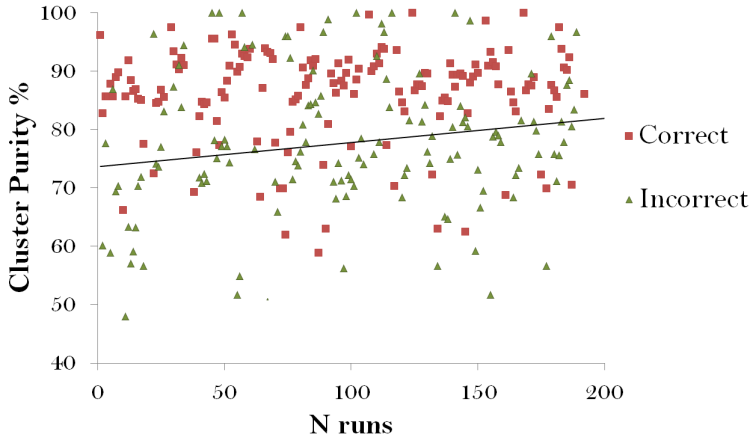
3.7.3 Analysis and discussion

In this section, we evaluate CBARS performance for both the BLFW and the ensemble classifier. We first discuss the clustering phase which is a middle phase in CBARS that is affected by many factors. Following on the discussion of cluster-based and instance-based metrics for evaluation, we start our analysis with further explanation of the impact of the cluster purity and the number of clusters on the recognition decision.

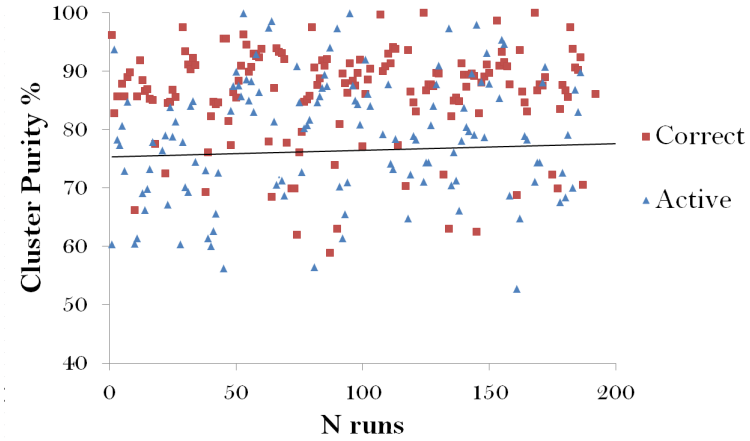
Cluster purity impact on the ensemble classifier

We start with illustrating the effect of cluster purity on CBARS accuracy. Three predicted labels are expected in CBARS. The cluster is either *correctly* classified, *incorrectly* classified, or requires *active* learning. Impure clusters that contain different combination of labels might cause confusion in the classifier decision. To visualise this effect, we run more than 150 experiments over the OPPORTUNITY dataset. The BLFW is built with data from one of the four subjects and tested with data from another subject. Subject data contains combined data across the 5 segmentations (ADL1-5). The total of 12 combinations are set for testing the OPPORTUNITY dataset. For each combination, we apply different clustering techniques, specifically EM, Kmeans, and DBScan, with a different number of clusters for each run to ensure the change in purity. The total number of runs for each combination is 15. Figure 3.9 describes the effect of cluster purity on classifier decisions. The incorrect and active decisions are clearly overlapping. Thus, for clarity, we display the results for the two pairs of decisions of correct with active, and correct with incorrect.

As shown in Figure 3.9, the clusters with purity of more than about 75% (above the line) are more likely to be correctly classified. As the purity decreases, the probability of either active learning or incorrect classification increases as a result of the confusion caused by the cluster impurity. With impure clusters, a decision is more likely to be incorrect or active. However, the probability of correct classification increases with pure clusters. The graph also shows that cluster purity is one of the factors that affect the classification



(a) Correct and Incorrect Recognition with Cluster Purity



(b) Correct and Active Recognition with Cluster Purity

Figure 3.9: Cluster Purity and Recognition Outcomes

outcome, but it is not the only factor. In some cases, clusters with high purity are actively or incorrectly classified. Thus, the purity can be considered as one of the factors that helps the classifier to reach a correct recognition decision. Other factors include the performance of similarity measures, noise in data, and the approach of building the BLFW. In order to increase the purity of the clusters, CBARS tends to produce more clusters, thus increasing the purity of each individual cluster. The following will discuss the impact of the number of clusters in test data on the classifier performance.

Ensemble classifier sensitivity to the number of clusters

CBARS is a cluster-based approach that applies the ensemble classifier to unlabelled clusters for recognition. CBARS does not require the actual number of activities/clusters in the incoming data as one of the parameters. CBARS

alternatively apply unsupervised learning for clustering incoming data into different clusters. It is possible that the number of clusters does not match the actual number of activities in the incoming data. Thus, CBARS has the flexibility to predict the label for each cluster separately. The relationship between incoming data clusters and BLFW clusters is many to one. That means, many incoming data clusters could be assigned to a single BLFW cluster (activity). One of the key advantages of CBARS is its flexibility in choosing m clusters in the incoming data. The number of clusters is either chosen by the clustering technique (such as EM), or pre-specified as a fixed number. Both the cluster purity and recognition accuracy are impacted with the number of generated clusters. The interrelation between the number of clusters, cluster purity, and recognition accuracy is shown in Figure 3.10. The previous discussion of Figure 3.9 shows the purity impact on the classification decision. In this discussion, we show the impact of the number of clusters on the recognition accuracy and purity.

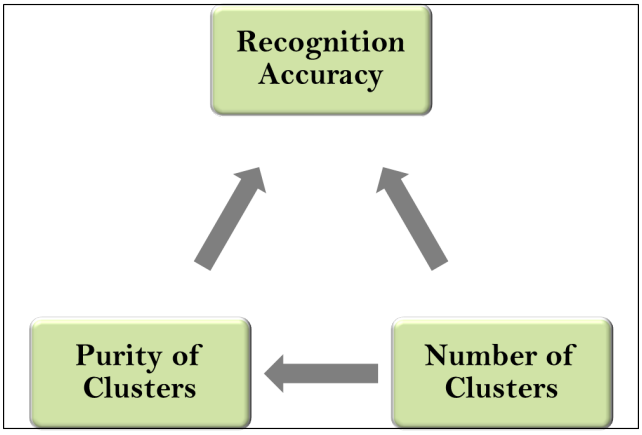
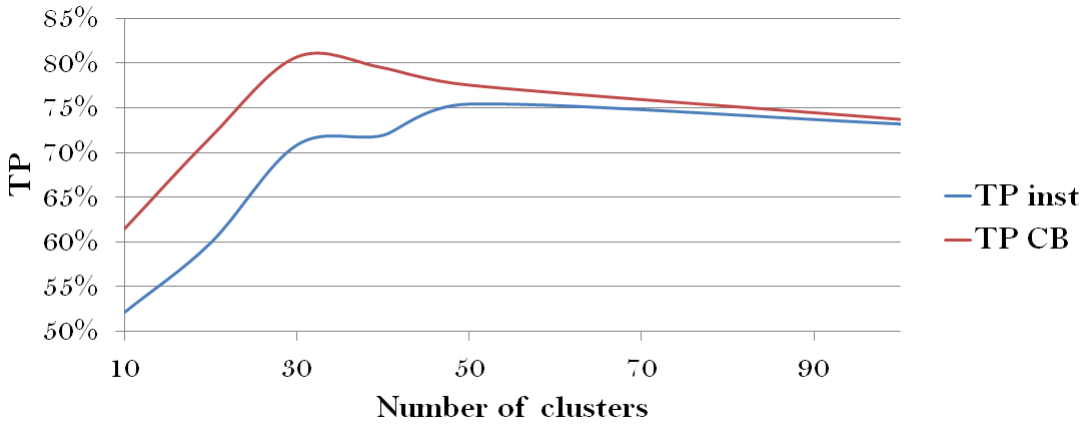


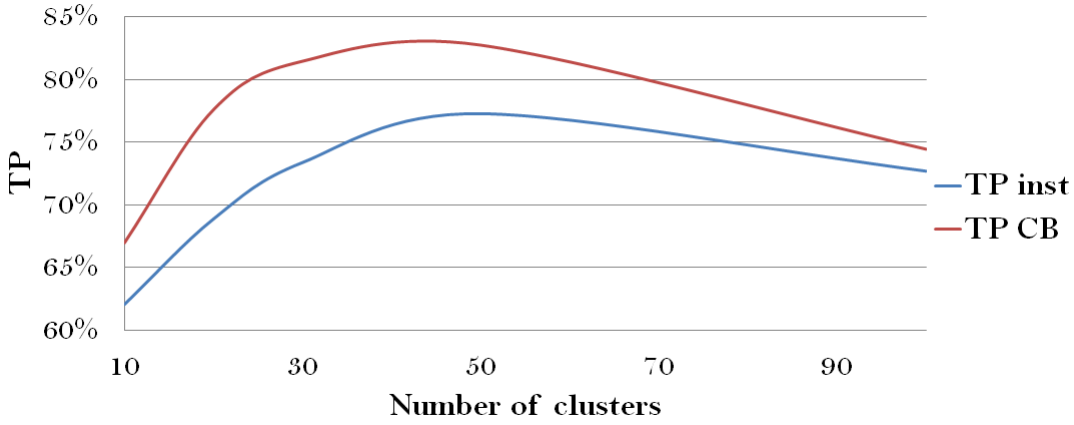
Figure 3.10: Relationship between Cluster Parameters and Accuracy

Figure 3.11 shows the recognition accuracy in terms of TP_{inst} and TP_{CB} with different numbers of clusters in two different subsets of the PAMAP2 dataset. For each subset, the BLFW is built form one user, while the test data is from another user. The graphs show an exponential increase in the accuracy where $m \leq 40$. TP_{inst} increased from 52% to 75% in Figure 3.11 (a), and from 62% to 73% in Figure 3.11 (b). The accuracy starts to stabilise as the number of clusters increases. Then, it slightly decreases as a result of the large number of clusters that might cause a distortion of the cluster representation, due to its small size. The purity of clusters is indicated by the difference between the two metrics (TP_{inst} and TP_{CB}). Both graphs show a gap between the two lines. The correctly classified instances within the cluster represented by TP_{inst} is much less than the cluster-based accuracy for TP_{CB}. Therefore the

gap reflects the impurity of the clusters that are correctly classified. The gap starts to decrease as the number of clusters increases to show an improved purity of the clusters. However, clustering data into too many clusters affects the representation of the clusters. That means, the characteristics of the cluster that are used for recognition will be distorted due to the cluster's small size - the cluster's features are not sufficiently represented. Therefore, the tradeoff between the purity of the clusters and an accurate representation of the clusters is necessary when choosing the number of clusters in unlabelled data.



(a)



(b)

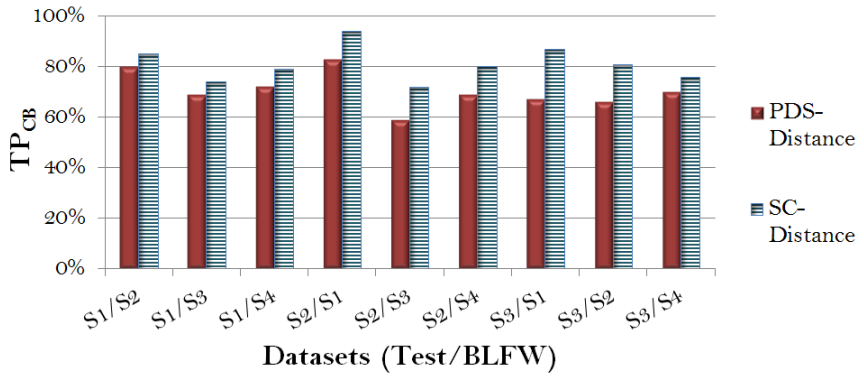
Figure 3.11: The Impact of Number of Clusters on the Ensemble Classifier Performance with Subsets of the PAMAP2 Dataset

After discussing the impact of different parameters on recognition accuracy, we discuss in the following the two approaches applied in the modelling component of CBARS (*Aim 1*).

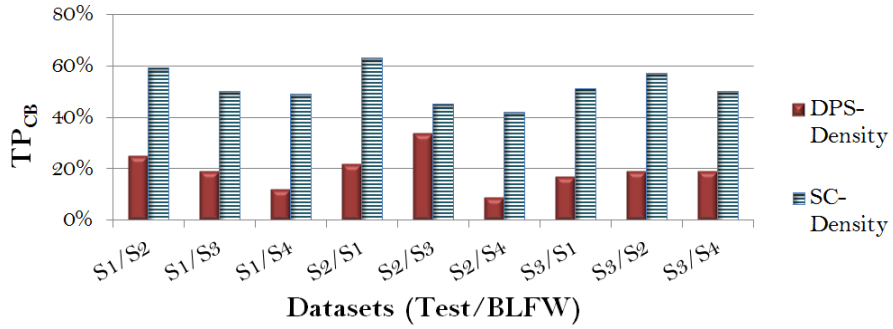
BLFW approaches

The first component in CBARS, as well as other techniques that are developed in this dissertation, is the modelling component where we build the BLFW. There are two approaches of building BLFW; clustering with dominant pattern selection (DPS) and clustering with sub-clusters (SC). Each approach has its own perspective of modelling labelled data in the training/offline phase. The DPS approach focuses on representing the classes with its dominant pattern. On the other hand, the SC approach builds a fine-grained model that contains a different set of patterns inside each cluster. We expect these approaches to affect the overall performance of CBARS as well as the performance of each measure in the ensemble classifier. The evaluation also considers the robustness of each approach in the presence of existing noise. We conduct the evaluation of the BLFW using the OPPORTUNITY dataset. The dataset contains four atomic and broad activities representing the mode of locomotion. The broad activities ensure the existence of patterns within activities that help evaluate the approach's fine-granularity. Also, OPPORTUNITY is a high dimensional dataset with a subset of the data that contains noise. For all experiments, we build the BLFW with data from one user and evaluate the system on data from other users. The first part of the experiments evaluates the impact of approaches in building the BLFW on each measure and also on the overall performance of CBARS.

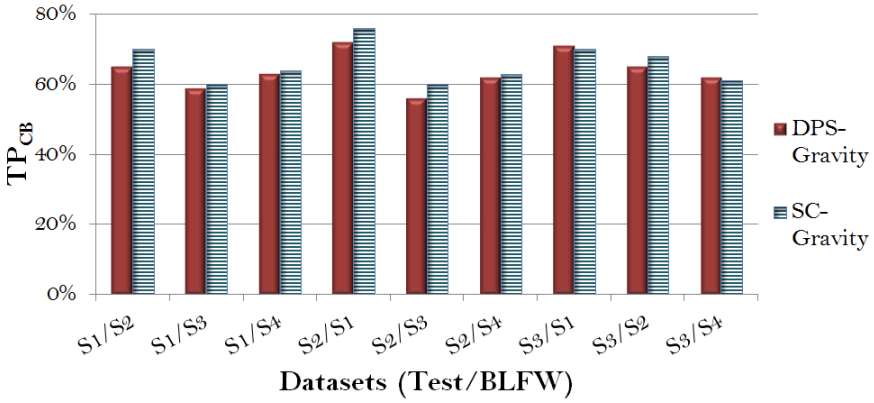
Figure 3.12 shows the change of the performance of each measure with the two approaches. We use the same settings for each experiment to be able to compare the two approaches. The displayed percentage is the accuracy rate (TP_{CB}) for each measure. Figure 3.12(a) and (c) shows a comparable performance for both measures of distance and gravity. However, with density and deviation, Figure 3.12(b) and (d), the accuracy is affected negatively with the DPS approach. The calculations of both distance and gravity measures rely mainly on the cluster centre, which is almost the same with both DPS and SC. This explains the similar performance for these measures with both approaches. However, the other two measures, density and deviation, are based on the distribution of data both in a holistic way for all existing clusters (density) or within each cluster (deviation). Presenting only the dominant pattern with the DPS approach limits the ability of both measures to recognise the distribution of data to only one major pattern. On the other hand, the SC approach is rich with different patterns and thus represents a more comprehensive image of data distribution.



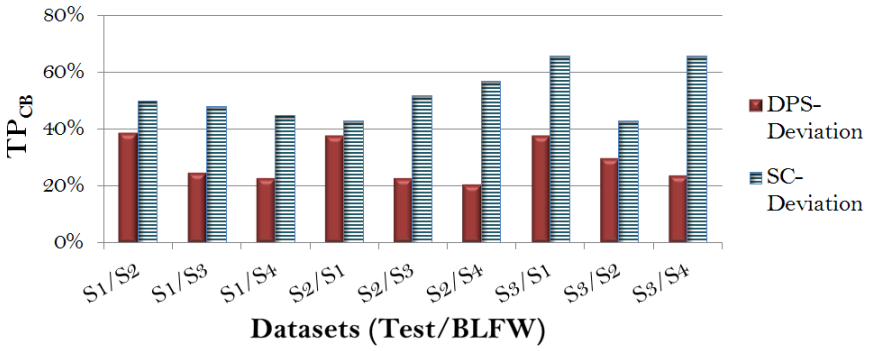
(a) Distance Performance with BLFW Approaches



(b) Density Performance with BLFW Approaches



(c) Gravity Performance with BLFW Approaches



(d) Deviation Performance with BLFW Approaches

Figure 3.12: Measures Performance with the BLFW Approaches

The performance of the similarity measures with the two approaches would affect mainly the rate of active learning. The decision of the ensemble classifier is based on the majority vote among measures. The negative impact of the DPS approach on density and deviation results in domination of the other two measures. Therefore, the likelihood of confusion among measures is decreased, which is the motive behind active learning. Figure 3.13 shows the percentage of active learning for the same combinations of data and rate of active learning ACT_{inst} with the two approaches. The data used for building the BLFW in these experiments exclude S4, which include noise, for a separate evaluation for the robustness of the BLFW to noise effect. In most of the runs, SC approach triggers a higher percentage of active learning.

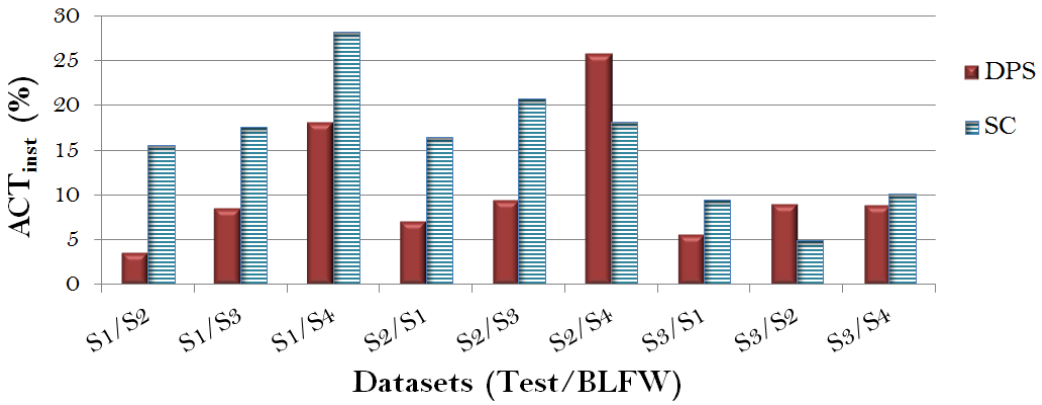


Figure 3.13: The Impact of BLFW Approaches on Active Learning

The higher active learning rate could be considered as a drawback rather than an advantage. This is valid only in the case that the error rate ERR_{inst} of corresponding experiments with SC is higher than the one for DPS. In Figure 3.14, the error rate for SC approach is below DPS approach for most runs. Thus, active learning privileges the combination of measures to reduce the error rather than affecting the accuracy.

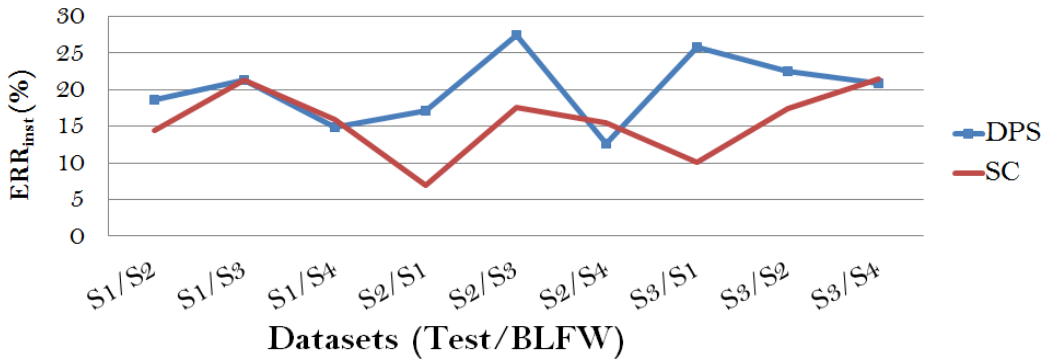


Figure 3.14: The Impact of BLFW Approaches on ERR Rate

BLFW robustness in dealing with noisy data is another important criterion in evaluating the two approaches. Table 3.2 shows the instance-based accuracy across users using the OPPORTUNITY dataset. Each row corresponds to the average performance metrics across users. This evaluation focuses on deploying data with noise for building the BLFW. The effect of noise on the ensemble classifier is discussed later in this section. Therefore, when we build the model with data containing no noise in (S1, S2, S3), the average metrics are calculated on all other users except S4. The first three rows in the table show the overall performance of CBARS with the two approaches. Although active learning is higher when using the SC approach, the error rate is lower than using the DPS approach. When we build the BLFW with data from S4 that contains noise, we found that the DPS approach attains better accuracy with a lower error rate than SC. These results are interpreted as follows. The DPS approach for building the BLFW selects only the dominant pattern, therefore all noise and outliers are automatically eliminated and thus the model is constructed with pure data that contains no noise. On the other hand, the SC approach deals with the noise as part of the data. Therefore, noisy data might be distributed among sub-clusters or separated in its own sub-clusters. The noise is not eliminated but instead reduced in this approach. This also helps with limiting the noise effect however without complete elimination as in the DPS approach. Although DPS outperforms SC in the case of noisy data, the small difference in performance could be tolerated by considering the big difference in performance between the two approaches with normal data in terms of error rate. As shown in Table 3.2, the error rate in SC, for S2 and S3, is about 10% less than DPS. On the other hand, the difference in error rate with noisy data is only around 4% less for DPS.

Table 3.2: BLFW Approaches and Robustness to Noise

BLFW data	Approach	TP_{inst}	ACT_{inst}	ERR_{inst}
S1	DPS(%)	74.0	6.0	20.0
	SC(%)	65.6	16.5	**17.9**
S2	DPS (%)	69.4	8.0	22.6
	SC(%)	69.2	18.5	**12.2**
S3	DPS(%)	68.6	7.3	24.2
	SC(%)	79.1	7.1	**13.8**
S4	DPS(%)	73.1	13.9	**13.0**
	SC(%)	69.5	13.4	17.1

Based on the above discussion, the gain from using the SC approach for building the BLFW outweighs the one from the DPS approach. Despite the achieved improvement of DPS over SC for noisy data, the negative impact of DPS on measures, error rate, and active learning rate as well as the minority of improvement with noisy data supports our preference to SC approach. After analysing the modelling component and its approaches, we present in next discussion an evaluation of the hybrid similarity measure approach, applied for the ensemble classifier (*Aim 2*).

Measures combination

The ensemble classifier in CBARS deploys a hybrid similarity measure approach for recognition. It combines four different measures that bring together different perspectives of the data in order to comprehensively predict the cluster label. Each measure has its own accuracy rate that might differ from some datasets to another. For example, distance measures might perform well with some datasets, whereas the density measure for instance could perform better with other datasets that rely mainly on data distribution. Also, in recognising activities, some of the activity classes are small and to be found in the middle of other big classes, such as the "Drive" class that in turn might appear in the middle of the "Sit" class. If we apply only the distance measure, the classifier will mostly fail to recognise the "Drive" activity. However, considering the gravity and the distribution of the data, with both the density and deviation measures, would enhance the recognition accuracy. Thus, combining measures is expected to enhance the overall accuracy of the classifier and outperforms any single measure.

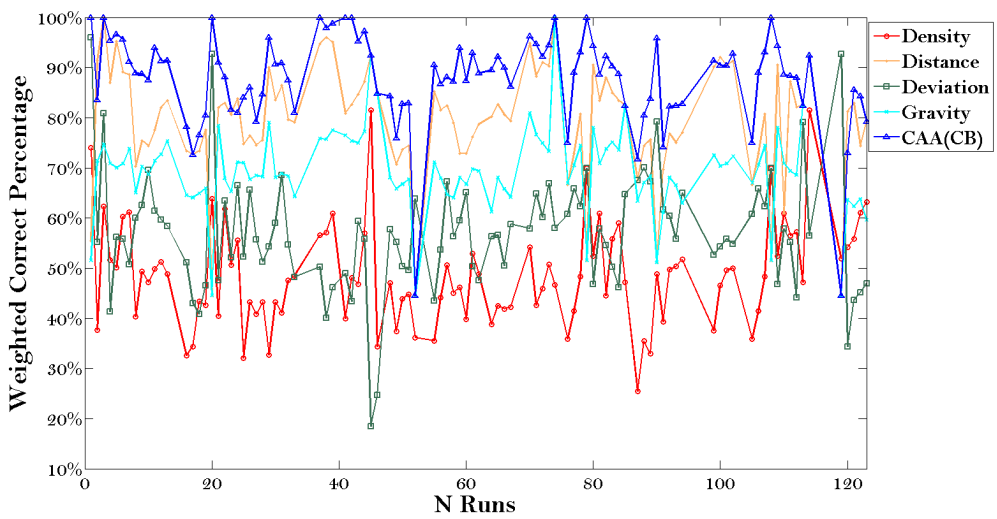


Figure 3.15: Performance of the Four Similarity Measures

Figure 3.15 shows the performance of each measure individually as well as the overall performance. The accuracy of each measure is calculated as a cluster-based accuracy rate (TP_{CB}). The overall performance of combining all measures is calculated by considering both accuracy rate and active learning rate. Active learning is an outcome of the confusion among measures, therefore we could not overlook its percentage when presenting the performance of the measures. As shown in the graph CAA_{CB} always outperforms any single measure in each run and across all runs. In numbers, the average TP_{CB} for distance across more than 120 experiments is $\simeq$ 80 %, density 45%, deviation 60%, and gravity 75%. The average overall accuracy for CAA_{CB} is $\simeq$ 90, with $TP_{CB} \simeq 70$ for combining all measures.

From the aforementioned results, we can conclude that combining various measures always outperforms applying an individual measure for recognising activities. Analysing the measure combination performance leads us to the discussion of the overall accuracy of CBARS on benchmarked datasets (*Aim 3*).

CBARS and other classification techniques

We compare CBARS classification accuracy with other state of the art classification techniques that are pervasively applied for activity recognition. The three classifiers are Decision tree (C4.5) [Qui93], SVM [CV95], and Naive Bayes [JL95]. The settings are the same for all experiments: we build the BLFW from one of the subjects and classify data from other subjects. Table 3.3 shows the performance of CBARS compared to the three classification methods used on the OPPORTUNITY dataset. Based on the aforementioned discussions, we build the BLFW in the following experiments with the SC approach. We show both instance-based metrics and cluster-based metrics for representing CBARS performance. Instance-based metrics are important for the same ground comparison with other *non* cluster-based classification techniques. The cluster-based accuracy shows the percentage of weighted clusters that have been correctly classified. When the difference between cluster-based metrics and their corresponding instance-based ones is minimised, that reflects the purity of the clusters and thus the efficiency of the cluster-based approach for recognition. The number of clusters in all experiments is set to 100.

Both TP and CAA values for cluster-based and instance-based metrics are reported in Table 3.3. Each row displays the overall average accuracy for testing the data from an individual subject, when the BLFW is built from other subjects' data. We always use data from different users for training and testing to show the performance of CBARS across users. For example, the

first row represents the average performance metrics for the pairs of (S1, S2), (S1,S3) and (S1,S4) for testing and building the BLFW respectively. As shown in Table 3.3, CAA_{inst} always outperforms the other classification methods. If we exclude the active learning percentage, TP_{inst} still outperforms other techniques except for Subject 3 when CBARS comes second. Results for Subject 4, in the last row, show the efficiency of CBARS and its ensemble classifier in handling data that contains noise. The best TP rate for all other classifiers is below 55%, however TP_{inst} with CBARS is 63.6% and CAA_{inst} is 82.3%. CAA_{inst} shows also the high rate of active learning with noisy data (in S4). In all experiments, when we compare cluster-based metrics with instance-based ones, the difference between CAA_{ins} and CAA_{CB} is $\simeq$ 6.6, 10, 2.6 and 0.8 (%) for S1, S2, S3 and S4 respectively. We can conclude from these numbers that in most cases the clusters could represent activities with a reasonable purity level within the clusters.

Table 3.3: CBARS Performance Compared to Other Classification Techniques using the OPPORTUNITY Dataset

Dataset	DT	NBayes	SVM	CBARS			
				TP_{inst}	TP_{CB}	CAA_{inst}	CAA_{CB}
S1(%)	63.0	67.3	76.8	**81.3**	87.9	89.9	96.5
S2(%)	54.5	63.0	71.8	**72.8**	86.7	83.7	93.7
S3(%)	56.8	62.4	**64.7**	62.8	65.3	80.2	82.8
S4(%)	43.4	31.9	52.9	**63.6**	66.2	82.3	83.2

We also evaluate CBARS performance on PAMAP2 dataset. The dataset is rich in terms of the number of activities as shown in Appendix A. CBARS analyses activity recognition data as clusters with sub-clusters that correspond to patterns within activities. Therefore, we apply a preprocessing step on the PAMAP2 dataset to aggregate data that represents different patterns of a single activity together. Specifically, we add the instances representing the activity of "Nordic walking" to the broader class of "Walking". We also combine "Ascending stairs" and "Descending stairs" in a single "Stairs" class. Thus, we choose users that have the greatest number of common activities. The number of clusters in test data for all experiments is set to 50. Figure 3.16 shows the accuracy of CBARS compared to other well-known classification techniques. The horizontal axis represents the pair of datasets applied for testing and building the BLFW respectively.While the vertical axis displays the percentage of the number of instances that are correctly classified for each technique. The figure shows a better performance for CBARS across different experiments. The

accuracy for CBARS ranges between 60% to 80% for all runs. The average active learning rate (ACT_inst) is 5.11%. The experiments show that, CBARS accuracy (even with excluding the active learning percentage) is higher than the performance of all other classification techniques.

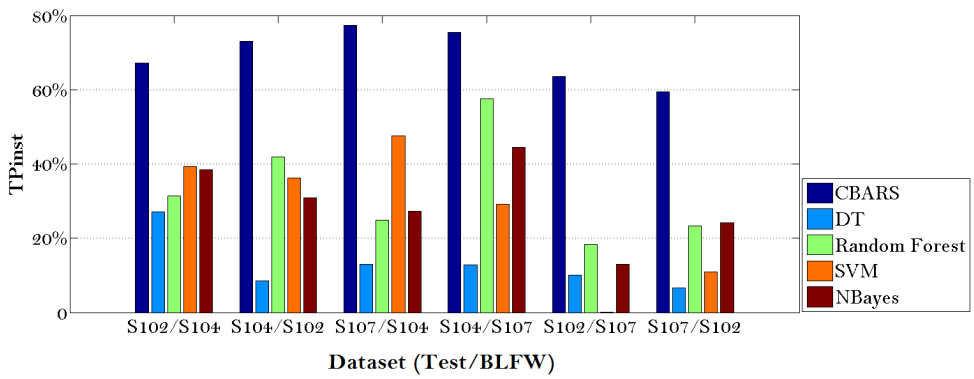


Figure 3.16: CBARS Performance Compared to other Classification Techniques using the PAMAP2 Dataset

CBARS is also evaluated with the SPAD dataset. We apply 10 folds cross validation for comparing CBARS with other classification methods. Table 3.4 shows the average performance across the 10 folds for each method. The recognition accuracy is generally good for this dataset across most techniques. The decision tree attains an accuracy of 95.4%. CBARS achieves almost the same accuracy, 95.6%, for instance-based metrics that exclude the percentage of active learning. The instance-based accuracy for CBARS including the active learning percentage is even better with 97.2%. The cluster base accuracy including the active learning percentage achieves the best performance with 99.0%.

Table 3.4: CBARS Performance Compared to other Classification Techniques using the SPAD Dataset

DT	NBayes	SVM	CBARS			
			TP_{inst}	TP_{CB}	CAA_{inst}	CAA_{CB}
95.4%	89.9%	68.3%	95.6%	97.4%	97.2%	99.0 %

In conclusion, the analysis of CBARS performance on the benchmarked datasets shows an improved accuracy of our technique compared to other state-of-the-art techniques applied for activity recognition. CBARS shows its best performance when applied on data containing noise and across users. This demonstrates the robustness and accuracy of both the BLFW and the ensemble classifier.

Table 3.5: OPPORTUNITY Challenge Datasets

User	ADL1	ADL2	ADL3	ADL4	ADL5
Subject 1	•	•	•	•	•
Subject 2	•	•	•		
Subject 3	•	•	•		
Subject 4	•	•	•		

Part of our evaluation of CBARS is to compare its performance to other benchmarked techniques in activity recognition. The OPPORTUNITY challenge announced in 2011 was mainly for evaluating the performance of different AR techniques on the OPPORTUNITY dataset. In the following, we use the same settings of the challenge to test CBARS performance and compare it to the performance of the winning technique that attains the best performance in the competition.

3.7.4 OPPORTUNITY challenge

The main motivation for the challenge is to provide a benchmarked database for activity recognition. Four tasks were announced in May 2011 for participation. The challenge evaluated and compared the results for different classification methods contributed by different research groups. The data released for the challenge are as in Table 3.5. The focus of this evaluation will be for the multimodel classification task for locomotion (Task A). The task A goal is to classify the subject mode of locomotion (i.e. "Sitting", "Standing", "Walking", "Lying") for the last two sessions in Subjects 2 and 3. In the evaluation of our technique we followed the same settings of the challenge. We participated in this challenge and our entry won the first place in Task A, as we achieved the best performance for this task by applying the DT grafting [Web99] classification method on sensor values [CSC+13]. At that time, CBARS was not completely ready to be tested for the challenge. In this section we show the performance of CBARS on the same challenge settings and compare it with the best results obtained in the challenge. A preprocessing step has been added to select set of attributes from the datasets. We apply the Weka [WF05] CfsSubsetEval [Hal99] method for attribute selection on subsets of data that are labelled from S2 and S3. It selects only 44 attributes based on the evaluation of the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. The displayed results show the accuracy associated with excluding the instances for null (transition) class.

Active learning is part of CBARS, yet it is not considered in the DT grafting method. Thus, we display the performance metrics for CBARS with and without active learning. To be able to show results with no active learning, we use unbalanced voting between measures in the case of active learning. When confusion occurs among similarity measures, the decision is biased towards the distance measure. The distance measure is chosen because it shows the best performance among all measures in the aforementioned discussion on measure combination. In Table 3.6, CBARS-biased shows the performance metrics when more weight in voting is given to the distance measure in case of confusion only. CBARS-active displays the performance metrics for CBARS including active learning with balanced voting among measures. The numbers show very similar results between TP_{inst} in CBARS and DTgraf. The accuracy is higher for S2 with about 4% when we include active learning; and slightly lower for S3 with about 1%. From the aforementioned results, we conclude that CBARS performance is up to the best results achieved in the challenge. The accuracy is even higher when we include active learning in the performance. Although CBARS' best performance was attained when applied across users, CBARS still achieves high accuracy with data from the same user.

Table 3.6: CBARS Performance on the Challenge Datasets (Task A)

Dataset	DTgraf TP		CBARS-biased TP_{inst}		CBARS-active TP_{inst}		CBARS-active CAA_{inst}	
S2(ADL4) (%)	87.9	86.3	87.6	**86.7**	84.8	83.0	91.8	**91.0**
S2(ADL5)(%)	84.6		85.7		81.1		90.2	
S3(ADL4)(%)	87.7	**87.5**	87.1	86.7	81.4	81.0	89.2	86.3
S3(ADL5)(%)	87.3		86.3		80.7		83.4	

3.8 Summary

In order to build efficient activity recognition systems for real time applications, researchers need to develop techniques for activity recognition that are robust, accurate, flexible, and efficient. Typical activity recognition techniques build a learning model from historical data that is available for training. The offline built model is static and not extendible for accommodating expected and natural change when the model is deployed for recognition. Our first objective in this thesis is to build a flexible learning model that enables personalisation and adaptation beyond the offline training phase, through incremental and continuous learning. In this chapter, we present our learning model for activity recognition, termed as the baseline framework (BLFW). The BLFW

has to essentially satisfy a set of desired features that includes flexibility to be extended, robustness to noise, accurate representation of historical data and computationally efficient. We discuss the two approaches for building the BLFW along with its characteristics.

Then, we present our first technique for activity recognition that consists of the modelling component in addition to the recognition component. The developed technique, termed CBARS, applies an ensemble classifier based on a hybrid similarity measure approach in the recognition component. Combining various similarity measures enables a more comprehensive view of the data from various perspectives and thus enhances the recognition accuracy especially across different users. The four deployed measures, namely distance, density, deviation and gravity, concern both closeness and cohesiveness of data. We discuss in this chapter the details of each measure as well as the methodology of applying the ensemble classifier with the hybrid similarity measure approach for activity recognition. CBARS also incorporates the concept of active learning that is part of the recognition process with no learning overhead.

The evaluation of CBARS includes analysing both the modelling component and the recognition component. We discuss the robustness, accuracy, and efficiency of the developed approaches for the BLFW on benchmarked datasets for activity recognition. From the results, we concluded that the clustering with sub-clusters approach for building the BLFW provides better performance than the dominant pattern selection approach. We also discussed the performance of CBARS recognition component and its hybrid similarity measure approach. The combination of all measures showed better performance than any individual one. The overall accuracy of CBARS has been assessed and compared to other state-of-the-art activity recognition techniques. The results demonstrated better performance of CBARS in recognising activities, especially when applied across various users and in datasets that represent large numbers of activities. CBARS achieves its best accuracy enhancement, around 10%, when applied on datasets contain noise and across users.

Although CBARS does not implement the actual personalisation and adaptation for activity recognition, it presents an efficient baseline framework and recognition technique for activity recognition especially across users. In the next chapter, we build on and extend CBARS by enabling personalisation through incremental learning for continuous refinement with the evolving activities in a streaming environment.

Chapter 4

Personalised Activity Recognition Technique with Evolving Data Streams

4.1 Introduction

In Chapter 3, we proposed, developed, and evaluated our first technique for activity recognition, CBARS. The technique integrates the baseline framework (BLFW) that is built from historical data with an ensemble classifier based on a hybrid similarity measures approach. The experiments showed the efficiency of the developed technique especially across different users. However, CBARS itself does not implement personalisation or adaptation of the learning model with evolving activities. There is no one model that could fit all different patterns of activities. As people perform activities differently, model personalisation is essential to tune the model for the user's personalised way of performing activities. In this chapter, we extend CBARS by enabling personalisation through incremental learning to cope with changes in activity data. Thus, we propose, develop, and evaluate a personalisation technique for activity recognition coined STAR [AGSK12b, AGSK15], which stands for **ST**ream learning for **A**ctivity **R**ecognition.

Recognising activities in real time requires learning and analysing sensory data that evolves from different data sources. That essentially emerges the need for handling the streaming nature of sensory data with its corresponding analysis and processing challenges. Data stream mining has unique characteristics that make it more challenging than static data mining. In a streaming environment, multi-dimensional data arrives at high speed with infinite length and requires real or near real time processing. Traditional mining techniques

that require several passes on data cannot be applied in a streaming environment. Moreover, the *concept drift* nature of data streams makes it hard to predict and classify new incoming data. Prior knowledge of data contains concept drift eventually becomes outdated while the stream evolves. Deployment on devices with limited resources is another challenge that needs to be addressed to ensure efficiency of the recognition technique with data streams.

This chapter discusses first the desired functionalities of STAR. Presenting a detailed illustration of STAR with its phases and components will follow this. Afterwards, we discuss how STAR addresses challenges in AR along with its contribution. Lastly, experimental studies and evaluation of STAR are extensively discussed. The chapter is concluded with a summary.

4.2 Desired Features and Challenges for STAR

STAR, our personalised activity recognition technique, operates in streaming environment. STAR has to satisfy a set of desirable features for achieving efficient recognition. These features include *accuracy*, *robustness*, *flexibility*, and *computational efficiency*. The deployment of STAR in a streaming environment imposes additional challenges and constraints to the context of activity recognition. STAR has essentially to address these challenges in addition to activity recognition challenges, to achieve the aforementioned desired features. Illustrations of challenges that are addressed by STAR are as follows:

- *Degradation of the learning model:* The learning model that is built from historical data becomes inaccurate over the time. Changes in both environmental settings and activity patterns from one user to another are realistic reasons for model degradation over time. Activity recognition typically deals with streaming data that evolves from sensors. One of the most important features of real-world data streams is concept drift. It refers to the change of stream characteristics while time evolves. The presence of concept drift in a data stream renders the traditional approaches for activity recognition unsuitable, and therefore new approaches must be developed to accommodate streaming activity recognition data with concept drift. Drifting of data streams in the context of activity recognition reflects changes of activities. Therefore, tuning the model to accommodate these changes is essential to enhance the recognition *accuracy* and maintain its *flexibility* and *robustness*.

- *Scarcity of labelled data in streaming settings:* It is unrealistic to assume that "labelled" data in data streams is available and accessible at all

times. Practically, sensory streaming data arrives at high speed and is rarely labelled. In order to continuously learn from data streams, true labels are essential to maintain a ground truth. Thus, it becomes essential to develop *computationally efficient* techniques that improve the data labelling process at real time with low cost, in order to boost the recognition *accuracy*.

- *Real time constraint:* Many applications in activity recognition are required to respond in real time. Detecting the sudden fall of an elderly person is a clear example that requires an immediate response to avoid risky situations. Another example is a targeted advertising domain when personalised advertisements or discount deals are provided to a user in smart shopping scenarios. Real time constraints require the recognition technique to be *computationally efficient* in order to process, analyse, and recognise activities in real time.

- *Limited resources of the deployment platform:* Efficient and opportunistic activity recognition emerged the need of deployment on portable devices such as smart phone or sensor device for real time processing and recognition. State-of-the-art techniques in activity recognition infer current activities by leveraging the rich sensory data that is available on sensory device. For instance, today's smart phones not only serve as the key computing and communication mobile devices of choice, but they also come with a rich set of embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone, and camera. The merit of deployment on portable device platform is that we do not need to deploy additional devices. Moreover, the system is simple and easy to use. Since people carry their personal companion devices all the time and have the full control of their own devices, those devices will not feel intrusive to the users. Furthermore, these devices are becoming increasingly intelligent and powerful. Thus, they appear to be the ideal platforms for detecting people's activities from data streams. However, deployment on such devices imposes other constraints related to the limitation of resources. Thus, the developed technique has to be *computationally efficient* in order to deal with the limited resources of the deployment device.

The aforementioned challenges are addressed by STAR for achieving the desired features of *accuracy*,*robustness*, *flexibility*, and *computational efficiency*. In the following sections, we explain details of the conceptual framework and

methodology of STAR. We start with showing the differences and extensions of STAR from CBARS.

4.3 STAR extensions from CBARS

Before introducing STAR in detail, we first explain the key differences in terms of approaches and goals between CBARS and STAR. Basically, STAR is a step forward that extends CBARS in two key directions. First, STAR implements personalisation of the learning model through incremental learning. Although, CBARS enables the personalisation process with a flexible learning model, CBARS itself does not implement the personalisation process. STAR extends the two components of CBARS, modelling and recognition, with a third component for personalisation as shown in Figure 4.1.

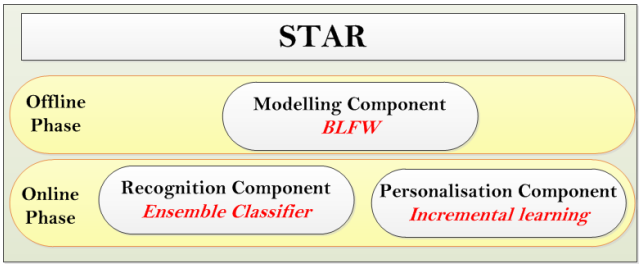


Figure 4.1: Components of STAR

STAR also extends CBARS with new techniques for handling the streaming nature of data representing evolving activities. In pursuit of this goal, it applies a continuous sliding window to evolving data streams. Unlike CBARS, STAR separates the learning process into an offline phase and an online phase. In the offline phase, the initial learning model is built from historical data. Both recognition and personalisation in STAR occur in the online phase.

In addition to these two main extensions, STAR addresses the need to cope with the limited labelled data through active learning that is incorporates with incremental learning in the personalisation component. Moreover, STAR facilitates processing of an AR sequential stream by handling of the occurrence of concurrent activities, i.e. more than one activity in a single time window. Lastly, STAR deploys activity recognition on a mobile platform to demonstrate its efficiency on a portable device with limited resources.

After illustrating how fundamentally STAR extends CBARS, we start in the next section describing the details of the STAR technique. We begin the problem formalisation in the next section.

4.4 Problem Formalisation

We first provide a definition and formalisation of data streams for activity recognition. A data stream can be defined in many ways depending on the entities of the stream. The first definition of a stream is based on the smallest entity of a data instance. Basically, a data stream is a continuous and unlimited flow of data instances that is represented as follows:

Definition 4.1 $Stream = D_{t_i}, D_{t_i+1}, D_{t_i+2}, ...D_{t_i+s}$ *and* $D_{t_i} = x_1, x_2, ...x_n$

Where D_{t_i} is an instance of the stream that arrives at time t_i. $x_1, x_2, ..x_n$ are features that represent D_{t_i} in an n-dimensional space. The actual class label for a data instance (D_{t_i}) at time t_i is y_{t_i}. The label of any data instance (D_{t_i}) is unknown. Thus our aim is to predict it.

The other definition of data stream is based on the occurrences of activities. A data stream consists of a set of consecutive and interleaved activities. Each activity occurs in groups of consecutive data instances as follows: $actOcc_{t_i} = D_{t_i}, D_{t_i+1}....D_{t_i+dur_i}$. $actOcc_{t_i}$ starts at time t_i for a duration of dur_i. We assume in this representation non-overlapping between activities. Yet, the presentation allows more than one activity to appear in a single chunk. Therefore, a data stream could be represented as a sequence of activity occurrences as follows:

Definition 4.2 $Stream = actOcc_{t_i...t_i+dur_i}, actOcc_{t_j...t_j+dur_j}, ...actOcc_{t_m...t_m+dur_m}$

Where $t_j = t_i + dur_i$, that means activities are performed sequentially; new activity instances follow the previous one. In STAR, we apply a sliding window to handle the consecutive activities in a data stream.

The stream could also be defined in terms of windowing data. As STAR applies a sliding window, each window captures a chunk of data ($Chunk_c$). A data chunk contains a set of data instances that exist in a particular sliding window as follows: $Chunk_{t_i} = D_{t_i}, D_{t_i+1}...D_{t_i+ws}$ where ws is the window size. Thus, a data stream could be defined according to windowing as follows:

Definition 4.3 $Stream = Chunk_{t_i...t_i+ws}, Chunk_{t_k...t_k+ws}...Chunk_{t_y...t_y+ws}$

Where $t_k = t_i + ws$. To be able to integrate the three definitions in a single problem formalisation, we need to explain the connection between entities in each. The small entity in all definitions is the data instance D_{t_i}; both activity occurrences and data chunks consist of groups of data instances. In a stream of activities, a single data chunk may contain more than one activity occurrence.

This is because of the existence of interleaved and interrelated activities in a data stream. For instance, in a window of few seconds, user might perform a "walking" activity with pauses of "standing". Also, the transition between two activities might occur in the middle of the window even if the window size is small. Therefore, each data chunk ($Chunk_{ti}$) might contain a single or multiple activity occurrences $actOcc_{t_i}$. At the same time, the occurrence of an activity $actOcc_{t_h}$ might elapse one or more data chunks depending on the duration of the activity occurrence and the window size. Figure 4.2 illustrates the integration among the three definitions of the data stream.

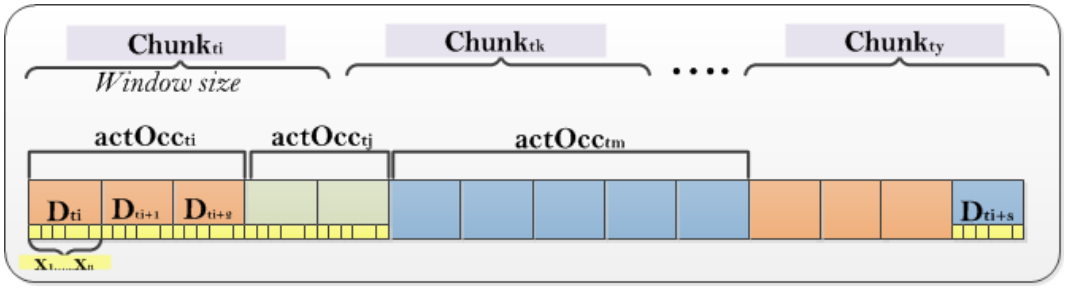


Figure 4.2: Explanation of the Data Stream According to Different Definitions

We deal in STAR with unsegmented data stream where there are no boundaries between activities. We scan the stream of data with a sliding window. The data chunk is the input unit for analysing and processing in STAR. The output of our technique is comprised of the labels of activities that occur in each data chunk. Based on the problem formalisation, we describe in the next section the conceptual framework for our STAR technique in terms of its phases and components.

4.5 Conceptual Framework

In this section, we introduce our technique for recognising personalised activities with incremental learning, STAR. In terms of the learning paradigm, STAR is divided across two phases: offline and online. In the offline phase, the modelling component (MC) builds the baseline framework (BLFW) from historical data that represents different activities for training. The output of the offline modelling phase is a fine-grained learning model that represents activities existing in the training instances. The learning model is built with training data that is collected from a group of users. However, users perform activities in different ways. What has been represented in the model as 'jogging' for one user could be 'walking' for another. Therefore, personalisation

of the learning model to suit a particular user is an important issue for improving the recognition performance. One way of model personalisation is to retrain the model with a user's personalised data. Given the known issue of scarcity of labelled data, the retraining solution is not applicable especially in a streaming environment when data arrives at high speed and requires real or near real time recognition. Therefore, the learning model needs to be tuned and personalised automatically and in an incremental approach to fit a specific user.

In order to achieve model personalisation with the evolving data streams, we introduce the online phase of our technique. Two components operate in the online phase, recognition component (RC) and personalisation component (PC). The recognition component integrates the learning model (BLFW) with an ensemble classifier for recognising activities from data streams. The initial BLFW is refined by the personalisation component through incremental and active learning. The refined BLFW then replaces the old one in the recognition component for more accurate recognition of performed activities.

Figure 5.4 shows the conceptual framework of STAR with its phases and components. The initial BLFW is built offline in the modelling phase, then integrated in the recognition component for initial recognition. The recognition component consists of an ensemble classifier integrated with the BLFW. The streaming data is handled by the recognition component. A data chunk is the processing unit for recognition and personalisation. The recognition decision is checked for eligibility with the personalisation component. The eligible data is used for either incremental or active learning. Both learning approaches modify/tune the BLFW based on the selected eligible data. The main difference between the two approaches for active and incremental learning is as follows. Incremental learning refines the model automatically with predicted information; active learning asks for user input for ground truth information. The refined model replaces the general/impersonal initial model for further recognition of streaming chunks.

The overall process for our technique is depicted in Algorithm 4.1. Subsets of the algorithm are described in subsequent sections corresponding to each component. The modelling component has been explained earlier in this dissertation in Chapter 3. The sliding window technique is applied on a data stream to capture the chunk of data at time t (line 3). The recognition component processes the chunk of data at time t to get a recognition decision (line 4). The decision is based on the ensemble classifier that is deployed for recognition. This decision comprises the predicted labels of activities occurring in the chunk with a specific confidence level. Based on the recognised decision, the

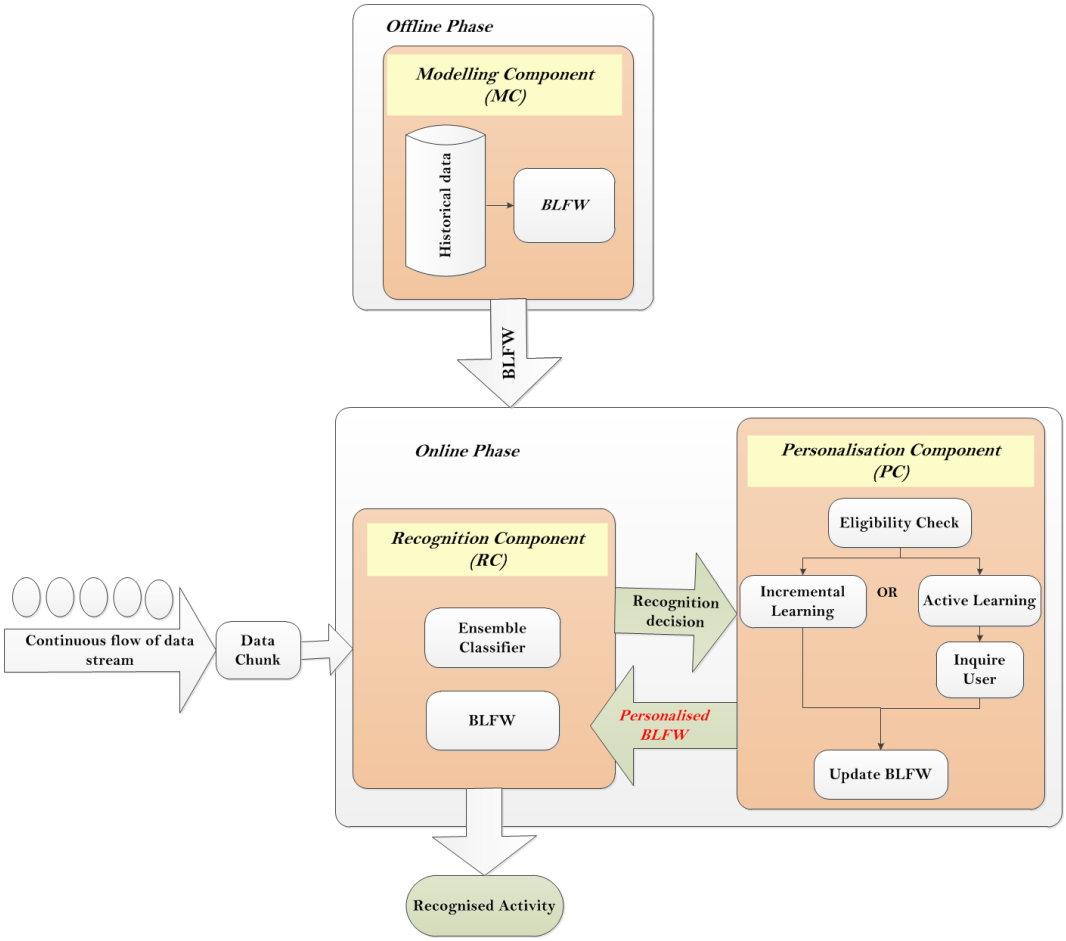


Figure 4.3: STAR Conceptual Framework

personalisation component may refine the model with data in a chunk when eligible (line 5). The updated model replaces the old model for the recognition of the next chunk at time (t+window size) (line 6).

Algorithm 4.1 STAR top level Algorithm

Input: $Data_{Historical}$: Annotated training data for building the initial $BLFW$
$Stream$: Sensory data evolving from sensors
Output: $RecognD$: The recognition decision

1: $BLFW \leftarrow$ **ModellingComponent** $(Data_{Historical})$
2: **while** $Stream$ not empty **do**
3: $Chunk_t \leftarrow$ **SlidingWindow**$(Stream)$
4: $RecognD_t \leftarrow$ **RecognitionComponent**$(Chunk_t, BLFW)$
5: $BLFW_{new} \leftarrow$ **PersonalisationComponent** $(RecognD_t, Chunk_t)$
6: $BLFW \leftarrow BLFW_{new}$
7: **end while**

After discussing a detailed overview of the STAR framework and its components, in the following sections we present an explanation of the methodology for each of STAR components. We start with a brief recap of the modelling and recognition components, in the context of STAR, that have been presented

in Chapter 3. That will be followed by a detailed explanation of the STAR personalisation component and methodology.

4.6 Modelling Component

This component has been explained in detail earlier in this dissertation in Chapter 3. There are two approaches to build the BLFW in the modelling component: clustering with dominant pattern selection and clustering with sub-clusters. Based on the advantages that are demonstrated in the evaluation of the BLFW approaches in CBARS, we choose to apply the clustering with sub-clusters approach for modelling in STAR. This approach satisfies a set of key desired features for building an *accurate*, *robust*, *computationally efficient*, and *flexible* technique for activity recognition. These desired features are depicted as follows:

- *Fine-grained model:* Clustering with sub-clusters approach for building the BLFW represents a more accurate representation of activities and patterns for each activity. Building a fine-grained model that describes clusters of different activities in detail is essential for *robust* and *accurate* recognition. The sub-clusters model provides microscopic information about each activity and the patterns inside the activity. For instance, in a cluster representing the "walking" activity. "Walking" is identified as a specific activity as it is carried out in several contexts (e.g. exercise, recreation, transport, at work) [ABD00]. The sub-clusters model creates sub-clusters of different walking patterns existing in the training examples such as brisk walking for exercising, nordic walking, and walking in a workplace. Distinguishing between different patterns existing in an activity helps with enhancing the recognition *accuracy*. The sub-clusters model tends also to decrease the effect of noise on genuine activity data and thus boost the model *robustness* to noise.

- *Lightweight model*: One key challenge in the modelling process is to keep the balance between accurate representation of the data and the complexity. The sub-clusters model achieves this balance by building the model with a detailed representation of activities, but at the same time extracts only the essential summarised characteristics of these activities to build the model and then dismiss all raw data. The compact and lightweigh representation of the data in the sub-clusters model helps enhancing the *computational efficiency* of the recognition technique. This feature also

allows the deployment of the recognition technique on a limited resources portable device such as mobile phone.

- *Flexible model*: A significant feature that is essential for deploying the BLFW in STAR is its ability to support personalisation in terms of flexibility. Since clusters and sub-clusters are represented by a set of summarised characteristics, in order to refine the BLFW we only need to update the summarised characteristics of the model with no need of re-processing the entire training examples - which are already dismissed at this point. Updating the BLFW requires the refinement of simple characteristics such as cluster centroid and size. Therefore, the updated BLFW is personalised to suit a user specific way of performing activities and thus enhances the accuracy for future recognition.

Using the BLFW with the aforementioned desired features is a key factor for an efficient recognition. The deployment of the BLFW into the online phase for recognition is discussed in the next section.

4.7 Recognition Component

The online phase in STAR consists of both the recognition and personalisation components. The two components are implemented for real-time recognition and personalisation. Firstly, the BLFW is integrated into the recognition component for prediction of activities in the incoming streaming data. Then, the personalisation component refines the BLFW in an incremental and active learning approach for future recognition. We focus on the details of the recognition component in this section. The recognition component has been discussed earlier in Chapter 3. The integration of the BLFW and the ensemble classifier is the same as explained with CBARS. However, there are two main extensions in the recognition component in STAR that make it different from the one applied in CBARS. The first extension is related to the deployment of the recognition component in the online phase. That includes the methodology of handling the streaming nature of data. The second extension is for the ability of STAR to recognise concurrent and interleaved activities. The key challenge of the online recognition component in STAR is to perform the required data processing for recognising activities from the data streams in an efficient and accurate manner in real time. The rest of this section describes the two extensions of the STAR recognition component: handling streaming data and detecting concurrent and interleaved activities.

4.7.1 Handling streaming data

The recognition component in STAR handles the streaming data in a sequential way. Data streams in activity recognition consist of sequences of chunks of data representing different activities. Activities occur in sequences and typically last for a (possibly brief) time window. Therefore, processing activity data in a sequential way is more intuitive and also more efficient. In the context of data stream mining, tailored techniques have been especially developed for handling the streaming data. The reason for developing these methods is the high speed nature of a data stream which requires special approaches that are capable of handling the incoming data. Examples of these techniques are sampling, windowing, and data segmentation. In our recognition component, we apply a *fixed size and continuous sliding window* for handling the sequential chunks of activities in data streams.

The recognition component aims to recognise the activity occurring in each data chunk with a cluster-based approach. Due to the sequential nature of activity data, the recognition component avoids the inefficient and unnecessary processing of each instance, which is costly in a streaming environment. Instead, it processes data in each chunk as a cluster rather than responding to each data instance. In stream classification, most techniques such as VFDT [DH00] - classify each instance in data chunks. Few studies have considered a cluster-based approach for classification in streaming settings such as in [SCG07]. Yet these techniques are not tailored for the context of activity recognition. In our approach, we process the chunk of data collectively. Thus, the processing unit in *STAR* is the chunk instead of instances of the chunk.

The predicted activities of the cluster-based approach correspond to the major activity performed in the target data chunk at time t. For instance, consider processing a chunk of data that contains 500 instances captured with a sliding window of length 5 seconds. The majority of instances (more than 95%) have the true label of the activity that is performed in this chunk: "walking" in this example. With the cluster-based approach, instead of reacting to each instance, the chunk can be processes as a whole for recognising the majority label which corresponds to the major activity performed in this chunk. Thus, the sequential and cluster-based approach for handling data streams in STAR is expected to enhance both the *accuracy* and *computational efficiency* in STAR.

4.7.2 Capturing concurrent and interleaved activities

One of the challenges with the sequential cluster-based approach for recognition is how to handle concurrent and interleaved activities. The chunk of data

may contain more than one activity. Typically, some activities are highly correlated such as "walking" and "standing". Even in a small chunk of data, one user would have pauses of "standing" activity, for instance, while "walking". In our cluster-based approach we aim to detect the major activity performed in each chunk. In this example, only the major activity, namely walking, will be detected. However, capturing concurrent and highly related activities improves the overall accuracy of the recognition technique. Therefore, we combine the cluster-based approach with concurrent activity detection to be able to recognise concurrent and interleaved activities in each chunk.

In order to capture concurrent and interleaved activities in a single chunk, we apply an online clustering method for each chunk - as explained in Figure 4.4. The clustering step aims to detect and separate various activities that appear in the same chunk. We apply well-known clustering techniques such as K-means or EM in the online phase to the small size chunk. Clustering of chunk data also helps isolate outliers and noisy data as it tends to group similar data together. Thus, it limits the noise and outlier impact on genuine activity data instances. The formed clusters of each chunk are assessed using the recognition component to recognise their corresponding activities.

Figure 4.4 shows the overall process of the recognition component. Firstly, a continuous fixed size sliding window is applied on the stream from time *(t)* to *(t+window size)*. The chunk of data is saved to an online buffer for real time processing while accumulating the new incoming data instances of the following window from time *(t+window size)*. For each chunk, we apply an online clustering method for detecting the concurrent and interleaved activities. The recognition component aims to recognise the major activity in each chunk cluster. The chunk clusters are then processed to recognise the activities with the ensemble classifier that is integrated with the BLFW in the recognition component.

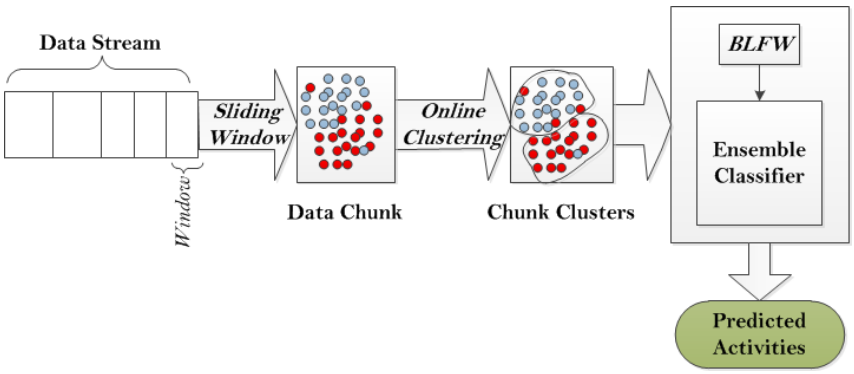


Figure 4.4: Recognition Component

The ensemble classifier is a lightweight algorithm based on a hybrid similarity measures approach for prediction. Thus, the classifier deploys an ensemble of four measures to assess the similarity of each cluster in the chunk with the BLFW. State-of-the-art classification methods adopt a single measure for prediction, combing various measures benefits from the strength of each and provides a comprehensive understanding of data from various perspectives. Each measure votes for its own "best prediction" from that measure's perspective. Then, the classifier decides upon the predicted label as the one with the majority votes among all measures. The deployed measures are distance, density, deviation, and gravity. The details of the ensemble classifier has been revisited in Chapter 3.

Algorithm 4.2 describes the steps of the online recognition component in STAR. The online clustering method for detecting concurrent and interleaved activities is applied to each chunk of data (line 1). For each cluster in the chunk, the ensemble classifier applies the four similarity measures to predict the cluster label/activity. The procedure is repeated for all clusters in the chunk and for all chunks in the stream. The algorithm assimilates the recognition component algorithm in Chapter 3 (Algorithm 3.5), yet with modification to adapt to data streams.

Algorithm 4.2 RecognitionComponent

Input: $BLFW$: the baseline framework (the output of the modelling component)
$Chunk_t$: The chunk of the data stream from the sliding window
Output: $RecognD_t$: The recognition decisions for the chunk activities

1: $chunkClusters \leftarrow$ **OnlineClustering** ($Chunk_t$)
2: **for all** $cluster_i \in chunkClusters$ **do**
3: $DisCand \leftarrow$ **FindDistanceCandidate** ($cluster_i$, $BLFW$)
4: $DensCand \leftarrow$ **FindDensityCandidate** ($cluster_i$, $BLFW$)
5: $GravCand \leftarrow$ **FindGravityCandidate** ($cluster_i$, $BLFW$)
6: $SDCand \leftarrow$ **FindSDCandidate**($cluster_i$, $BLFW$)
7: $RecognD_t[i] \leftarrow$ **Vote** ($DisCand$, $DensCand$, $GravCand$, $SDCand$)
8: **end for**

The recognition component aims to handle the stream of data for recognising activities in each chunk in an online manner. However, the recognised activity is not personalised yet. To refine and personalise the learning model, the recognition component sends the recognition decision to the personalisation component for refining the BLFW then it can predict the personalised activities. Details of the personalisation component are in the next section.

4.8 Personalisation Component

The second online component in STAR is the personalisation component. The key objective of the personalisation component is to update and refine the BLFW while the stream evolves with the user's personal data. The initial BLFW, which has been built in the modelling component, needs to be tuned to fit a user's specific way of performing activities. The degradation of the BLFW while a stream evolves requires a continuous refinement to cope with changes in data. The personalisation component implements this continuous refinement through batch incremental and active learning approaches. Thus, STAR updates the BLFW to be consistent with the most recent changes in data streams.

Efficient personalisation has to satisfy a set of key criteria. For a *computationally efficient* recognition, the personalisation processes have to use simple calculations for refining the model with low time and space complexity. The personalisation has also to perform in real or near real time to boost the recognising *accuracy* for the future incoming data. The personalisation intuitively addresses the desired feature of *flexibility* for an efficient recognition technique. However, it is challenging to update the BLFW that is only represented by a set of characteristics, as all raw data applied to build the BLFW is dismissed beyond the offline modelling component. We address the desired features of computational efficiency, accuracy, and flexibility in our novel personalisation component.

The personalisation component receives as an input the recognition decision from the recognition component. An eligibility check is performed on the decision to check its validity to perform either incremental or active learning. By applying the incremental or the active approach, the BLFW is refined and thus replaces the outdated model. In the following, we explain in detail the eligibility check for the recognition decision. Then, we present the procedure of refining the BLFW for personalisation with incremental and active learning.

4.8.1 Eligibility check

The eligibility check in the personalisation component is based on the *Vote* procedure in Algorithm 4.2. The vote procedure chooses the best candidate from four candidates selected by the represented measures. We apply a balanced vote among measures to select the candidate with the majority votes. Each recognition decision has a confidence level attached that represents the certainty among all measures for choosing the best cluster. For example, when three out of the four measures vote for the "jumping" cluster/activity, while

one measure votes for "climbing stairs", then the recognition decision will choose the activity label of "jumping" with the majority vote. The confidence level in this example is 75% because three out of the four measures have voted for "jumping". In some cases, the voting procedure lacks the ability to decide because of confusion occurred among measures. For example, when each measure votes for a different label and not any pairs are alike, then the confidence level for any of these labels will be only 25%. The eligibility check is tightly related to the confidence level of the recognition decision. The personalisation is applied with the recognition decision that has the lowest confidence level. There are two kinds of learning applied for personalisation: active and incremental learning.

The approach of active learning deals with the known problem of the limitation of the availability of annotated data. In typical personalised activity recognition systems in the literature, they require the user to spend some time for retraining the model to his personal way of performing activities [GKG+12a]. However, the retraining process is impractical and costly especially in a streaming environment. Alternatively, it only selects the data with the lowest confidence level (the most uncertain data) from the recognition decision to be labelled by user. Also, a batch active learning approach helps reduce the labelling cost, as it makes inquiries to determine a cluster label instead of inquiring for each instance. We augment our model with this data to inject expert knowledge for more informative and flexible learning. Refining the model with the true labelled data through active learning is necessary to maintain a reasonable level of certainty. Yet, inquiring about true labels has to be at low cost, as too many inquiries might cause system inefficiency.

We first explain the two cases that trigger active learning in the personalisation component. In both cases, the recognition component is incapable of predicting the activity label (cases when the recognition component cannot reach a decision). The first case occurs when the confidence level for recognising a label of the chunk cluster is at the lowest certainty level of 25%. That means each measure votes for a different label, and not any pairs are alike. The other case of active learning happens when the four measures are confused between exactly two labels with equal votes to each. The confidence level in this case is 50% for each of the two candidates. With the aforementioned cases, no decision is made by the recognition component. Thus, in these two cases, the personalisation component asks the user for the true label of the chunk cluster data. The user is not required to label each single instance in the chunk cluster. Alternatively, the user is only asked to provide the majority label in the cluster, which corresponds to the major activity performed at this particular

time. The user also has the option to choose no label if it was not a specific activity (i.e. transition between two activities). Then the provided true label is assigned in a batch mode to all instances in the cluster. The cluster data along with the provided label are fed back to the system in batches to refine the BLFW. The refinement procedure is discussed later in this section.

Unlike active learning, incremental learning does not seek user input for refining the BLFW. It uses the recognition decision with the *predicted* label for automatic personalisation. The case of incremental learning happens when the recognition component can successfully reach a decision with the lowest confidence level of 50%. To illustrate more, consider an example when two of the four measures vote for the "standing" activity while the other two vote for two different labels such as "running" and "walking". The recognition decision in this example will be for "standing" based on the majority votes yet with the lowest confidence level of 50%. We apply incremental learning in this most uncertain case of recognition. The reason for choosing the most uncertain data is to expand the BLFW representation of activities for including the borderline cases that is recognised the lowest certainty. The recognised data that is most uncertain is more likely to represent a drift of the model. This way, the representation of the model will be strengthened by refining that includes this data. Therefore, the selected cluster data with its predicted label are applied to refine the BLFW in batches with the refinement procedure.

Both active and incremental learning are triggered automatically based on the recognition confidence level. They both aim to adjust the BLFW to cope with dynamic data streams. However, active learning updates the BLFW with the true label provided by the user, whereas incremental learning updates it automatically with the predicted label for unlabelled data. Thus, active learning has the advantage of providing a ground truth of an activity's true label. Yet, it requires the user's input at real time, which is costly and might impact the efficiency negatively. On the other hand, incremental learning applies an automated refinement that does not require user input for personalisation. Nevertheless, the accuracy of the predicted labelled that used for refinement is not certain. That consequently might increase the risk of model drift from the correct representation of activities due to accumulated refinements with uncertain values. Therefore, it is important to keep the balance between active learning with low cost and incremental learning to attain an efficient personalisation.

Table 4.1 summarises the different decisions of active, incremental, or none based on the confidence level of the recognition decision. Both active and incremental approaches refine the BLFW with the selected data.

Table 4.1: Confidence Level with Recognition and Personalisation Decisions

Confidence level	Description	Recognition decision	Personalisation decision
25%	Each measure chooses a different cluster	None	Active
50%	Equal votes between exactly 2 clusters $activity_i$ and $activity_j$	Inquire to choose $activity_i$ or $activity_j$	Active
50 %	Two measures vote for $activity_i$ with no equal votes	$activity_i$	Incremental
75%	Three measures vote for $activity_i$	$activity_i$	none
100%	all measures vote for $activity_i$	$activity_i$	none

4.8.2 Personalisation methodology

The personalisation component aims to refine the BLFW extracted characteristics in real time, with the most recent changes in activity data streams. Once the data is selected for updating the BLFW, the model is refined and replaces the old one for processing the data of the new incoming data.

As discussed in Chapter 3, the characteristics of the BLFW are categorised in three levels: holistic, clusters, and sub-clusters characteristics. For refining the model, we update the characteristics of the smallest entity which is at the sub-cluster level and that will consequently update the characteristics of the bigger entities at the clusters and holistic levels. An example of that is the *centroid* which appears at the three levels as sub-cluster centroid, cluster centroid, and global centroid. Movement of the sub-cluster centroid will cause changes in both cluster and global centroids. When refining the model, we update the centroid of the sub-cluster, and then the centroid of the cluster that contains the updated sub-cluster is consequently updated. The holistic characteristic of global centroid is also refined with the updated values of the clusters' centroids.

The personalisation methodology has an input of the annotated selected data ($data_{sel}$). The label of selected data ($label_{sel}$) is either provided by the user, in active learning; or predicted by the recognition component, in incremental learning. $label_{sel}$ matches the label of one of the BLFW clusters, C_i. As we apply the personalisation methodology on the sub-clusters level, we run a one pass distance test to choose the closest sub-cluster sc_j to update within the cluster C_i. Then, the personalisation methodology applies a set of simple algorithms to update sc_j characteristics. The updates of some characteristics, such as *Weight*, are as simple as accumulating the value with the weights of

the selected data. While other characteristics such as the *cluster gravity - GF* - is automatically updated according to Equation 3.10 in Chapter 3 with the refined sub-clusters' *weights* and *centroids*. Some other characteristics, such as *centroid*, require a special incremental learning technique for updating. The personalisation methodology updates higher level characteristics based on the updated lower level ones. An overview of the personalisation methodology is presented in Figure 4.5.

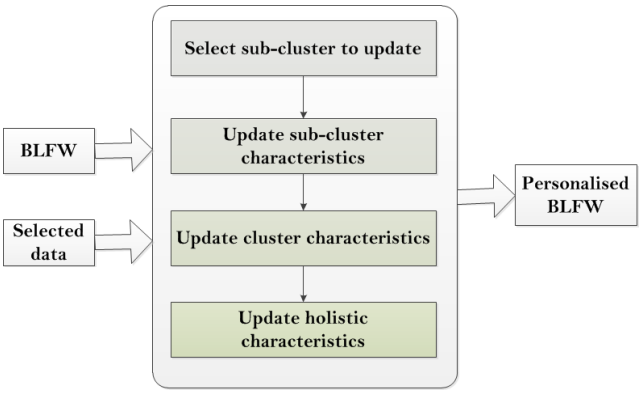


Figure 4.5: Personalisation Methodology Overview

One of the most dominant characteristics of the BLFW is the *centroid*. Accumulation of the selected data will cause movement of the sub-cluster centroid. Algorithm 4.3 explains the procedure to incrementally update the centroid of the sub-cluster. A numerically stable algorithm for weighted centroid update is given below in applied in our technique. It computes the mean due to Knuth [Knu97].

Algorithm 4.3 Centroid_{sc}IncrementalUpdate

Input: $data_{sel}$: The data selected for updating the BLFW
  $Centroid_{old}$: centroid of the sub-cluster before update
  $Weight_{old}$: weight of the sub-cluster before update
Output: $Centroid_{sc}$: The updated sub-cluster centroid
  $sumweight = Weight_{old}$
  for all x in $data_{sel}$ **do**
    $temp = weight_{data_{sel}} + sumweight$
    $delta = x$ - $Centroid_{sc}$
    R= $delta$ * $sumweight$ / $temp$
    $Centroid_{sc} = Centroid_{sc} + R$
  end for

The centroid update algorithm is extended for variance and standard deviation [CGL83]. Algorithm 4.4 illustrates the refinement of *WISCSD* for the selected sub-cluster -sc_j.

Another essential characteristic in the BLFW is the *density*. The density calculation is previously described in Equation 3.6 in Chapter 3. When merging the new instances of the selected data with the instances of the selected

Algorithm 4.4 WISCSD$_{sc}$IncrementalUpdate

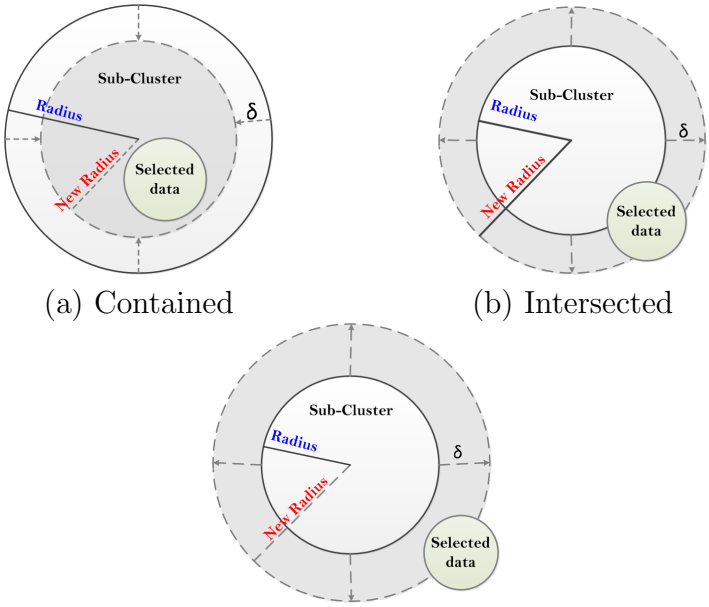
Input: $data_{sel}$: The data selected for updating the BLFW
$WISCSD_{old}$: The WISCSD of the sub-cluster before update
$Weight_{old}$: The weight of the sub-cluster before update
Output: $WISCSD_{sc}$: The updated standard deviation of
$sumweight = Weight_{old}$
$M = WISCSD_{old} * n^2$
for all $x \in data_{sel}$ **do**
$sumweight = sumweight + 1$
$delta = x$ - $Centroid_{sc}$
$Centroid_{sc} = Centroid_{sc} + delta/n$
$M = M + delta*(x$ - $Centroid_{sc})$
end for
$variance = M/sumweight$
$WISCSD_{sc}$= SQRT($variance$)

sub-cluster sc_j, we recalculate the density based on the updated characteristics. We initially update the sub-cluster's weight by adding up the weights of the selected data. Then, we update the sub-cluster radius, which defines the boundary of the sub-cluster. The updated sub-cluster radius relies on the positions of the selected data from the sub-cluster and the weight of the selected data as well. As illustrated in Figure 4.6, there are three possible positions of the selected data from the sub-cluster: contained, intersected, or separated. The radius of the sub-cluster shrinks or expands based on the selected data position. When the selected data is fully contained inside the sub-cluster, the radius is tightly adjusted to fit the updated data. In the other two cases, the radius is expanded to accommodate the new data. The radius shrinkage and expansion is related to the centroid movement due to refinement as well (the distance between old and updated centroid). Therefore, we calculate the change of the radius according to Equation 4.1. We expand (+) or shrink (-) the radius with a value of δ that relies on both the relevant weight of the selected data to the sub-cluster weight and the centroid movement. Increasing or decreasing the radius with δ helps maintain the stability of the sub-cluster and other entities' representations.

$$Radius_{new} = Radius_{old} \pm \delta \tag{4.1}$$

$$\delta = (\frac{weight_{sel}}{weight_{sc}}) \times CentroidMovement \tag{4.2}$$

Where $weight_{sel}$ is the selected data weight, $weight_{sc}$ is the sub-cluster weight, and $CentroidMovement$ is the Euclidean distance between old and updated centroids.



(a) Contained (b) Intersected

(c) Separated

Figure 4.6: Radius Update Cases

To sum up, the personalisation component tailors the BLFW in STAR to fit a user's personalised way of performing activity. While streams of activities evolve, the personalisation component automatically selects the user's data that is eligible for personalisation. The user's personalised data is used directly to tune the characteristics of the BLFW to fit this particular user. The personalised BLFW then replaces the outdated model for more accurate and customised recognition of the user's activities at the next data chunk. We also address in this component the known challenge of the scarcity of labelled data by applying active learning. The approach of active learning selects only a small amount of data for annotation to personalise the learning model. Thus, active learning overcomes the need of labelling all the data for retraining the model for personalisation.

4.9 STAR Contribution

In this Chapter, we proposed and developed a personalised technique for activity recognition that addresses key challenges in AR. The developed technique deals with high speed, multi-dimensional streaming data to learn, model, and recognise personalised and evolving user's activities. The novel approach extends the state of the art in activity recognition by providing the following advantages:

- *Model personalisation with evolving data streams:* There is no one model that fits all in activity recognition. Thus, we build a novel learning

approach for refining and personalising the learning model for a specific user. Our technique refines the learning model with an incremental learning approach to continuously learn from evolving activities. We also address in our technique the key challenge of handling the high speed, multi dimensions streams of data that evolve dynamically. The flexibility of the learning model to be personalised will directly enhance the recognition *accuracy*.

- *Effective active learning with lowest cost:* Manual labelling of data streams is usually impractical, costly, and time consuming. Therefore, we present our active learning approach that asks to label only a small amount of data that is most uncertain in the stream for a dynamic model refinement and personalisation.

- *Capture sequential and concurrent activities:* Activity recognition data is typically represented as sequential chunks of activities evolved from sensory data sources. Thus, our approach processes data sequentially with a sliding window to capture either a single activity or concurrent activities appearing in each data chunk. A highlight of our novel technique is its ability to capture multiple activities occurring at the same data chunk. This is very likely to happen in the activity recognition domain, especially with interleaved and interrelated activities. Our technique applies a clustering online to capture the occurrences of concurrent and interleaved activities.

- *Efficient activity recognition technique:* Our proposed technique is computationally efficient with a lightweight model and simple recognition and personalisation algorithms. We demonstrate the effectiveness of our technique by deploying the activity recognition application on a mobile platform for real time recognition.

4.10 Experimental Study

In this section, we evaluate our proposed and developed technique for activity recognition. Our technique aims to achieve an automatic personalisation with incremental learning while streams of activities evolve. To assess the efficiency of the developed technique, we aim to evaluate the following:

- *Aim 1: demonstrate the impact of personalisation on the performance:* The evaluation tests the ability of the developed technique to perform personalisation for an accurate recognition. To highlight the effect of

personalisation, we train the learning model with data from only one user and test on data from another user. We discuss the impact of personalisation on the overall accuracy and also on the accuracy of each activity. The personalisation evaluation aims to show the flexibility of the model to change over time to fit a user's personalised way of performing activities.

- *Aim 2: evaluate active and incremental learning efficiency:* Part of the performance evaluation of our technique is to monitor the rate of active learning along the stream. Keeping the balance between inquiring for user input very often and providing the model with a sufficient amount of true labels is a key challenge that we investigate for fulfilling this aim.

  We also examine the correlation between active and incremental learning. Active learning has the advantage of feeding true labels into the learning process. However, it requires users input to get the true label, which is a costly process if required very often. On the other hand, incremental learning occurs automatically with the recognition decision. However, the label of fed back data is not always correct and therefore it might cause the model to drift away from the correct presentation of activities. Therefore, finding the balance between active learning to get the ground truth and incremental learning to attain continuous and automated learning is a key motivation of our analysis.

- *Aim 3: investigate the efficiency of STAR to run in real time with a limited resource device:* We aim to evaluate the efficiency of the developed technique for real time recognition with high dimensional data with high sampling rates. We deploy our technique on a mobile phone for analysing the technique's efficiency on a limited resource device. The time and space complexity are analysed for evaluating the system performance.

We start by discussing the datasets that are applied for evaluating our new technique. Then, we explain the setup of the experiments. That will be followed by analysis and discussion of the results to achieve the aforementioned aims.

4.10.1 Datasets

We conduct our experiments on the three real and publicly available datasets presented in Chapter 3, namely OPPORTUNITY [RFC+09], PAMAP2 [RS12b, RS12a], and SPAD [DLL13]. Both OPPORTUNITY and PAMAP2 are collected from on body sensors, while SPAD contains data collected from mobile

phone accelerometer data. We add to the list an additional dataset for activity recognition that was collected from accelerometer mobile sensors. The WISDM [KWM11] dataset contains six activities that were performed by the user during data collection, namely, "Walking", "Jogging", "Sitting", "Standing", "Upstairs", and "Downstairs". The dataset is collected by different users and contains more than 1 million instances of annotated accelerometer data. The raw data, collected from a mobile accelerometer sensor in three dimensions (x,y,z), is provided for STAR evaluation with the SPAD and WISDM datasets. On the other hand, OPPORTUNITY and PAMAP2 contains higher dimensional data with 52 features in PAMAP2 and 110 features in OPPORTUNITY collected from on body sensors. The diversity of the datasets' sources and characteristics is important, to show the ability of our technique to perform effectively in different settings.

4.10.2 Experimental setup

We run our experiments on both mobile phone and desktop. The mobile phone is a *Galaxy SIII Android* with the specifications of a Quad-core 1.4 GHz processor and 1 GB of RAM, while the desktop is Core i7 with speed of 2.7 GHZ and 4 GB RAM. For all experiments unless otherwise stated, part of the data is used to build the BLFW offline. Other data from different users is applied for testing. The BLFW is built using the clustering with sub-clusters approach. We assume that the maximum number of patterns inside any activity is 5. Thus, the default number of sub-clusters in each cluster for building the BLFW is set to 5 as a default. The testing data is a stream of unlabelled data that has not been used for training the model. Labels are only revealed for the evaluation purposes. In this analysis, we define the terms of performance metrics as follows:

- TP= total number of instances in the stream that are correctly classified,
- $ALRate$= number of triggered active learning inquiries,
- ACT= percentage of data triggers active learning,
- INC= percentage of data triggers incremental learning, and
- $CPur$= percentage of the cluster purity; which is the percentage of instances with the major label inside the cluster,

TP is the basic measure to show the overall accuracy and also the accuracy for recognising each activity. The performance of active learning is measured

by two metrics: $ALRate$ and ACT. $ALRate$ shows the number of inquiries to label the data. The active learning approach is a batch technique that labels a whole cluster instead of labelling each instance that requires active learning. The size of the cluster relies on the sliding window length and thus the chunk size. For instance, for ACT=500, which means that 500 instances require active learning, $ALRate$ is equal to 50 when the chunk size is 10, and 5 when the chunk size is 100. ACT and INC represent the percentage of selected data based on the recognition decision for incremental and active learning. This data is fed back to update the BLFW in the personalisation component.

The selection of the window size is based on many parameters related to the dataset. These parameters include the sampling rate and the minimum duration of any activity occurrence in the dataset. For instance, the sampling rate of PAMAP2 dataset is 100 Hz, a window size of a single second will create chunks of 100 instances. However, the minimum duration of the activities in this dataset is 1 minute. In order to reduce the computational cost, we still need to minimise the window size. Given the above parameters, we tune the window length for PAMAP2 dataset between (6–8) seconds that contains 600-800 instances in each chunk. For other datasets that are highly fluctuating, such as the OPPORTUNITY dataset, we tune the window length between (1–2) seconds. The sampling rate for WISDM and OPPORTUNITY datasets is 20–30 Hz. Thus a window size of (1–2) seconds will contain chunks of 20–30 instances. The chunk size of the SPAD dataset that has a sampling rate of 5 Hz is 5–10 instances.

For each data chunk, we apply EM clustering in the online phase for detecting concurrent activities. We assume the number of concurrent activities in each single chunk is limited to two activities. Considering the small chunk size and sampling rate for sensors, we believe this limitation is valid.

4.10.3 Analysis and discussion

This section addresses the aforementioned aims of the evaluation. We start with an analysis of the deployment of STAR on the mobile device and an analysis of its computational efficiency (aim 3).

Time and space complexity analysis

Two parameters influence the processing time in STAR. The first one is the dimensions of data in terms of number of features while the other is the percentage of incremental learning along the stream. We start with analysing the STAR processing time with 3–D data collected from a mobile accelerometer

sensor. Both WISDM and SPAD are 3–D datasets with sampling rates of 20 Hz and 5 Hz respectively. The chunk size of 20 instances is equal to an observation time of 1 second in the WISDM dataset; 4 seconds in the SPAD dataset. STAR is deployed on the mobile device and desktop with the aforementioned specifications. The processing time for each chunk in both datasets is (50–80) msec on the mobile phone and drops to only (4–10) msec when deployed on the desktop. While the sampling rate in both datasets is longer than the processing time, STAR successfully fulfils the real time constraint for these datasets.

Unlike, the 3–D datasets, number of features in the OPPORTUNITY dataset is 110 and 52 for PAMAP2. In the OPPORTUNITY dataset, the processing time for a chunk of 20 instances is on average 900 mesc on the mobile phone, (57–90) msec for desktop deployment. The sampling rate for the OPPORTUNITY dataset is 30 Hz (approximately 1 sample every 33 msec). Thus, the observation time for a chunk of 20 instances is about 700 msec. Indeed STAR still performs a real time or near real time recognition (for mobile deployment) despite the high data dimensionality. The sampling rate of the PAMAP2 dataset is 100 Hz. Because of the high sampling rate, we apply K-means instead of EM in online clustering of chunks for faster processing, to cope with the high sampling rate. For a chunk of 100 instances and observation time of 1 second, the processing time varies in the range (350–500) msec on the mobile phone and (10–15) msec on the desktop. Despite the high sampling rate of the PAMAP2 dataset, STAR successfully achieves real time recognition.

The sampling rate of WISDM and OPPORTUNITY are 20–30 Hz. SPAD has a lower rate of 5 Hz. A chunk size of 20 instances corresponds to an approximate observation length of 0.7 sec, 1 sec, and 4 sec for OPPORTUNITY, WISDM, and SPAD datasets respectively. The reason for the small chunk size is to keep the computational complexity to a minimum. The number of instances in each window influences the order of the computational complexity. The recognition algorithm as a cluster-based approach tests new chunk clusters. We apply EM clustering for each chunk with a complexity of $O(nki)$, where n is the number of instances in each chunk, k is the number of clusters and i is the number of iterations. As the observation time for each window varies from 0.7 seconds to 4 seconds, we limit the number of interleaved, concurrent, or transitional activities to only 2, which is very reasonable based on the small window size. Thus, the complexity for clustering data in each window is $O(2ni)$.

In the PAMAP2 dataset with sampling rate of 100 Hz, we increase the chunk size to 100 instances for an observation time of 1 second. Thus, we limit the effect of the big number of instances in PAMAP2 by applying K-means for online clustering instead of EM. The better running time of K-means over EM is because of the number of iterations (i). In EM, each instance is assigned relatively to all clusters. Then the clustering results are optimized until reaching a threshold (which could be after many iterations) [DND77]. K-means, on the other hand, assigns instances to the nearest cluster based on the distance metric. It converges when the assignments no longer change [Har75]. As the termination behaviour of K-means is faster than EM, K-means is more suitable to be applied on data with high sampling rate that requires processing bigger data chunks.

The main computational complexity in STAR arises from the online clustering algorithm. The recognition technique complexity is O(m), and m is the number of sub-clusters in the BLFW. Whereas the complexity of the incremental personalisation techniques is O(nm), where n is the number of instances within a single chunk. The computational complexity in a typical KNN approach is O(nM) where M is the number of stored instances. Obviously (nM) is much higher than (nm), where m (number of sub-clusters) $<< M$ (number of stored instances in the training phase). Moreover, the personalisation process is activated only for a subset of the data along the stream (through model personalisation). Therefore, the complexity of the personalisation technique has less impact on the overall computational complexity.

We deployed STAR on a mobile phone to demonstrate its efficiency in an ubiquitous computing environment. Figure 4.7 displays screen shots of the developed mobile application for activity recognition. The accelerometer sensor reading along the three axes is displayed at the top as shown in the screen shots. The application displays the sequence of recognised activities that correspond to each window in real time as per Figure 4.7(a). In case that STAR is unable to recognise an activity, an active learning alert pops up for user to select the correct label for the unrecognised activity - as shown in Figure 4.7(b). The selected label along with the confusing data are fed back to the system in order to personalise the model instantaneously. In Figure 4.7(c), the application confirms the user selection of the "Sitting" activity as an input for active learning. The application continues to recognise the streaming activity with the adapted model as per Figure 4.7(d).

One of the key contributions of STAR is the ability to build a fine grained, lightweight learning model. Table 4.2 shows different datasets' characteristics and the corresponding BLFW size in Kilobytes. As shown in the table, the

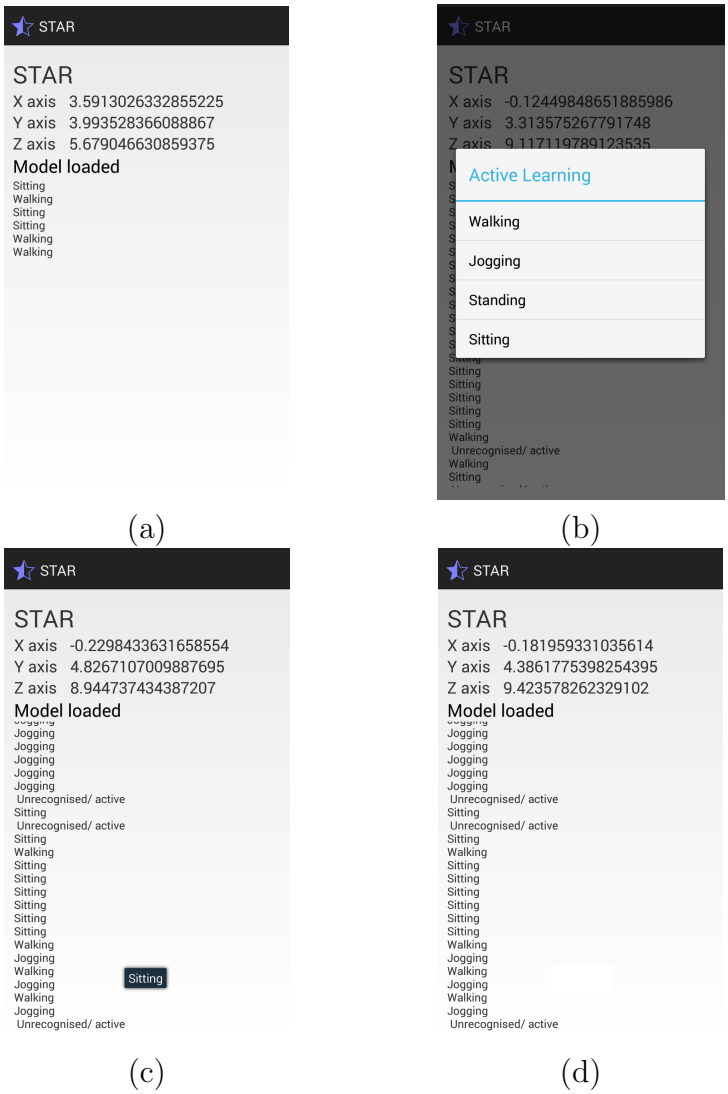


Figure 4.7: STAR Mobile Application Screen Shots

size of the BLFW with a high-dimensional training data is just above 100 KBs. On the other hand, the model size is as small as a few kilobytes using datasets with smaller dimensions. The number of training instances to build the model has no effect on the model size as we only store the extracted characteristics and dismiss the actual raw data. Only the number of attributes and the number of clusters or sub-clusters (within each cluster) would have an effect on the model size. $Opp - S_i$ is the data across all segments for $subject_i$ in the OPPORTUNITY dataset. Hence, the only difference among $Opp - S_i$ training datasets is the number of instances, the model size remains static regardless of the number of instances. For accelerometer-only datasets, the only difference between $WISDM_{6act}$ and $WISDM_{4act}$ is the number of classes inside. $WISDM_{4act}$ is a subset of $WISDM_{6act}$ with data representing only four activities. Each class has a set of sub-clusters inside. Therefore, a lesser

number of classes means fewer characteristics to be stored and consequently a smaller size for the learning model.

Table 4.2: Model Size and Training Data Characteristics

Training Data	N. Attributes	N.Instances	N.Classes	$BLFW$ Size (KBs)
$Opp - S_1$	110	134,613	4	102
$Opp - S_2$	110	133,023	4	102
$Opp - S_3$	110	124,320	4	102
$Opp - S_4$	110	105,082	4	102
$WISDM_{6act}$	3	32,262	6	9
$WISDM_{4act}$	3	166,226	4	4
$SPAD_{train}$	3	2,622	4	4
$PAMAP2_{user101}$	52	215,780	10	104
$PAMAP2_{user103}$	52	17,4338	7	89

In summary, the analysis of time and space complexity of STAR shows a real time and efficient performance for both mobile device and desktop deployment. We addressed the *computational efficiency* desired feature of STAR with the aforementioned analysis. In the following discussions, we analyse STAR performance for *flexible* and *accurate* recognition to achieve aims 1 and 2 for STAR evaluation.

Active and incremental learning efficiency

The key objective of the personalisation component is to achieve flexibility and accommodate changes in activities at a minimum cost. The personalisation component applies both incremental and active learning approaches to fulfil its goal. For building an efficient system, $ALRate$ should be at a minimum along the stream, to not bother the user with too many inquiries. At the same time, active learning has to operate along the stream with a balanced rate, to handle expected changes when necessary. As explained in the personalisation component section, the decision of active learning relies on the recognition decision. Therefore, we expect a higher rate of active learning with activities that are more challenging for recognition.

Figure 4.8 shows the active learning rate along the stream of activities in a subset of PAMAP2 dataset. In this experiment, the BLFW is built from data of user 107 and tested on data from user 104. The observation of the stream runs for about 30 minutes. In Figure 4.8, the above line (in red) displays the instances of the stream moving across the seven activities over time. The lower (blue) line represents the $ALRate$ along the stream of activities. We have noted some observations from this evaluation. Firstly, the number of inquiries is distributed among activities. Thus, active learning is uniformly triggered and therefore expected to be efficient in coping with changes whenever they occur.

Secondly, *ALRate* is reasonably small with the total around 7% of processed data. Thirdly, active learning is tightly related to the recognition decision and the rate of change of activities. Activities that are hard to recognise such as "Stairs" and "Vacuuming" are more likely to trigger active learning. With less confusing activities, the areas of plateau which represent a sequence of data representing the same activity are most likely to trigger less active learning inquiries as in "Lying" and "Sitting" in Figure 4.8(a). Also, *ALRate* may increase at the beginning of the activity in order to personalise the model. Eventually, *ALRate* decreases. This reflects the refinement of the model to the recent changes in the activity - as shown at the "Walking" activity in Figure 4.8(b). In contrast, *ALRate* increases along the stream with the high rate of activity change, especially in transitional periods.

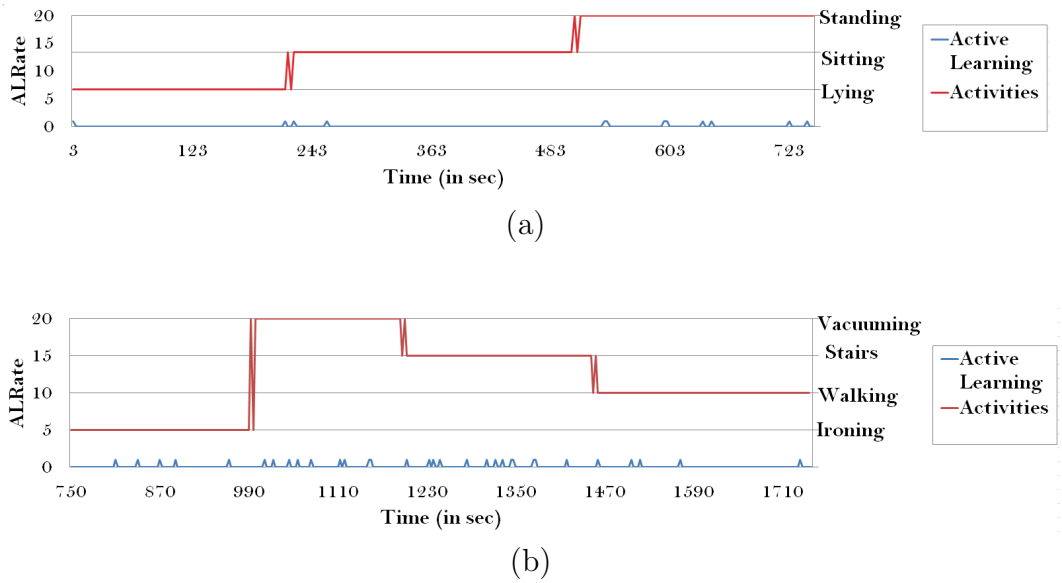
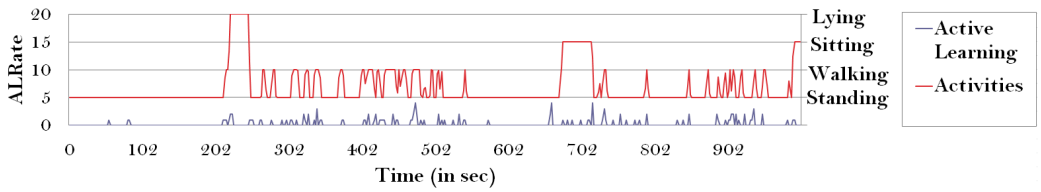


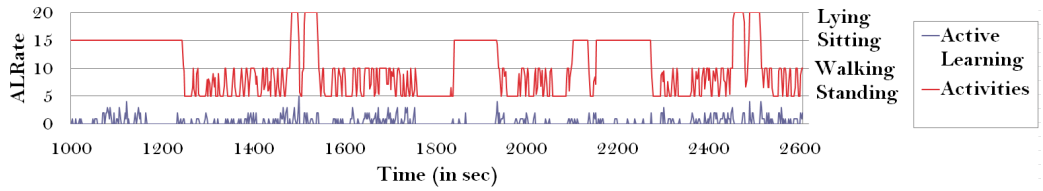
Figure 4.8: Active Learning Inquiries with the PAMAP2 Dataset

Figure 4.9 displays the behaviour of active learning with the OPPORTUNITY datasets. In this run, the BLFW is built from all segments of subject S4 (that contains noise). All data from subject S1 is applied for testing. Active learning with the OPPORTUNITY dataset is uniformly triggered along the stream. This dataset presents more challenges because it is highly fluctuating and contains noise. Both of these characteristics impact the performance of active learning. The BLFW that contains noise affects the recognition decisions for different activities. As the confusion increases in the recognition decision, the percentage of *ALRate* increases. Besides, the transitional activities and the frequent alternation between activities could be also common motives for frequent triggering of active learning in the OPPORTUNITY dataset. Thus, the technique could be improved by including a filtering step for recognising

the transitional activities at a higher level (prior to applying STAR). This filter would isolate data that represents the transitional activities. Therefore active learning will be triggered in case of real change. Another suggested approach is to postpone the label inquiring until the change becomes persistent. Therefore, it keeps the suspected data in short term memory until a real and recurrent change is detected and validated. Once the change is validated, it could be used to refine the model with the true labels.



(a)



(b)

Figure 4.9: Active Learning Inquiries with the OPPORTUNITY Dataset

In order to test the correlation between active and incremental learning rates, we run STAR while enabling only active learning (incremental learning is deactivated in this case) and evaluate the percentage of data that requires active learning. Then, we run STAR with both active and incremental learning enabled. Figure 4.10 shows the inverse relationship between the percentage of data used for incremental and active learning. The evaluation is carried for N different subsets of the OPPORTUNITY dataset. Each run has the same default setup of training on a single subject data and testing on a new one. The percentage of incremental learning varies from one run to another based on the nature of the data itself. Our focus in this figure is to display the effect of incremental learning on active learning, comparing the "solo-active" bar and "active with incremental" bar. The "solo-active" bar represents the fraction of data that triggers active learning with different runs while disabling incremental learning. On the other hand, the "active with incremental" bar displays the fraction of data that requires active learning when enabling both active and incremental learning in STAR. The above line is related only to the "active with incremental" bar. It represents the fraction of data that requires incremental learning for each run. The figure shows an average of

20–35% of the data is used to update the model with the incremental learning approach. The percentage of active learning data is much lower with an average of 8% of the processed data. We compare the behaviour of active learning with and without enabling the incremental learning. When the percentage of incremental learning is small, i.e. during the first 5 runs, the percentage of data that requires active learning when enabling incremental learning is more than or equal to the percentage of data for "solo-active".

The inverse happens as the incremental learning percentage increases. The percentage of active learning needed becomes lower with the introduction of incremental learning, once the percentage of incremental learning exceeds 27% in Figure 4.10. After this percentage, the percentage of active learning needed decreases, when incremental learning is enabled, compared to "solo-active". This applies for each individual run with an increasing amount of incremental learning.

In a nutshell, applying incremental learning has a positive impact on reducing the amount of data that requires active learning. Incremental learning has the advantage of updating the learning model with the selected data with no need for asking the user for true labels. As the percentage of incremental learning increases, the percentage of data requires active learning in contrast decreases. That means the system requires fewer inquiries for annotated data via active learning when incremental learning refines the model automatically.

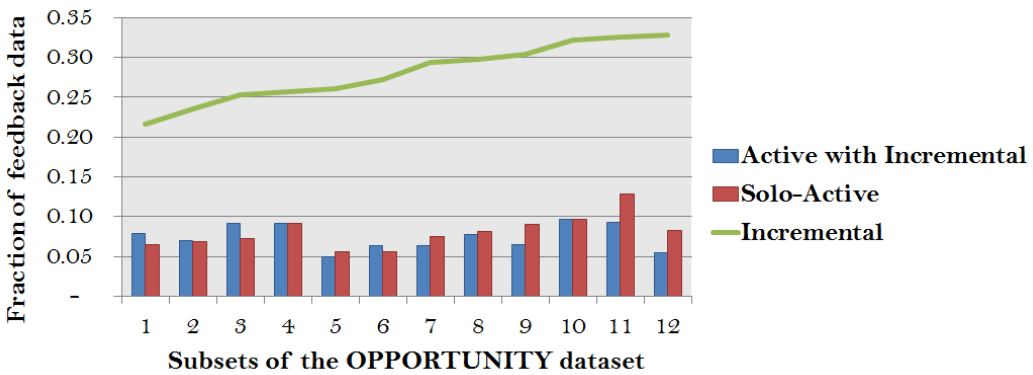


Figure 4.10: Impact of Incremental Learning on Active Learning Rate

The discussed analysis shows the performance of active and incremental learning with the evolving data streams. Our technique triggers a low rate of active learning with the PAMAP2 dataset despite its high sampling rate. However, other datasets, such as OPPORTUNITY, which contain very frequently changing data and many occurrences of transitional activities in addition to noise, require higher rates of active learning. The analysis also shows the positive impact of increasing the percentage of incremental learning on reducing

the percentage of active learning. This discussion leads us to the next section for evaluating the effect of incremental and active learning on the overall recognition accuracy of STAR.

Recognition accuracy

In this section, we evaluate STAR performance in terms of accuracy and flexibility with different datasets. Table 4.3 displays STAR measures across users in the OPPORTUNITY dataset. We build the BLFW from the data of one user and evaluate STAR with the other users' data. Each row represents the average measures of evaluation for particular subject data. The BLFW for each subject is built from data of other subjects. STAR is evaluated and compared to other static classification methods, which are C4.5 decision trees (DT), incremental Naive Bayes (NB_{inc}), and support vector machines (SVM). Both SVM and DT have been applied pervasively in the literature for activity recognition. NB_{inc} is also applied for personalisation in activity recognition in [GKG+12a]. We run the traditional classification methods with Weka 3.7 built-in techniques [WF05]. Despite the efficiency of these methods for activity recognition in a static environment, these techniques lack the ability to adapt with the evolving data streams. Also, traditional classification techniques preserve training data and allow many iterations on data for prediction. However, this is not realistic in streaming settings especially with deployment in a limited resources device such as a mobile phone. The reported results also display the personalisation effect separately for each activity. w is the percentage of activity instances in the data stream. The effectiveness of detecting concurrent activities in each window is noted by the $CPur$ percentage. $CPur$ is the average purity of all clusters generated along the stream. A high value of $CPur$ indicates well separated pure clusters and demonstrates the ability to recognise concurrent activities in data chunks.

As illustrated in Table 4.3, the overall accuracy of STAR in terms of the TP rate always outperforms other classification techniques. The average purity of clusters, $CPur$, is 99% across all runs. The high percentage indicates an efficient performance of STAR sliding window and online clustering in separating activities apart. STAR shows a significant accuracy enhancement (for more than 18%) for S4 that contains noise. The two activities of "Lying" and "Standing" are effectively recognised with STAR across all users. For the heavy weighted "Standing" class, STAR shows a stable accuracy - well above 80%. "Standing" is a challenging activity because it can easily overlap with other activities, such as "Walking". The "Lying" class is small - its weight is only 5% of the data and hence tends to be misclassified. For instance, DT

Table 4.3: STAR Performance with OPPORTUNITY

Test	*Technique*	*TP*	*Standing* w: 45%	*Walking* w: 26%	*Lying* w: 5%	*Sitting* w: 24%	*ACT*	*INC*
S_1	*STAR(%)*	**80.7**	**88.4**	56.5	**93.1**	52.1	7.0	28.1
	DT (%)	63.0	57.4	68.1	46.6	**65.5**	-	-
	SVM(%)	76.8	88.0	39.3	66.4	33.2	-	-
	NB_{inc}*(%)*	67.3	58.7	**88.9**	57.6	**65.5**	-	-
S_2	*STAR(%)*	**74.9**	**82.5**	50.1	**89.9**	47.3	6.8	26.9
	DT(%)	54.5	49.1	70.0	39.2	48.2	-	-
	SVM(%)	71.8	71.5	56.4	60.6	**87.1**	-	-
	NB_{inc}*(%)*	63.0	60.9	**82.2**	16.1	55.0	-	-
S_3	*STAR(%)*	**68.6**	**87.0**	44.3	**86.1**	7.2	7.5	25.5
	DT(%)	56.9	76.7	56.7	0	40.0	–	-
	SVM(%)	64.7	65.3	57.4	27.7	**56.2**	-	-
	NB_{inc}*(%)*	62.4	63.4	**87.8**	0	45.1	-	-
S_4	*STAR(%)*	**71.1**	**84.2**	43.6	**90.0**	30.3	8.4	31.7
	DT(%)	43.4	30.1	70.0	0	50.2	-	-
	SVM(%)	52.9	71.4	13.7	69.1	**59.6**	-	-
	NB_{inc}*(%)*	31.9	0	**100**	0.7	20.7	-	-

and NB_{inc} failed to detect almost all occurrences of the "Lying" class in both S3 and S4. However, STAR could accurately recognise the "Lying" activity because of the efficient learning. The percentage of data used for incremental and active learning is displayed for each subject. The percentage of data used for active learning (ACT) is on average of 7%. The percentage is still as small as 8.4% with the noisy data in S4. The small percentage indicates the efficiency of the developed active learning approach to select only a small amount of the most profitable data to label. On the other hand, as the data of incremental learning does not require user input, the higher percentage is better given that it improves the overall accuracy. The results demonstrate the efficiency of STAR in terms of *robustness*,*accuracy*, and *flexibility* compared to other techniques.

We also evaluate STAR on the WISDM dataset. From data collected in the WISDM lab, users 36 and 27 have performed the five activities. To show the personalisation impact on accuracy, we train the model on one of them and test on the other. The sub-clusters structure of BLFW tends to aggregate the same activity with different patterns together. Therefore, we combine the "Upstairs" and "Downstairs" activities into a single "Stairs" class with sub-clusters inside. Table 4.4 shows the performance of STAR compared to traditional classification techniques. The average purity for clusters ($CPur$) is 99.9% which demonstrates efficient separation of concurrent and interleaved

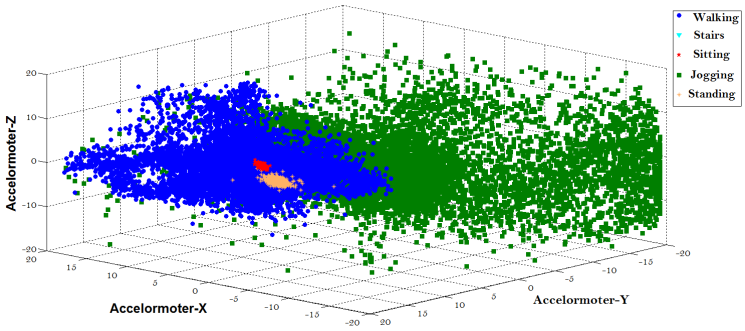
activities. Figure 4.11 shows a snapshot of activities in the WISDM dataset. In Figure 4.11(a), we hide the occurrences of the "Stairs" class for better visibility to other classes. As shown in the graph, the "Sitting" and "Standing" classes are small yet dense clusters. The small sized classes of "Sitting" and "Standing" are totally misclassified with other methods, while the STAR recognition rate is up to 99% for same classes. The figure depicts also the overlapping between the "Stairs" class and all other classes that results in poor recognition accuracy for the "Stairs" class. STAR still outperforms other techniques in recognising "Stairs" activity with about 20% enhancement in accuracy. The personalisation effect is clearly demonstrated in the "Walking" activity with more than 25% improvement on accuracy compared to other techniques. It is also noted, the percentage of data for incremental learning is 19.8%, while the percentage of data for active learning is 16.5%. The "Jogging" activity is the one that most triggers active learning, followed by "Walking", then "Stairs".

Table 4.4: STAR Performance with WISDM

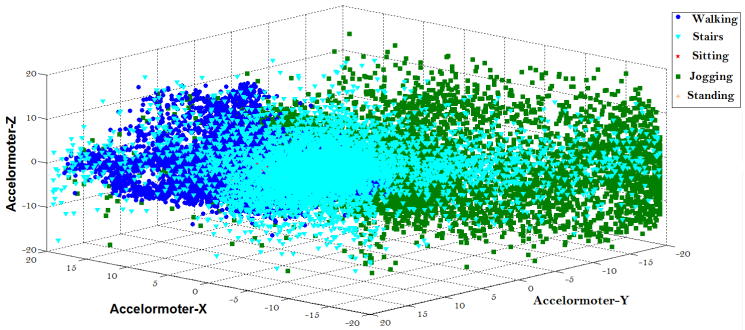
Technique	*TP*	*Walking* w: 35.7%	*Jogging* w: 34.4%	*Sitting* w: 6%	*Standing* w: 4.6%	*Stairs* w:19.2%
STAR (%)	**71.2**	**80.4**	65.0	**99.1**	**99.6**	**49.2**
DT(%)	41.4	24.6	86.7	0	0	13.6
SVM(%)	44.3	58.5	52.7	0	0	27.7
NB_{inc}	42.8	20.5	**94.7**	0	0	15.5

We also evaluate STAR on the SPAD mobile sensor dataset which is collected from an accelerometer sensor. Table 4.5 shows that STAR outperforms all other techniques in terms of prediction accuracy. Because of its small size, "Run" is the most challenging activity in this dataset as it spans a large spatial area and could be confused with the "Walk" activity. Although the "Drive" activity is small too, its corresponding recognition rate is high because it is well-distinguished from other classes/activities as shown in Figure 4.12.

Lastly, we evaluate STAR performance on the PAMAP2 dataset. This dataset contains a larger set of ten different activities performed by 8 users. The dataset also is remarkable for its high sampling rate. In all experiments, we build the BLFW with data from only one user and evaluate the performance with data from other users. We noted the average purity of the clusters is more than 99%. We compare the performance of STAR with SVM, Random Forests, and incremental Naive Bayes. The Random Forests technique is an ensemble learning method that uses multiple decision trees. Random Forests in general



(a) Excluding stairs



(b) Including stairs

Figure 4.11: Snapshot of the the WISDM Dataset

shows better accuracy with problems that has large number of classes [KS13, AGPV13]. Therefore, in this experiment, we apply Random Forests instead of C4.5 decision trees. We aggregate sub-clusters that belong to the same general activity in one class with sub-clusters. "Upstairs" and "Downstairs" activities are combined in a single "Stairs" class with sub-clusters. The same aggregation concept applies for different patterns of the "Walking" activity.

The average percentage of active learning (ACT) for all runs is 15%, while the average incremental learning percentage is 21%. The amount of active learning explains the relationship between the accuracy of recognition and active learning triggers. For instance, with $User_{101}$ the accuracy is less than 50%; the active learning percentage is 21%. On the other hand, with $User_{103}$ there is a better accuracy of 69%, and the active learning rate drops to only 7.5%. Thus, more confusion in recognising activities triggers a higher percentage of active learning. By applying active and incremental learning for personalisation, STAR outperforms all other techniques for all users.

Table 4.6 shows the performance of STAR and other classification techniques. In the dataset, users perform different sets of activities. NA in the table corresponds to activities that are not performed by this user (or not in

Table 4.5: STAR Performance with SPAD

Technique	*TP*	*Walk* w: 60.6%	*Run* w: 12.4 %	*Still* w: 20%	*Drive* w: 7%
STAR (%)	**97.2**	**100**	**77.9**	**100**	99.1
DT (%)	95.4	**100**	65.7	**100**	94.8
SVM (%)	68.3	**100**	5.0	0	**100**
NB_{inc}	89.9	**100**	35.5	**100**	70.7

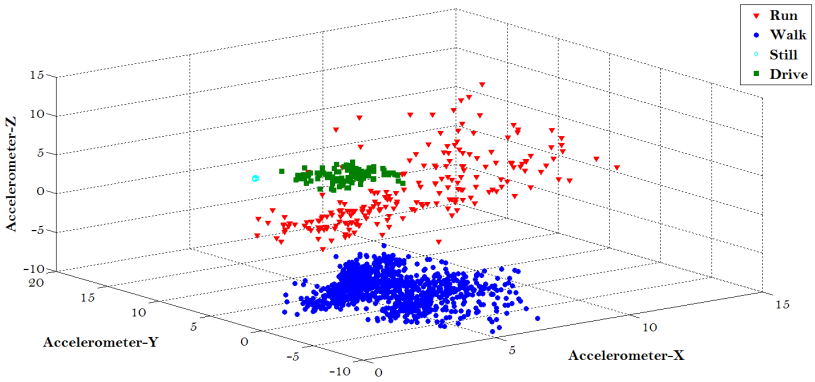


Figure 4.12: Snapshot of the SPAD Dataset

the BLFW). STAR showed a stable accuracy across all activities, except the "Jump" activity. The reason of STAR's failure to detect the "Jump" activity is its very small weight (0.1%) that does not allow time for applying incremental or active learning to achieve better recognition. STAR showed its best accuracy in personalising and recognising accurately in the "Iron", "Stand", and 'Sit" activities for almost all runs. STAR still performs well for an average of (90%) for the "Lying" activity. The activities of "Vacuum", "Walk", and "Stairs" are the most challenging for STAR to recognise. The advantage that enables STAR to give a better performance than other methods is its stability and reasonable accuracy in recognising across different activities. Although, some of the other methods can recognise a subset of the activities accurately, they completely fail to recognise other activities. For instance, the SVM for $User_{103}$ that successfully recognises the "Vacuum" activity with 100% accuracy fails completely to recognise "Stand" and "Sit" and most occurrences of "Iron" and "Stairs". In sum, STAR shows an accurate and flexible recognition of activities in the PAMAP2 dataset that has many activities and a high sampling rate.

The evaluation of STAR on different datasets shows an improved accuracy of STAR over well-known techniques applied for activity recognition. The impact of personalisation in STAR is highlighted in recognising small sized activities, that are not well represented in the stream. These activities are mostly misclassified by other classification techniques. However, STAR could successfully recognise these activities with an accuracy more than 90%, such as "Lying" in OPPORTUNITY, and "Standing" and "Sitting" in WISDM. STAR also shows its robustness in recognising activities from data containing noise with more than 15% accuracy enhancement. Lastly, the evaluation of STAR's accuracy on the PAMAP2 dataset, which is of high sampling rate and contains many activities, depicts the efficiency of the incremental learning approach in STAR. Across all experiments, STAR outperforms other techniques in the overall accuracy. The incremental approach demonstrates a stable recognition accuracy for different activities across users.

In conclusion, the experiments reported in this section show the performance of our developed technique in terms of accuracy, robustness, flexibility, and computational efficiency. We evaluate the technique performance for personalised recognition in streaming environments and on limited resource devices. We also demonstrate the efficiency of active and incremental learning approaches for enhancing the recognition performance.

4.11 Summary

Activity recognition is an important aspect in the area of pervasive computing, especially when dealing with non-stationary sensory data. Activities are evolving and typically changing from one user to another. Thus, building a personalisation technique that can effectively tune the recognition model for a specific user is crucial to enhance the recognition performance. The streaming environment of processing sensory data in activity recognition imposes more challenges for personalisation especially for labelling cost and dealing with high speed data at real time. In this chapter, we proposed, developed, and evaluated our technique for personalised activity recognition from evolving data streams (STAR). The developed technique applies incremental learning to refine the recognition model continuously with recent changes in the evolving activities. It also applies active learning for addressing the scarcity of annotated data by labelling only a small amount of data. Our technique contains three components of modelling, recognition and personalisation across two phases of offline and online learning. The modelling component builds the initial baseline framework from historical data. The recognition component integrates

the baseline framework with an ensemble classifier for recognising the incoming data stream. While the personalisation component updates the baseline framework continuously through batch active and incremental learning.

We conducted our analysis on four benchmarked datasets in activity recognition. The performance of STAR is evaluated according to its accuracy, robustness, flexibility, and efficiency. STAR is deployed on a mobile phone to demonstrate its efficiency to run on a limited resources device at real time or near real time even with a high sampling rate. The performance of active and incremental learning is also evaluated. The analysis shows the behaviour of active learning inquiries along the stream, and links this behaviour to the recognition accuracy and the evolution of activities. The positive impact of incremental learning in reducing the percentage of active learning is demonstrated in the discussions. Finally, we show the substantial improvement in the recognition accuracy for STAR compared to other state-of-the-art techniques in activity recognition. The results demonstrate the personalisation effect on each activity and shows that STAR has accurate recognition, especially for activities that are dense and small which other techniques tend to completely misclassify. STAR also shows good performance in handling multi-dimensional, multi-activities, and noisy data.

As it is important to refine the recognition model for changes in existing/known activities by personalisation, further development is required for detecting novel/unknown activities that have not been seen by the recognition system. In the next chapter, we apply the concept of continuous learning for dynamic adaptation of the recognition model for detecting entirely new activities and forgetting abandoned ones.

Table 4.6: STAR Performance with PAMAP2

Test	*Technique*	*TP*	*Lying* w=11%	*Sit* w=11 %	*Stand* w= 11%	*Iron* w=14%	*Vacuum* w= 10%	*Stairs* w=13%	*Walk* w=24%	*Cycle* w=10%	*Run* w=5%	*Jump* w= 0.1%
$User_{101}$	*STAR(%)*	**49.5**	84.5	**34.8**	0	**83.3**	49.6	**16.6**	66.5	**48.0**	NA	NA
	RF(%)	25.1	0	0	**0.2**	42.7	**99.0**	0	3.9	1.0	NA	NA
	SVM(%)	22.8	**99.0**	0	0	32.9	61.7	0	0.8	0	NA	NA
	$NB_{inc}(\%)$	25.9	0	0	0	0	57.9	0.1	**99.9**	0	NA	NA
$User_{102}$	*STAR(%)*	**66.9**	89.4	**88.9**	**85.0**	**96.8**	47.0	39.3	50.3	NA	NA	NA
	RF(%)	38.0	**90.0**	0	0	9.3	87.3	41.3	42.8	NA	NA	NA
	SVM(%)	17.0	0	0	0	0	**100**	**49.2**	0	NA	NA	NA
	$NB_{inc}(\%)$	23.7	0	0	0	0	0	45.1	**58.6**	NA	NA	NA
$User_{103}$	*STAR(%)*	**69.0**	92.5	**94.8**	**55.6**	**77.2**	38.3	**28.1**	85.0	NA	NA	NA
	RF(%)	39.2	90.5	0	0	50.3	85.6	10.8	48.9	NA	NA	NA
	SVM(%)	34.2	**95.7**	0	0	0.2	**100**	0.4	62.0	NA	NA	NA
	$NB_{inc}(\%)$	25.2	67.7	0	0	0	0	0.1	**100**	NA	NA	NA
$User_{104}$	*STAR(%)*	**65.9**	94.0	**74.5**	**40.7**	**93.3**	**45.7**	**30.1**	70.5	79.7	NA	NA
	RF(%)	48.6	96.4	31.0	39.4	0.4	4.5	0.1	83.1	97.9	NA	NA
	SVM(%)	28.6	**96.8**	1.2	0	0	3.5	0	33.8	**100**	NA	NA
	$NB_{inc}(\%)$	40.1	59.7	0	0	0	0	0	**96.4**	95.8	NA	NA
$User_{105}$	*STAR(%)*	**56.2**	**92.3**	**77.2**	**48.7**	**66.8**	19.7	**25.8**	62.4	**45.6**	NA	NA
	RF(%)	12.5	0	0.4	21.2	0.2	99.6	3.1	0	0	NA	NA
	SVM(%)	10.2	0	0	0	0	**100**	0	0	0	NA	NA
	$NB_{inc}(\%)$	24.3	0	0	0	0	0	0	**100**	0	NA	NA
$User_{106}$	*STAR(%)*	**61.6**	**94.9**	**91.4**	**36.5**	**70.1**	**39.2**	**40.6**	**46.0**	87.9	67.0	0
	RF(%)	28.8	93.6	10.2	17.0	45.8	2.7	0.3	3.6	5.1	99.5	0
	SVM(%)	13.1	38.2	0	4.0	0.1	0	0	0	0	**100**	0
	$NB_{inc}(\%)$	16.2	0	0	0	0	0	3.6	5.0	**99.3**	72.9	0
$User_{107}$	*STAR(%)*	**62.3**	**94.7**	**94.0**	**66.3**	**70.5**	**28.2**	34.8	**54.1**	87.1	41.0	NA
	RF(%)	44.4	94.2	0	0	0	0	84.3	49.0	**92.2**	83.3	NA
	SVM(%)	14.2	0	0	0	0	0	**100**	0.1	0	**100**	NA
	$NB_{inc}(\%)$	22.1	5.4	0	0	0	0	98.9	10.0	63.9	10.4	NA
$User_{108}$	*STAR(%)*	**47.9**	**93.7**	**91.4**	**15.0**	0	24.5	**61.8**	**57.5**	NA	NA	NA
	RF(%)	20.9	77.3	0	0	0	94.7	4.9	2.5	NA	NA	NA
	SVM(%)	11.5	0	0	0	0	**100**	0	0.2	NA	NA	NA
	$NB_{inc}(\%)$	8.9	0	0	0	**0.1**	0	0.4	31.0	NA	NA	NA

Chapter 5

Concept Evolution Technique for Activity Recognition in Evolving Data Streams

5.1 Introduction

Our main target in this dissertation is to build efficient techniques for activity recognition that enable personalisation and adaptation with evolving data streams. We proposed, developed, and evaluated STAR in Chapter 4. STAR enables personalisation in activity recognition through incremental and active learning. In this chapter, we present our technique for concept evolution and adaptation through a continuous and active learning approach. While STAR refines and tunes the learning model for personalised recognition of existing activities, the technique developed in this chapter aims to detect and incorporate completely new activities appear in the evolving data streams. Model adaptation includes also pruning the model by detaching activities that are no longer relevant.

Activity recognition typically deals with data streams that evolve from different sources such as on-body sensors, environmental sensors, and mobile sensors. It is impractical in activity recognition to assume that the number of activities is static along the stream. Dynamic changes in data streams that reflect variation in users' activities are expected and natural. Thus, novel activities may emerge at any time in the stream and old activities may have abandoned. State-of-the-art activity recognition techniques rely strongly on prior knowledge. The learning model in activity recognition is built from historical annotated data that contains a specific set of activities. The set of

predefined activities is assumed to be fixed in the traditional activity recognition approach. However, dealing with evolving data streams for recognising activities and across different users emerges the need for a dynamic learning model that adapts continuously. Thus, the dynamic model would allow model extension by adding newly detected activities or pruning by removing irrelevant and outdated activities. Furthermore, the scarcity of annotated data restricts the ability to retrain the learning model for each user with all possible activities. Therefore, learning continuously and incrementally for detecting novel activities or forgetting abandoned activities then incorporating them into the learning model became essential for boosting the recognition performance.

As discussed in Chapter 2, detecting novel activities in data streams overlaps the two areas of research for both activity recognition and mining data streams. From the activity recognition perspective, traditional classificatory models deployed for recognising activities are unable to detect novel activities and consequently misclassify all instances representing novelties. In order to cope with the evolving activities, the recognition technique has to be able to handle the appearance of a novel concept as soon as it arrives without being trained with labelled data. In data stream mining research, the essential issue of "concept evolution" in data streams refers to the detection of the appearance of a novel class while a stream evolves. Few studies addressed the concept evolution issue in stream mining such as in [SCG07] and [MAKK$^+$11]. However, none of the concept evolution techniques in stream mining are customised for detecting novel activities for activity recognition.

Any efficient streaming technique for detecting novel concepts and forgetting abandoned ones in activity recognition has to essentially consider the nature of activity recognition data. Data streams in activity recognition have two main characteristics of not identically distributed and not independent (violating both i.i.d conditions). Activity recognition data is not independent as people perform activities in a sequential manner (i.e., performing one activity followed by another). Therefore, data streams in activity recognition are typically comprised of sequences of data that represent sequences of activities. A stream of data is also non-identically distributed because of the anticipated changes in existing activities over time in addition to the possible appearance/disappearance of activities. Dealing with non-independent and non-identically distributed data when detecting changes in activities is a challenging task. The detection of novel concepts in activity recognition includes both the detection of abnormal activities (outliers) such as a "Sudden fall" and also the detection of new normal activities such as "Driving" (when the "Driving" activity did not exist in the training data).

In this chapter, we propose, develop, and evaluate our novel technique for **C**oncept ev**O**lution in data **ST**reams for **A**ctivity **R**ecognition - termed **COSTAR**. The term concept is often used interchangeably in this chapter with the term activity. We mean by concept evolution both detection of novel concepts and also forgetting outdated ones. Our technique applies continuous and active learning to handle the appearance/disappearance of activities in data streams. COSTAR has the merit of considering both the dependency and distribution of data in the stream of activities. The idea of COSTAR is to monitor the movement of data along the evolving stream over time. The sequence of data that moves away from all existing concepts and at the same time forms a stable and separate cluster is expected to be novel. Moreover, COSTAR dynamically adapts the learning model by incorporating detected novel activities or removing outdated ones. Therefore, the adapted learning model is capable of recognising any recurring occurrences of the novel activities without any manual intervention.

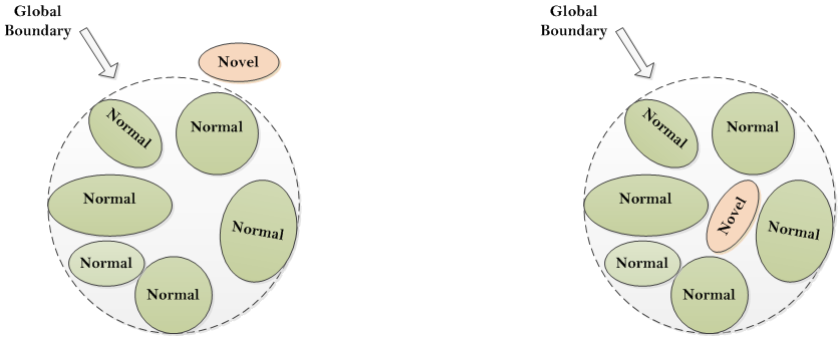
The rest of this chapter is organised as follows. We first define the meaning of novel concept in the context of activity recognition. Then, we begin with providing an overview of COSTAR. Assumptions and formalisation of the problem are explained next. Then, we discuss the implementation details of COSTAR framework, components and methodology. The contribution of COSTAR is discussed and highlighted. We evaluate and analyse COSTAR on benchmarked datasets later in this chapter. Finally, the chapter is concluded with a summary.

5.2 Novel Concepts in Activity Recognition

The notion of a novel concept in the context of activity recognition can be described in two major categories of *abnormal* and *novel normal* activities. In our developed technique, we aim to detect novel concepts in both categories. Most of the existing techniques in concept evolution for stream mining target the detection of *abnormal* instances/concept in data streams [AF07, AKBS12, PLL07, Agg13]. Abnormality detection is applicable in activity recognition for detecting abrupt activities such as sudden fall or malicious actions.

Other techniques have been proposed to capture *novel normal* concepts in data streams. There are two approaches for detecting normal novel concepts. The difference between the two approaches is the perspective of the underlying learning model. The single-class model approach has been proposed to detect novel concepts, such as in [SCG07]. The learning model in this approach is represented by a single concept and all incoming data is either part of the

underlying learning model or novel. This approach assumes that there is only one "normal" concept and any other different concepts are "novel". Multi-class model is the other approach that considers the existence of more than a single concept in the underlying learning model. ECSMiner [MGK+11] and MINAS [FGC13] are stream mining algorithms for novelty detection that use the multi-class model approach. In these algorithms, they define global boundary as the union of underlying concepts' boundaries. The novel concept appears outside the global boundary as shown in Figure 5.1(a) and must satisfy certain separation and cohesion criteria.



(a)Novel Concept Outside the Global Boundary

(b) Novel Concept Inside the Global Boundary

Figure 5.1: Explanation of the Novel Concept Positions

Both approaches detect the novel normal concept that appears outside the global boundary. However, this assumption is not always true. Precisely, in activity recognition, novel normal activities might appear in the middle of other normal concepts/activities. Figure 5.1(b) illustrates the appearance of novel cluster spatially inside the global boundary and outside the local' boundaries of existing concepts. For example, the learning model that represents the basic activities of "Running", "Walking", "Sitting", and "Standing" may be extended to accommodate for the novel activities that were recently performed by the user. The "Exercise" activity, for instance, is a novel activity that might appear in the middle of the underlying defined concepts of "Standing", "Running" and "Walking". Another example is the "Driving" activity that might appear in the middle of "Sitting" and "Running" activity clusters. The appearance of a novel and yet normal concept is expected and naturally occurring in activity recognition. However, it is not considered with any of the techniques represented in the literature as far as we know for novelty detection and concept evolution in either stream mining or activity recognition. In our technique, we target the detection of novel concepts regardless of their positions from the global boundary (that includes abnormal and novel normal concepts).

5.3 COSTAR Overview

We start in this section introducing our novel technique, COSTAR. In terms of the learning components, COSTAR consists of two main components: modelling and adaptation. If we look from a learning paradigm perspective, COSTAR is divided into two phases: the offline phase where the modelling component operates; the online phase that implements the adaptation component.

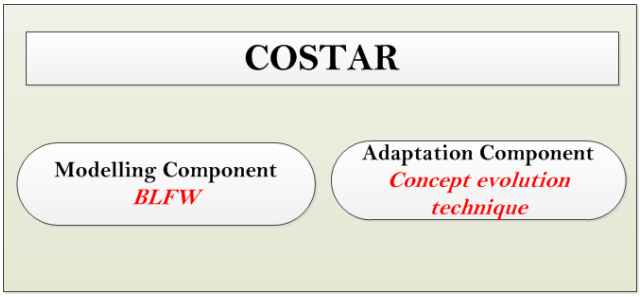


Figure 5.2: Components of COSTAR

In the offline modelling phase, COSTAR builds the baseline framework (BLFW) from historical data with the "clustering with sub-clusters" approach as discussed in Chapter 3. The output of this first phase of modelling is a fine-grained BLFW that represents activities that exist in the training data. The BLFW is a "multi-class" model that can distinguish between different clusters in existing data as well as sub-clusters within each cluster. In the online phase, we introduce our new adaptation component that implements a continuous and active learning from evolving streams for detecting novel concepts and forgetting outdated ones. Moreover, the adaptation component modifies the BLFW to accommodate for the detected changes at real time. Thus, COSTAR extends the BLFW by adding the novel activity and/or removing abandoned activities. The assimilated BLFW is capable of detecting the recurring occurrences of the novel activities. After explaining COSTAR's main components, we then describe assumptions and formalisations of the problem and introduce COSTAR in more detail.

5.3.1 Assumptions and formalisations

The formalisation of the data stream has been presented in Figure 4.2, Chapter 4. In this chapter, we extend the same formalisations to present different notations in COSTAR. Data streams arrive at high speed and require real or near real time processing. We assume that the nth chunk in the stream arrives at the nth time stamp. In traditional stream mining techniques, it is assumed that the delay in data classification is 0, which means data is classified as soon

as it arrives. In COSTAR, while a concept evolves, decisions may need to be delayed until enough data arrives. The more data we have, the more likely we reach a decision with sufficient information. The collected data is kept in a *Buffer* till either reaching a decision or reaching the limit of the *Buffer* capacity. In other words, the maximum allowed time for reaching the decision is the time required to reach the *Buffer* maximum capacity. The decision made is based upon any available information in this case.

Figure 5.3 illustrates different time snapshots along a data stream. A stream of data is segmented into n chunks of equal size. COSTAR starts at time t_0, when the *Buffer* is empty. As stream evolve, a new data chunk arrives. Therefore, the *Buffer* is accumulated with instances of the new data chunks. The algorithm tries to make a decision about data in the *Buffer* whenever a new chunk arrives. A declared decision is one of three possibilities: existing, novel, or unknown. The *Buffer* data might have been declared as part of an existing cluster (not novel), while a novel concept is declared when the *Buffer* data satisfies all conditions of a novel concept. An unknown decision is declared when a high level of uncertainty occurs between the two decisions of existing and novel. t_{dc} is the time when reaching any of the three decisions. At this time, we reset the *Buffer* and start processing new chunks of data.

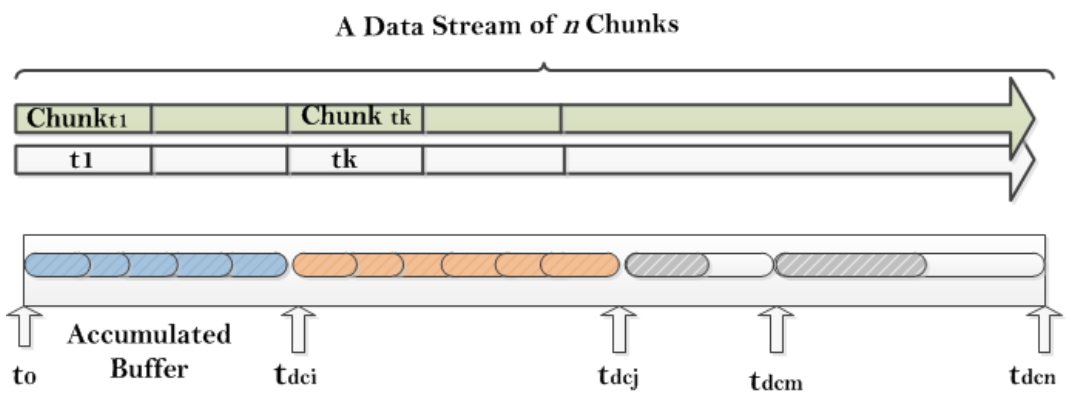


Figure 5.3: Time Snapshots Along the Data Stream

COSTAR handles three types of data repositories across various time stamps. For example, in Figure 5.3, assume the latest *Buffer* data is declared at time t_{dc_i}. After the time of declaration, the three data repositories are:

- *Data Chunk* ($Chunk_{t_k}$): Chunk of a data stream that arrives at time t_k. Where $t_{dc_i} < t_k$, which means the chunk arrives after the latest declared *Buffer* data.

- *Buffer*: Temporarily short memory to accumulate continuous undeclared chunks of data. It starts to accumulate from the last declared data at

t_{dc_i}. The *Buffer* is accumulated until either a decision or capacity limit is reached at time t_{dc_j}, where $t_{dc_i} < t_{dc_j}$. For each chunk of data, we check first if there is sufficient information to declare data with no need to keep it in the *Buffer*. Otherwise, all undeclared data between t_{dc_i} and t_{dc_j} is kept in the *Buffer*. Thus, Buffer data is either the entire data or a subset of the data that is received between t_{dc_i} and t_{dc_j}.

- *Just predicted* (JP): Very short memory that keeps a reference of the last declared data at time t_{dc_i}. This short memory is released once checked at the following time stamp, when the next data chunk arrives.

Based on the defined concepts and assumptions, we then discuss the conceptual framework of COSTAR and its components.

5.4 COSTAR Conceptual Framework

The process in COSTAR starts with building the BLFW offline from annotated collected data representing different activities. The model consists of m clusters built in feature space that correspond to m activities existing in the training data. Each cluster consists of a number of sub-clusters that represents patterns inside a particular activity/cluster. At the online phase, COSTAR receives the unlabelled sensory data of performed activities.

Figure 5.4 illustrates the conceptual framework of COSTAR with its phases and components. The modelling component in COSTAR builds the BLFW of clusters with sub-clusters. We extract a summary of the statistics. Then we dismiss all raw data instances when concluding the modelling phase. Cluster /sub-cluster characteristics are the extracted information that describes each cluster/sub-cluster and distinguishes it from others. Extracted characteristics span across three levels of a description. These levels are: sub-cluster characteristics, cluster characteristics, and holistic characteristics for the entire baseline model. Table 5.1 summarises the most commonly used characteristics for describing the baseline learning model.

The adaptation component integrates the BLFW and analyses chunks of data streams, in order to reach a decision. There are two key units in the adaptation component: the Cohesion Validation Component (CVC) and the OBSERVER as in Figure 5.4. The two units interact together to analyse data chunks for reaching a decision. The CVC checks and validates the dependency between the new incoming data chunk and the most recent declared data and/or *Buffer* data. The merit of the CVC is to aggregate dependent data that represents the same activity together, thus it enhances the OBSERVER

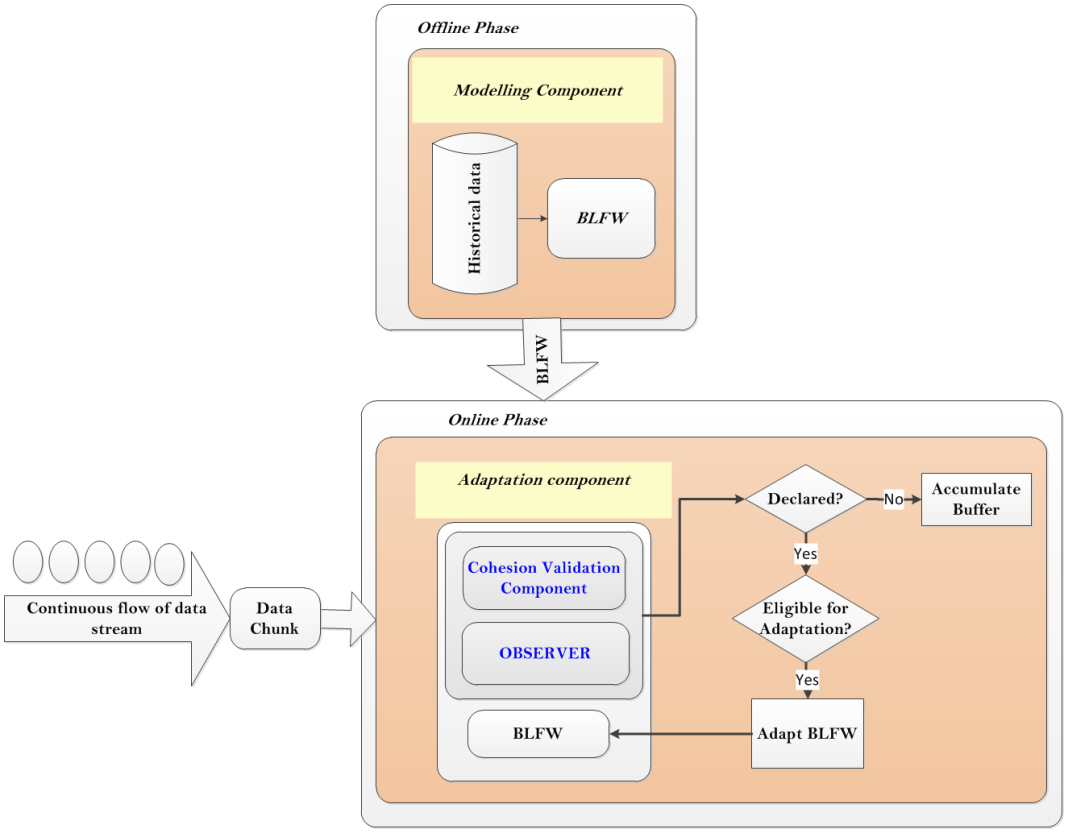


Figure 5.4: COSTAR Conceptual Framework

capability to reach a decision about the aggregated data. The OBSERVER monitors the evolution of data along the stream. Thus, it checks the movement, separation, and stability of the continuously aggregated data that is accumulated in the *Buffer*.

The OBSERVER makes a decision about the data in the *Buffer* whenever sufficient information is available. Three decisions are declared by COSTAR: novel activity, existing activity, or unknown. The unknown decision includes many categories such as noise data, outliers, or confusing instances. When a decision is made, the data in the *Buffer* is released and checked for eligibility to adapt the BLFW. Then, the adapted model replaces the outdated one for future recognition of recurring occurrences of the detected novel concept or other novel concepts. Otherwise, if there are not enough pieces of evidence to declare any decision, a data chunk is stored in short term memory - *Buffer* - until collecting sufficient data for making a decision or reaching the *Buffer* size threshold.

The online phase of adaptation in COSTAR applies a cluster-based approach for processing data chunks. In other words, the processing unit in COSTAR is the entire chunk instead of each instance in the chunk. In the context of activity recognition, a cluster-based approach is more relevant to

Table 5.1: Commonly Used BLFW Characteristics

Symbol	Description	Level
$Centroid_{sc_i}$	The mean of n dimensional points inside sc	Sub-cluster
$Weight_{sc_i}$	Total number of data samples belong to sc	
$Radius_{sc_i}$	The maximum distance within a sub-cluster between $Centroid_{sc}$ and any data sample belonging to sc	
$Nsub_{C_j}$	Number of sub-clusters inside the cluster	Cluster
$Centroid_{C_j}$	C_j centroid which is the mean of all sub-clusters centroids belonging to this cluster	
$Weight_{C_j}$	Total number of data instances inside the cluster	
$Radius_{C_j}$	Max distance between $Centroid_{C_j}$ and all data instances inside the cluster	
$Dmax_{C_j}$	Max distance between any pair of sub-clusters	
$Nclus$	Number of clusters/activities in the data domain	Holistic
$Centroid_{glob}$	Global centroid of all instances in the training data	
$Capacity$	Total number of training data instances	

the nature of activity recognition data where a sequence of data chunks representing a sequence of performed activities evolves over time. People perform activities in a sequential manner (i.e., performing one activity followed by another). Therefore, an activity recognition data stream is typically a sequence of chunks representing sequential activities. Instead of processing each data instance, which is costly in the streaming environment, our approach is a cluster-based technique that deals with each chunk of data as a cluster instead

of processing each single point in the stream. The majority of same-class instances inside the cluster guide the decision in COSTAR. The motivation for applying a cluster-based approach is as follows: (i) processing data as clusters is coherent with the nature of an activity recognition stream that originally consists of a sequence of chunks of activities. (ii) the cluster-based approach plays a vital role in preserving crucial device resources that are genuinely limited, especially when dealing with unbounded very fast sensory data streams. The conceptual framework describes the overall process of COSTAR. We then present more details about the COSTAR algorithm and steps in the next section.

5.5 Algorithm Outline

COSTAR is a cluster-based approach for monitoring the evolving activities, in order to detect the evolution of concepts in the stream of data. Concept evolution in an activity data stream refers to detecting novel activities and forgetting abandoned ones. The adaptation component in COSTAR monitors the movement of dependent data while the stream evolves. When data moves *closer* to or falls within any BLFW clusters boundaries, then it is expected to belong to an existing cluster. On the other hand, data is suspected to be novel when it moves *away* from all existing BLFW underlying concepts/clusters. The suspected concept is declared novel whenever it satisfies the required *stability conditions*. Our technique assimilates the detected novel concept into the BLFW, which enables future better classification and detection of recurring novel concepts and moreover detecting other novel concepts if any.

Algorithm 5.1 outlines a summary of COSTAR adaptation component in the online phase. The stream of data is divided into fixed sized chunks that are continuously provided as inputs to the concept evolution algorithm. The *Buffer* contains *undeclared* data from the most recent previous chunks. While just predicted data (JP) is the most recent *declared* data. Data chunk ($Chunk_{t_n}$), *Buffer*, and JP represent data in three time spectra. $Chunk_{t_n}$ is the chunk of the data stream that is received at time t_n. The *Buffer*, at time t_n, (if not empty) contains accumulated undeclared data up to time t_n. When a concept is declared, the data in the *Buffer* is released while we keep a copy of the data in JP. Therefore, either JP or the *Buffer* is empty in most cases (unless the case of outliers that will be discussed later in the CVC section). JP (if not empty) contains data that is just declared at time t_n-1. The three data repositories are processed initially through the Cohesion Validation Component (CVC) (line 1). The key aim of the CVC is to check the dependency between $Chunk_{t_n}$

and the *Buffer*, or $Chunk_{t_n}$ and JP if the *Buffer* is empty. the CVC reforms the data in the *Buffer* based on a dependency test. Thus, the reformed buffer is monitored by the OBSERVER based on the underlying concepts in the BLFW (line 2). The OBSERVER accumulates the *Buffer* (line 3) and either declares a decision (line 4) or only waits for processing the next data chunk (at time $t_n + 1$) (line 12). Upon reaching a decision, the data in the *Buffer* is declared and set as Just Predicted data (JP) (line 5). The algorithm checks the validity of data in JP for model adaptation (line 6). The BLFW is adapted with the JP data if eligible and replaces the outdated BLFW for processing the following data chunks (line 7).

Algorithm 5.1 COSTAR Online Phase

Input: $Chunk_{t_n}$: data chunk at time t_n
Buffer: short time memory of previous undeclared data
JP: most recent declared data
BLFW: the most updated baseline framework
Output: updated $BLFW$, declared data if any
1: *Buffer* ←**CohesionValidationComponent**($Chunk_{t_n}$, *Buffer*, JP)
2: *declared* ← **OBSERVER**($Chunk_{t_n}$, *Buffer*, BLFW)
3: *Buffer* ← **AccumulateBuffer**($Chunk_{t_n}$, *Buffer*)
4: **if** declared **then**
5: JP ← SetAsJustPredicted(*Buffer*)
6: **if EligibleForAdaptation**(JP) **then**
7: BLFW ← **AdaptBLFW**(JP, BLFW)
8: **else**
9: {No update}
10: **end if**
11: **else**
12: {Keep going}
13: **end if**

Figure 5.5 shows the flow between the two key units of the CVC and the OBSERVER in the adaptation component of COSTAR. The CVC detects and validates the dependency between *Buffer* and JP data from one side and the new data chunk ($Chunk_{t_n}$) from the other side. It also aims to isolate suspected outliers by checking the cohesiveness at the three levels of JP, *Buffer*, and $Chunk_{t_n}$. Therefore, the CVC finally generates a reformed *Buffer* that is fed to the OBSERVER along with the new data chunk for monitoring the movement of data stream. The OBSERVER is integrated with the latest updated BLFW for declaring a decision based on the most recent information.

In the following, we discuss both the Cohesion Validation Component and the OBSERVER in detail. We start with discussing the Cohesion Validation Component as a preliminary step for reforming the *Buffer* to feed it into the OBSERVER for actual declaration.

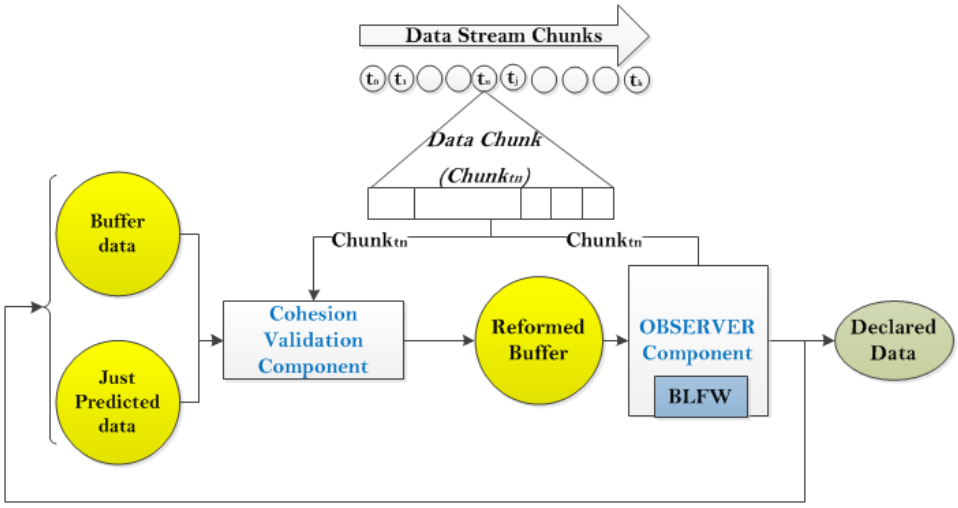


Figure 5.5: COSTAR Adaptation Component Overview

5.6 Cohesion Validation Component (CVC)

The goal of this component is to perform the essential reformation of the *Buffer* for monitoring the movement in the following OBSERVER component. The CVC performs the reformation by processing the three repositories of JP, *Buffer*, and the new data chunk $Chunk_{t_n}$. The flow chart of Figure 5.6 shows the flow of data for reforming the *Buffer* in the CVC. The principle function to check cohesiveness in the CVC is the Cohesion Validation Test (CVT).

There are four operations performed on the *Buffer* based on this algorithm, namely seed, replace, keep, and reset. The *Buffer* is a decisive parameter in the algorithm. The case when the *Buffer* is empty refers to the beginning of the stream or when a decision is just declared. Intuitively, no reformation is required when the *Buffer* is already empty as well as JP being empty as well. We also do no change when JP is not empty but not cohesive (continuous) with the new incoming data. The case when there is a dependency between JP data and the incoming data chunk triggers the seed operation, which adds a reference for JP data to be a seed in the new *Buffer*. Thus the seed operation states the dependency between the JP data and the new incoming chunk of data. The data movement is monitored by the OBSERVER considering this dependency.

Whenever the *Buffer* is not empty, it means it contains undeclared data because there was insufficient information to declare a decision. Therefore, the *Buffer* is still accumulated with new data until it reaches its size limit or a decision was made. However, before accumulating the *Buffer* with new data, we need essentially to check the dependency between the data already existing in the *Buffer* and new data to be added. If data in both repositories

are coherent, which indicates dependency, then the CVC feeds *Buffer* data to the OBSERVER with no reformation required. Otherwise, the non-cohesion between incoming data and *Buffer* data would indicate either suspected outliers or alternation between activities. The two operations of replace and reset rely on performing another layer of testing in this case to check the possible dependency between JP and the incoming data.

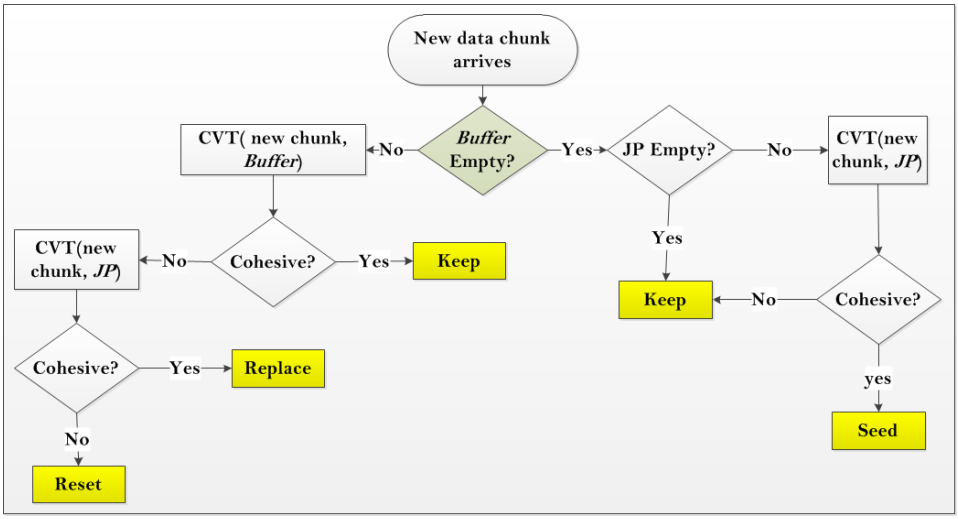


Figure 5.6: Flow Chart of Cohesion Validation Component

In the context of activity recognition, the cohesion validation component has the following key advantages:

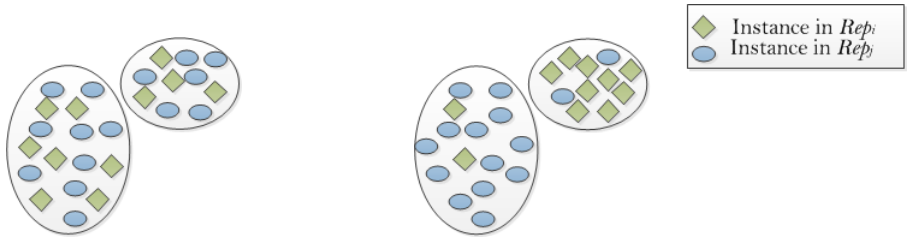
- *Ability to detect a continuous class/activity:* When data of a particular activity (act) streams from time t_i to time t_j across k chunks, the algorithm may declare *Buffer* data after processing c chunks where $c < k$. Once data is declared, the algorithm resets the *Buffer* and sends declared data to the short term memory of JP. Then, COSTAR continuously receives new data chunks that represent the same activity act - following chunk c. When the CVC confirms the cohesiveness between JP and the new chunk, we seed the empty *Buffer* with JP data and thus monitor data movement based on the previous related information.

- *Outlier isolation:* The CVC checks the *Buffer* for suspected outliers. *Buffer* data is released if the CVC detects no cohesiveness between the incoming data chunk and both the *Buffer* and JP.

The key function for testing the continuity and dependency among data is the Cohesion Validation Test. We discuss the definition of the cohesion validation and details of the test procedure.

5.6.1 Cohesion validation test

Any data repository consists of a set of data instances. We first define data repository as follows: $Rep_i = \{ inst_{i0}, inst_{i1}, inst_{i2}, ..., inst_{in} \}$ Where n is the number of instances belonging to the repository. In CVT, the repository definition applies for the *Buffer*, JP and $Chunk_{t_n}$ -stream data chunk at the nth time stamp. CVT checks a pair of repositories by the following procedure. It merges instances from both repositories and applies an online clustering technique to the merged data to generate exactly two clusters. If the generated clusters contain mixed up instances from both repositories, that means they are strongly related. Thus, the two repositories are cohesive as they are strongly attached. On the other hand, when the two clusters are well separated, each contains almost pure instances from one of the repositories, then CVT declares a separation between the two repositories. By the definition of the cohesion test, two repositories are separated if any of them is an empty set of instances. Figure 5.7 explains the two expected outputs of the cohesion validation test.



(a) Cohesive Repositories (b) Separated Repositories

Figure 5.7: Cohesion Validation Explanation

Based on the CVT, there are four operations to reform the *Buffer*. The four operations are seed, replace, reset, and keep. Description of the four operations is described in the following.

SPRK operations

Using the aforementioned cohesion validation test, one of the following four operations is performed to reform the *Buffer*. The four operations, abbreviated SPRK, are Seed, rePlace, Reset, and Keep. Definition of each operation is defined as follows:

- **Seed Operation:** This operation is the one that declares the dependency between just declared data and the new data chunk.
  Preconditions: the *Buffer* is empty and JP is cohesive with $Chunk_{t_n}$.
  Actions: Initiate the *Buffer* with JP data.

- **Keep Operation:** This is the case when the *Buffer* is being accumulated and contains data that is coherent with new incoming data. That means there is a dependency between the new incoming data and the data in the *Buffer*. Therefore, no reformation is required.
  Preconditions: When the *Buffer* is in cohesion with $Chunk_{t_n}$ or when the *Buffer* is empty and JP is not cohesive with $Chunk_{t_n}$.
  Actions: No change.

- **Replace Operation** [suspected outlier case]: This is a special case occurs when both the *Buffer* and JP are not empty yet $Chunk_{t_n}$ is incoherent with *Buffer* data. In this case, the CVC checks the dependency between $Chunk_{t_n}$ and JP for the detection of possible outliers. When data that has been accumulated in the *Buffer* is not cohesive with the dependent data in both JP and the new incoming data in $Chunk_{t_n}$, then it is suspected to be outlier or noisy data.
  Preconditions: the *Buffer* is not empty and not cohesive with $Chunk_{t_n}$, however JP is cohesive with $Chunk_{t_n}$.
  Actions: the *Buffer* is declared, based on available information and JP data replaces *Buffer* data.

- **Reset Operation:** In this case, new incoming data is not in cohesion with both the *Buffer* and JP. Therefore, it represents a different and independent concept.
  Preconditions: When the *Buffer* is not empty and both *Buffer* and JP are not cohesive with $Chunk_{t_n}$.
  Actions: Reset all and release data in the *Buffer* with available information.

The decision tree through the four operations based on repository contents is described in Figure 5.8. In Reset and Replace operations, we release data in the *Buffer* and declare a decision based on available information. The available information might indicate an existing or novel activity with uncertainty. Therefore, the decision will be declared accordingly with uncertainty.

The CVC uses the incoming data chunk to decide upon the cohesiveness and reform the *Buffer* accordingly. Thus, the CVC is an essential preparation step for checking the dependency among data and enabling the monitoring of a continuous stream of data. The reformed *Buffer* along with the new data chunk are sent as an input for actual monitoring of data movement by the OBSERVER. Details of the OBSERVER procedure are depicted in the next section.

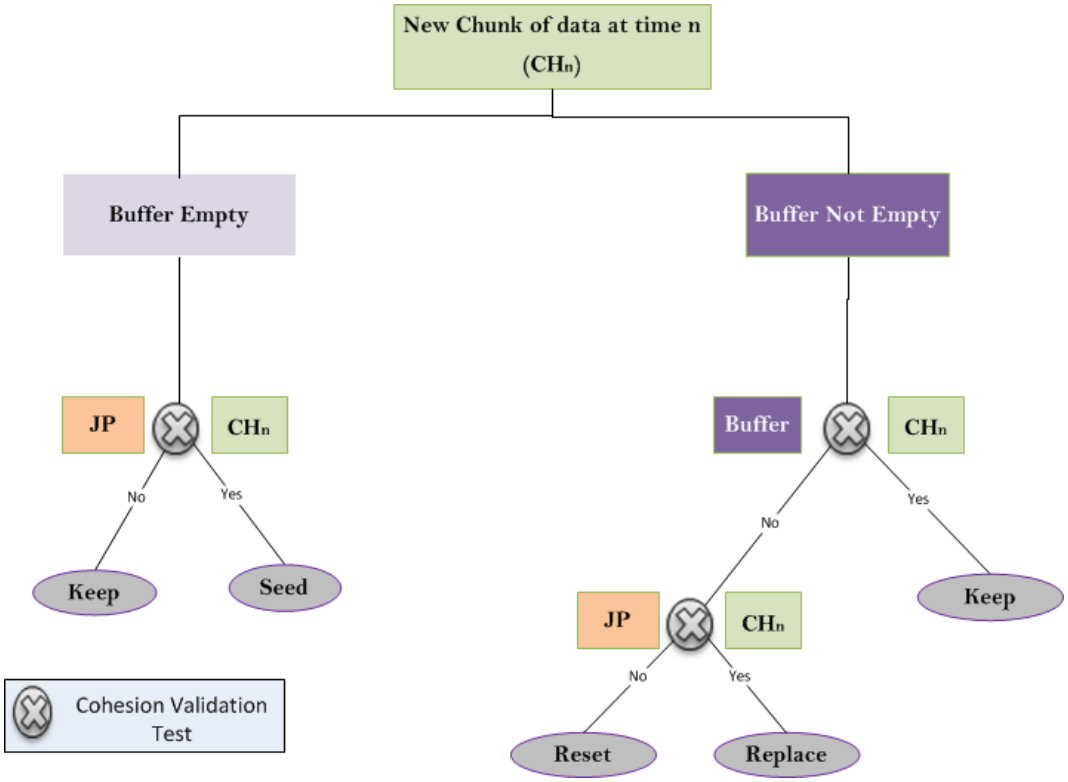


Figure 5.8: SPRK Operations

5.7 OBSERVER Component

The OBSERVER is the key component in the concept evolution technique that is responsible for monitoring different types of concepts and declaring decisions. It receives the new data chunk along with the reformed *Buffer* (output of the Cohesion Validation Component) in order to track the data movement and decide accordingly.

The core idea in this algorithm is to observe data movement along the stream until reaching either a decision or the maximum allowed *Buffer* capacity. There are two data repositories that are processed with the OBSERVER component, $Chunk_{t_n}$, and *Buffer*. *Buffer* contains the undeclared data which is accumulated since the last declared decision at t_j and up to t_n. In terms of time, $Chunk_{t_n}$ arrives at t_n while *Buffer* data is collected along duration from t_j to t_n, where $t_j < t_n$. The movement of data is monitored by checking the evolution in data from *Buffer* and $Chunk_{t_n}$ between t_j and t_n. In abstract, the OBSERVER takes one of three decisions which are *existing*, *novel*, or *unknown*. An *existing* decision reflects the recognition of a data repository as part of the underlying BLFW existing clusters. This decision is declared when data falls inside the boundaries of any of the BLFW clusters or consistently moves closer to existing clusters. On the contrary, a *novel* decision is declared when data resides outside all of the BLFW clusters' boundaries and consistently moves

away from all clusters. The novel concept must also satisfy a stability cohesion and separation criteria to be considered as an eligible novel concept. *Unknown* decisions are taken mostly when the maximum *Buffer* capacity is reached, with no confidence to take any of the other two decisions. In this case, the OBSERVER declares the data as unknown and resets all of the settings. In a nutshell, the OBSERVER either takes one of the aforementioned decisions or keeps the *Buffer* accumulating, seeking more data to reach a decision whenever more data chunks arrive.

After giving an overview on the OBSERVER component, we then define the problem space, this will be followed by the algorithm details and illustration of the different outputs of the OBSERVER component.

5.7.1 Problem space definition

The OBSERVER is integrated with the most updated BLFW that consists of a set of clusters forming the normal concept. Each cluster has its own boundaries. The OBSERVER has a unique advantage of detecting novel clusters that appears inside or outside the normal concepts global boundary as explained in Figure 5.1. Before we get in details of the OBSERVER methodology and scenarios, a few definitions and assumptions have to be set first.

The problem space consists of the BLFW with clusters, which represent activities and sub-clusters of each cluster, which represent patterns inside each activity. We start with the explanation of the clusters' decision boundaries in the problem space. Each cluster in the BLFW is a hypersphere in a feature space that has its own local decision boundary. The decision boundary of the cluster is identified based on $Dmax_c$ characteristic. $Dmax_c$ is a cluster level characteristic that defined in Table 5.1, as the maximum distance between any pair of sub-clusters within the cluster.

Surrounding the decision boundary of each cluster, there is a slack that accommodates for drift of data that represents the cluster. We term this slack as DRAB that stands for **D**ynamic **R**el**A**xed **B**oundary. DRAB is a dynamic boundary as it is flexible and adaptable for each cluster. The definition of DRAB is given as follows:

Definition 5.1 ***(DRAB: Dynamic RelAxed Boundary)*** *An adaptable boundary surrounding the cluster beyond the decision boundary, for each cluster $C_i \in BLFW$. DRAB allows an extended space to accommodate concept drift for each cluster.*

Figure 5.9 shows the position of both decision boundary and DRAB around a single BLFW cluster with sub-clusters. DRAB height is the threshold that defines the space between the decision boundary and DRAB.

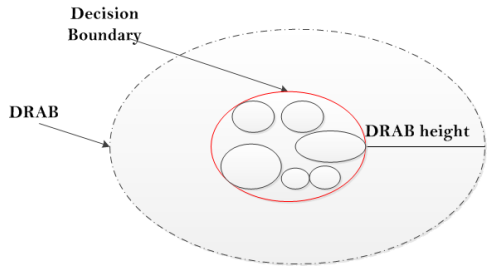


Figure 5.9: Illustration of DRAB and Decision Boundary Surrounding a BLFW Cluster

OBSERVER component processes the data in two repositories. The two repositories are $Chunk_{t_n}$ which is the chunk of the data stream arriving at time t_n, and the *Buffer* which is a short memory that contains undeclared most recent data. To be able to check the movement along time, we need first to define the different possible positions of data repository with respect to the BLFW. First, a BLFW with m clusters and k sub-clusters across all m clusters is defined as
$BLFW = \{C_1\{sc_{11}, sc_{12}....sc_{1P_1}\}, C_2\{sc_{21}, sc_{22}....sc_{2P_2}\}, ...C_m\{sc_{m1}, sc_{m2}....sc_{mP_m}\}\}$.
Where $k = \sum_{i=1}^{m} P_i$. Rep_l is a general term that refers to any data repository. In the OBSERVER, term Rep_l refers to either $Chunk_{t_n}$, the *Buffer*, or a union of both. We define the spatial position between repositories based on their centroids. The spatial position of data repository in respect to the BLFW clusters is one of three: *Inside*, *InSlack* or *OutsideALL*.

- *Inside:* Rep_l is Inside $C_j \in$ BLFW, if the Euclidean distance between the centroid of Rep_l and $Centroid_{C_j}$ is less than $Dmax_{C_j}$. In other words, the repository is inside a BLFW cluster, if the centroid of the repository resides inside the decision boundary of this particular cluster.

- *InSlack:* Rep_l is InSlack of $C_j \in$ BLFW if the Euclidean distance between the centroid of Rep_l and $Centroid_{C_j}$ is above $Dmax_{C_j}$ yet less than $DRAB_{C_j}$. $DRAB$ is the dynamic relaxed boundary created around each cluster of the BLFW as in Definition 5.1. InSlack explains the case when the centroid of the new repository is outside the decision boundary, yet it still resides inside the relaxed boundary surrounding the cluster. Slacks of different clusters might overlap. When data resides in the slack of more than one cluster, then we monitor the movement in regards to all suspected clusters.

- *OutsideALL: Rep_l* is OutsideALL when the repository centroid is outside the $DRAB_{C_j}$ $\forall$ $C_j \in$ BLFW. This definition applies also for a repository that resides inside the global boundary but outside the boundaries of all clusters in the BLFW.

The OBSERVER focuses on both spatial and time domains. In order to monitor the movement of data stream, we keep information about the most recent undeclared data in the *Buffer* and current data in $Chunk_{t_n}$. We define a new term that explains the expected next position of merged instances of both the *Buffer* and $Chunk_{t_n}$. The term *VINPos*, which stands for VIrtual Next Position, indicates the virtual position when the *Buffer* is accumulated with new data chunk as well as the movement of data. The position of any repository is calculated based on the centroid location. *VINPos* is defined as follows:

Definition 5.2 *(VINPos: VIrtual Next Position:) The prospective centroid position of the Buffer when accumulated with $Chunk_{t_n}$. VINPos has one of two values: either IN when the merged position resides Inside or InSlack of any BLFW clusters or OUT otherwise.*

Based on the definition of the problem space and formalisations, we explain the details of the OBSERVER methodology for detecting the evolution of concepts along the stream.

5.7.2 OBSERVER overview

The OBSERVER monitors the evolution of the reformed *Buffer* and $Chunk_{t_n}$ over time. Algorithm 5.2 illustrates steps of the OBSERVER procedure. The algorithm starts with checking data in the *Buffer* at time t_n (line 1). The *Buffer* is empty in two scenarios. The first is at the beginning of the stream, when no data has been processed yet. The second scenario is when the algorithm resets all settings and empties the *Buffer*, when data is just declared or released at t_{n-1}. In these two scenarios, the new chunk is processed with the **P**rocess **D**ata with **N**o **H**istory (PDNH) procedure (line 2). Otherwise, in case the *Buffer* is not empty, the algorithm calculates *VINPos* and checks its position in respect to the BLFW clusters (line 4). The OUT value of *VINPos* (line 5) indicates its location outside the C_j boundaries $\forall C_j \in$ BLFW where $j \in 1...m$. When the OUT value of VINPos is validated, the data is suspected to be novel as its virtual position is spatially outside the boundaries of all existing clusters. The suspected novel data is tested with the **ObserveSuspNovel** procedure for monitoring and detecting novel concepts (line 6). Otherwise, when *VINPos*

resides inside the boundaries of any cluster $C_j \in$ BLFW, then the accumulated data is declared as an existing concept (line 8).

Algorithm 5.2 OBSERVER Algorithm

Input: $Chunk_{t_n}$: data chunk at time t_n
Buffer: short time memory of previous undeclared data
BLFW: most recent baseline framework
Output: declared data if any
1: **if** Empty(*Buffer*) **then**
2: **PDNH**($Chunk_{t_n}$, BLFW)
3: **else**
4: *VINPos* = CalculateVINPos($Chunk_{t_n}$, *Buffer*)
5: **if** *VINPos* is OUT **then**
6: **ObserveSuspNovel**($Chunk_{t_n}$,*Buffer*, BLFW)
7: **else**
8: **DeclareExisting** ($Chunk_{t_n}$, *Buffer*, BLFW)
9: **end if**
10: **end if**

The OBSERVER implements one of three procedures of PDNH, ObserveSuspNovel, or DeclareExisting. Each of the three key procedures may declare a decision whenever adequate information is available. The PDNH procedure deals directly with $Chunk_{t_n}$ and checks its position from the BLFW clusters when there is no previous history of its movement. The position is either *Inside*, *InSlack*, or *OutsideALL*. A $Chunk_{t_n}$ that falls *Inside* any clusters' decision boundaries is classified and declared as existing. The declared data is moved to the JP short memory for checking the dependency with the following data chunks. In all other cases, PDNH initiates the *Buffer* with data in the $Chunk_{t_n}$ that resides *InSlack* of any of BLFW clusters or *OutsideALL*. At this stage, there is not enough information to check the movement of data. Therefore, the data is stored in the *Buffer*, for further processing when more stream chunks arrive.

PDNH is the basic procedure that is implemented at the beginning of the stream or when all settings are reset. When the *Buffer* is not empty, i.e., undeclared data exists in the *Buffer*, the virtual position decides on the movement according to new data whether closer or away from the BLFW underlying concepts. Moving closer declares an existing concept; moving away suggests the appearance of a novel concept. The two main processes in the OBSERVER are for declaring existing concepts and observing suspected novel concepts. Details of the two key processes are discussed in the following.

5.7.3 Detecting existing concepts

Due to the evolving nature of the data stream, the OBSERVER has to accommodate for the expected concept drift. Changes in decision boundaries are

expected when dealing with data streams. For activity recognition, the way people are performing activities is different from one user to another. Thus, in our algorithm we create DRAB which is a dynamic slack that surrounds a cluster's decision boundary for coping with concept drift in data streams. A key objective for the OBSERVER is to distinguish between existing concepts with drift and completely novel concepts.

The OBSERVER declares an existing concept including concept drift, when *VINPos* has IN value according to Definition 5.2. Table 5.2 states the set of scenarios that causes *VINPos* to have the value of IN - the general term for InSlack and Inside. The declaration of the existing decision is limited to these five scenarios. All other scenarios correspond to either suspected novel or unknown concept and will be discussed later in this section. Rep_i and Rep_j refer to the *Buffer* and $Chunk_{t_n}$ or vice versa. The order is not necessary here because the merged position of both repositories is the important factor, regardless of each individual position.

By the definition of positions, Inside and InSlack are attached to a specific cluster c_i. For each row in the table, if the three conditions - for columns in each row- are satisfied, then the value of VINPos is IN. For instance, at row 4 when both repositories are inSlack of the same cluster c_i and the centroid of merged data locates $InSlack_{c_i}$ as well, then VINPos has an IN value. Thus, the data in the *Buffer* accumulated with $Chunk_{t_n}$ is declared as existing with possible concept drift. Another examples are at row 1 and 2, when one repository resides $Inside_{c_i}$ and the other one is $InSlack_{c_i}$, therefore the resultant merged position can be either $InSlack_{c_i}$ or $Inside_{c_i}$. Both are considered as an IN value of *VINPos* and thus declared as existing possibly with concept drift of cluster c_i. In general, when the incoming data drags the data closer to any of the underlying existing concepts, this means it is moving closer. Thus the accumulated data is declared as existing and possibly with a drift. The declared data is then moved to JP short memory. Then, all other settings are reset to default as per the beginning of the stream.

Table 5.2: Cases when *VINPos* is IN

Rep_n Pos	Rep_k Pos	Merged Pos
$Inside_{c_i}$	$InSlack_{c_i}$	$Inside_{c_i}$
$inside_{c_i}$	$InSlack_{c_i}$	$InSlack_{c_i}$
$Inside_{c_i}$	$OutsideALL$	$InSlack_{c_i}$
$InSlack_{c_i}$	$InSlack_{c_i}$	$InSlack_{c_i}$
$InSlack_{c_i}$	$OutsideALL$	$InSlack_{c_i}$

The value of *VINPos* decides on the movement of the evolving data. Therefore, when the data neither resides in the boundaries nor moves closer to any of the existing clusters in the BLFW, *then it is suspected to be novel.*

5.7.4 Filtration process

Unlike the *VINPos* IN value in the aforementioned cases, the OUT value denotes the position of the merged data centroid OutsideALL clusters. However, the OUT value does not always indicate the appearance of a novel concept. In order to find a genuine OUT, we have to understand the genuine reason that causes the merged data to reside outside. Therefore, we can decide whether the accumulated data is suspected novel concept or existing one with concept drift. We defined genuine out as follows:

Definition 5.3 *(genuine out (genOUT)): VINPos for observing Rep_n and Rep_k has a genuine out value if the position of both repositories in addition to the merged position are OutsideALL.*

Table 5.3 states the possible scenarios that cause OUT and genOUT values for VINPos.

Table 5.3: Cases When *VINPos* is OUT and genOUT

Rep_n Pos	Rep_k Pos	Merged Pos	VINPos
$Inside_{ci}$	$OutsideALL$	$OutsideALL$	OUT
$InSlack_{ci}$	$OutsideALL$	$OutsideALL$	OUT
$OutsideALL$	$OutsideALL$	$OutsideALL$	genOUT

According to Definition 5.3, *VINPos* may reside outside with no genuine reason, because of one or more of the following. (i) one of the data repositories represents noise, (ii) one of the repositories is an outlier, (iii) repositories refer to interleaved classes (in an activity recognition context, interleaved classes are commonly occur especially with strongly related activities such as "Walk" and "Stand"), and/or (iv) there is an error in positioning repositories because of the shifting in BLFW clusters boundaries due to concept drift.

The filtering process of non-genuine OUT cases resolves the first three issues. Whereas, the adaptation of the BLFW resolves the last case. In all cases of non-genuine OUT, the filtration process is triggered when the VINPos is OUT, yet one of the repositories resides Inside or InSlack (row 1 and 2 in Table 5.3). The most confident decision in these cases is to separate the two repositories; declare repository that resides *Inside* or *Inslack* and keep *OutsideALL* data in the *Buffer* for further investigations whenever new other

chunks arrive. The declared data is kept in the JP repository for reference. The filtration process does not stop here. In a non-genuine case, we collaborate with the CVC for deciding on the *Buffer* and JP data. The CVC complements the filtration process with either the *Replace* or *Keep* operation depending on the new incoming chunk at (t+1).

When new data arrives while JP and the *Buffer* are not empty, the Cohesion Validation Test checks the cohesion with both JP and the *Buffer* based on the decision tree in Figure 5.8. Outliers and noise will be filtered out if the new chunk is a continuation of JP data by the *Replace* operation. Thus, the *Buffer* data that contains noise/outlier is isolated and declared with whatever available information. In case that both JP and the *Buffer* represent two real classes (because of the interleaved classes), the new data chunk flows through the normal procedure of the CVC to choose between the two classes with either *Keep* or *Replace* operations.

After resolving the case of non-genuine out data, in the following we discuss the case of moving away from all existing clusters with a suspected genuine case of novelty.

5.7.5 Detecting a novel concept

Once genuine OUT (genOUT) data has been detected, we perform further analysis to ensure the arrival of a novel concept. Data that resides genuinely out is suspected to be novel until satisfying stability criteria.The algorithm commences by checking the movement of data along time. Nevertheless genOUT indicates the virtual position of OUT and for a genuine reason, it does not indicate the actual movement from all existing clusters. Data might be at a genOUT position, yet moving closer to any BLFW existing cluster. Thus, genOUT data that moves closer to any of the existing BLFW clusters is kept in the *Buffer* for further investigation and is suspected to be concept drift rather than a novel concept. The novel concept is the one that is genOUT and "moves away" from all existing BLFW clusters. The suspected novel concept has also to satisfy a set of stability criteria, to be declared as novel.

To materialise the notation of moving away, the novelty detection technique considers data in $Chunk_{t_n}$ as more recent along the time line than data in the *Buffer*. Thus, the algorithm computes the distances from the $Chunk_{t_n}$ centroid to BLFW cluster centroid and compare them to the distance between the *Buffer* and cluster centroid for all clusters existing in the BLFW. The increase of distance depicts the away movement of data while time evolves. Figure 5.10 explains an example of the detection of away movement from

a BLFW containing four clusters. Di_t is the distance between the *Buffer* centroid (up to time t) and the C_i centroid, while Di_{t+1} is the distance between the centroid of new data chunk arrives at time (t+1) and the C_i centroid. As shown in Figure 5.10, the moving away criterion is satisfied only when $Di_{t+1} > Di_t \forall C_i \in BLFW$.

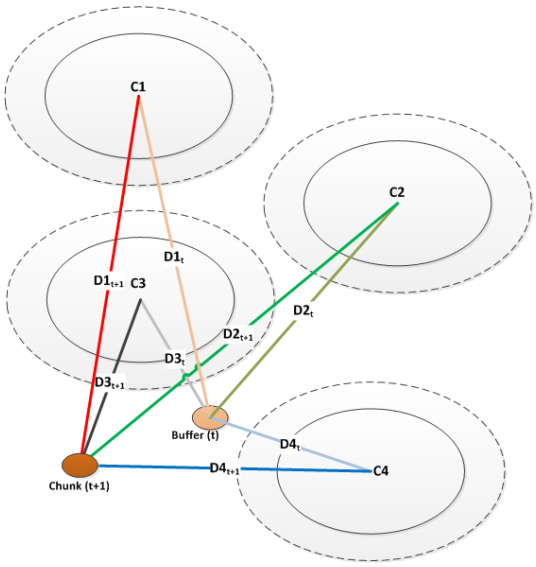


Figure 5.10: Away Movement of Novel Concept

A moving away concept is a suspected novel class that must satisfy a set of stability criteria, to be declared as a newborn novel concept. The stability criteria that concern both separation and cohesion of the novel concept are as follows:

- *Weight:* The suspected cluster has to reach a specific size threshold before considered as a newborn novel concept. The choice of weight threshold has to range in between the $Chunk_{t_n}$ size and the maximum *Buffer* size.

- *Density:* This criterion maintains the cohesiveness required, for suspected novel clusters to be declared as newborn novel concepts. The data repository density is defined as per equations 5.1 and 5.2.

$$Density_{Rep} = \frac{Weight_{Rep}}{Volume_{Rep}} \tag{5.1}$$

$$Volume_{Rep} = \frac{4}{3}\pi Radius^n_{Rep} \tag{5.2}$$

where $Volume_{Rep}$ is the volume of the data repository as a hypersphere in n-dimensional space.With high dimensional datasets,we deal with data as a hypersphere in 3-D (n=3) to avoid a possible curse of dimensionality.

The *Buffer* density is first computed and then compared to the density of the merged repository when $Chunk_{t_n}$ accumulates the data in the *Buffer*. The cluster satisfies density stability criteria if it attains a density gain when $Chunk_{t_n}$ accumulates the data in the *Buffer*.

- *Centroid movement:* To maintain the required stability, candidate cluster centroid movement has to be minimal before the cluster is declared as a novel newborn concept. Continuous movement of the centroid reflects a lack of cohesion and stability in the suspected data. Therefore, the algorithm keeps accumulating data in the *Buffer* until meeting the minimal movement criterion.

Suspected data that satisfies all stability conditions is declared as a newborn novel concept. A newborn novel concept adapts the BLFW to facilitate the recognition of future recurring occurrences of the newborn concept. Due to the flexible and lightweight model, the process of incorporating a novel concept into the BLFW is computation and time efficient. This is essential especially with stream mining time and resource constraints. The OBSERVER can also detect more than one novel cluster, as long as it follows the same procedure and satisfies the same criteria.

Lastly COSTAR decides whether data should be declared as unknown. The data is declared unknown when it is released from the *Buffer* with no confidence towards either of the two decisions of existing or novel. The reason for releasing data from the *Buffer* is either reaching the maximum capacity or SPRK operations of Reset and Replace, in the Cohesion Validation Component. While accumulating the *Buffer*, we keep records about data movement. When we reach the maximum *Buffer* capacity, we endeavour to reach a decision by checking the movement records of the data. If the data movement is confusing and does not represent evidence for either existing or novel, then the data is declared unknown. The other reason for releasing data comprises the SPRK operations of Replace and Reset. The reset operation is triggered when no sufficient information exists in the *Buffer* and new data arrives that represents different and independent concept. While the replace operation is triggered when outliers and noisy data detected and filtered out. Data in the *Buffer* is also replaced in the case of interleaved concepts. In general, data is declared unknown when it is released from the *Buffer* with insufficient information for either existing or novel decisions.

With continuous learning in the Cohesion Validation Component and OBSERVER, COSTAR is capable of detecting the emerging of novel concepts in the evolving streams. The two components are integrated for continuous

monitoring for dependency and movement of data from underlying existing concepts. The algorithm is proposed and developed for mining the evolving activities from sequential data streams in activity recognition. Three decision outcomes are expected from the OBSERVER: data belongs to an existing activity, data represents a novel activity, or data is unknown. Based on these declared decisions, COSTAR adapts the model continuously to cope with the detected changes and thus enhance the recognition efficiency.

5.8 Model Adaptation

Upon a decision declaration with OBSERVER, we adapt the BLFW with the most recent changes detected in the activity streams. Both decisions of novel and unknown are eligible for model adaptation, while the decision of existing is not. Declaring a novel concept includes the first appearance of the newborn concept and also recurring occurrences that belong to the same novel concept. When we detect the appearance of the novel concept at the first place, we adapt the BLFW by adding the newborn novel concept to the BLFW. We do that by simply forming a cluster of the data that is declared as a newborn concept and generate sub-clusters of the newborn cluster. We then extract all characteristics of the newborn cluster and its sub-clusters and incorporate them into the BLFW characteristics.

Our algorithm assimilates a newborn concept into the BLFW. Therefore, it enables classification of recurring occurrences of the novel concept and distinguishes it from existing concepts. In order to strengthen and boost the newborn existence in the BLFW, we incrementally accumulate the newborn cluster with the automatically recognised recurring data chunks. The detection of recurring instances is part of the learning algorithm as the newborn concept is integrated into the BLFW after adaptation. We apply incremental learning only for the newborn cluster, in order to enrich its presentation with more data. The incremental learning approach updates the characteristics of the newborn cluster to reflect the accumulation of new data. Eventually, a complete reformation of the newborn cluster is required to maintain more accurate representation of the cluster. Because of the fact that a complete reformation consumes resources, especially when the cluster becomes bigger, we limit the reformation to few runs and before the cluster grows heavier. We assume that the cluster is heavy weighted if its weight exceeded the minimum weight of any of BLFW clusters. A heavy weighted cluster is considered as a normal cluster instead of being newborn. Upon reaching the heavy weight threshold, no reformation is permitted and all raw data is dismissed.

The other decision that triggers BLFW adaptation is the declared unknown decision. The most common cause of the unknown decision is reaching the limit of the *Buffer* size with insufficient information to make a decision. Therefore, the data in the *Buffer* in this case is uncertain and thus it requires model adaptation to accommodate for it. Other reasons for unknown decision are noise and outliers. We incorporate an active learning approach in COSTAR to ask the user to label the unknown data. When uncertain data exits, active learning is required. Active learning triggers inquiries in the form of unannotated instances to be annotated by user. When the inquiries have been processed, the BLFW is adapted in batches accordingly with the true labels of the data.

One of the reasons for the existence of unknown decisions is the shifted and erroneous boundaries. Therefore, BLFW adaptation for both decision and DRAB boundaries is essential to reflect to the most recent changes in data streams. The decision boundary of any cluster relies on $Dmax$ which is the maximum distance between any pair of sub-clusters centroids within the clusters. Thus, the centroid is one of the most important and dominant characteristics that requires adaptation. We adapt the centroids of the sub-clusters according to the algorithm 4.3 for incremental learning and active learning. This was discussed in Chapter 4.

Since higher level characteristics are based on the lower level ones, the adaptation process recalculates both cluster and holistic characteristics with the updated values of sub-cluster characteristics. In addition to refining the decision boundaries of BLFW clusters, we also adapt the DRAB height to adjust the elastic boundary surrounding the updated cluster. Thus, we expand or shrink DRAB to contain or exclude the unknown data. The expansion or shrinkage of DRAB height is dependent on the position of unknown data from the cluster with the true label, as well as the weight of unknown data.

Model adaptation is essential to detect novel activities and their recurring instances. Equally important, the adaptation process enhances the model representation by including the most uncertain (i.e., unknown) data. The adaptation of the model helps in reducing the occurrences of false positives. A false positive occurs when data belonging to an existing activity is falsely declared as novel. Reasons for false positives are many, including insufficient training data, an outdated model, or inaccurate boundary positions. To avoid this, we apply the adaptation process for updating both BLFW characteristics and boundaries. A false positive also occurs when recurring instances of the already declared novel concept are detected as totally novel concepts instead of recurring. This case may occur because of the poor representation of the newborn novel concept due to the small number of data instances. We try to

resolve this by incremental learning with recurring instances and reformation of the newborn cluster, until it is well presented in the BLFW. In the following, we highlight all cases that trigger active learning in COSTAR.

5.8.1 Active learning

Active learning in COSTAR is triggered in two cases. The first case occurs with the most uncertain data which refers to an unknown decision. Also, active learning is provoked when a novel concept is first detected. That happens when (genOUT) data is moving away from existing concepts and satisfies all the stability criteria. We denote this case a First Appearance case (FA). Any instances that belong to the novel concept following the FA in the time line are termed recurring instances. Once a novel concept is detected in the FA case, COSTAR requires a confirmation from users on the appearance of a novel concept. The validation of the arrival of a novel concept is essential, as that avoids the erroneous adaptation with incorrectly detected data. Once the user confirms the novel concept, the OBSERVER adapts the BLFW automatically by assimilating it into the existing model. The adapted BLFW is deployed for further recognition of recurring instances of the recently detected novel concept. The detection of recurring concepts does not require active learning as it is part of the learning process.

There are three possibilities that happen when OBSERVER triggers active learning in case of FA. First, the user confirms the arrival of a novel concept, and therefore the BLFW is adapted accordingly. The second possibility is the case of a false positive (FP) when an existing concept is incorrectly classified as novel. The algorithm checks whether the DRAB is the reason for the misclassification and thus expands or shrinks the DRAB accordingly. The third possibility would be when recurring instances of already declared novel concepts are declared as the FA of a new novel concept. The OBSERVER accumulates the already declared novel concept in BLFW with recurring data in this case.

In addition to detecting a novel concept and adapting the BLFW accordingly through continuous and active learning, we present in the following a technique for forgetting abandoned concepts.

5.8.2 Forgetting outdated concepts

In activity recognition, the set of activities is not static along the stream. As novel activity may emerge, outdated activities might disappear. In order to keep the BLFW up to date with the most recent changes in the stream, we

aim also to detect the abandoned activities and exclude it from the BLFW automatically. COSTAR is capable of detecting activities that are no longer relevant to the recent data streams. While COSTAR continuously monitors the evolution of concepts, three decisions are declared: existing, novel, and unknown. In case of an existing decision, COSTAR automatically assigns a time stamp for each activity that appears in the stream and is classified as existing. The algorithm scans the BLFW periodically for existing clusters that have not been stamped for a long time. When the outdated activity is detected, COSTAR triggers an inquiry in an active learning approach to get user confirmation for removing the abandoned activity from the BLFW. Deletion of the activity is simple as it only removes the characteristics that represent the abandoned activity from the BLFW.

5.9 COSTAR Contribution

In this chapter, we propose and develop an adaptive technique for concept evolution in activity recognition. COSTAR provides the following major contributions:

- *Detecting novel activities:* COSTAR applies continuous learning to monitor the evolving activities in data streams to detect novel activities whenever they arrive. The novelty detection technique is tailored to activity recognition as it considers the sequential dependency in data, which is intuitive in activity recognition. The new technique is capable of recognising both abnormal activities and new normal activities that appear spatially in the middle of other normal underlying concepts in the learning model.

- *Distinguishing novel activity, concept drift, and outliers:* COSTAR declares data as novel, existing, or unknown. The learning technique in COSTAR can distinguish between the concept drift of an existing concept and the appearance of a novel concept. We created an elastic slack around each cluster to accommodate concept drift. The declaration of a novel concept requires data to reside outside all slacks and consistently move away from all existing activities. This way, our technique can distinguish between a completely new concept and an extension or drift of an existing concept. We also propose a dependency check and filtration process to isolate outliers that possibly appear in the stream.

- *Forgetting abandoned activities:* Equally important, COSTAR can detect the disappearance of activities in the recent stream. As we aim to detect

the appearance of novel activities, COSTAR also is capable of detecting activities that are no longer relevant. Therefore, it adapts the learning model continuously, using the detected changes.

- *Dynamic adaptation of the learning model:* In COSTAR, the underlying learning model is continuously enriched with novel detected concepts to accommodate recent changes in the stream and detect recurring data that belongs to novel classes. It also applies active and incremental learning with the most uncertain data, for alternation of the learning model and thus enhancement of the recognition accuracy.

- *Incorporating active learning with low cost:* The annotation process is very costly and impractical in a streaming environment. It is also non realistic to assume that annotated data that represents various activities are available at the training phase. Therefore, COSTAR triggers active learning only for the small amount of data that is the most uncertain.

5.10 Experimental Study

In this section, we evaluate the performance of COSTAR in terms of the efficient recognition of novel and recurring activities. The evaluation of COSTAR aims to analyse the following:

- *Detection of the first appearance of novel concepts:* We aim to evaluate the performance of COSTAR in detecting a novel concept once it arrives in data streams. The first appearance of any novel concept is very crucial as the model is adapted once the novel concept is declared. Therefore, an efficient model adaptation is dependent on an accurate detection of the first appearance of the novel concept.

- *Recognising the recurring occurrences of the novel concepts:* Following the first appearance of the novel concept, we aim to evaluate the performance of our technique to detect recurring occurrences of the detected novel concept. The efficiency of the adapted model is analysed according to its performance in recognising recurring instances of the newly added concept.

- *Distinguishing between concept drift and novel concepts:* A real challenge of our technique is to differentiate between concept drift and entirely novel concepts. The key cause of concept drift in activity recognition is the change in the ways of performing activities, from one user to another.

Thus, we aim to evaluate the performance of COSTAR across users to demonstrate its ability to differentiate between concept drift and novel concepts.

- *Evaluating active learning behaviour with evolving data streams:* COSTAR enables active learning in a few cases. In the batch active learning approach, COSTAR automatically triggers inquiries to request true labels from the user. The amount of inquiries along the stream as well as the amount of data required for refining the model are analysed to evaluate the efficiency of the developed active learning approach.

We conducted our experiments on three benchmarked activity recognition datasets: OPPORTUNITY, WISDM, and SPAD. The datasets are collected from different sensors including mobile accelerometer sensors and on body sensors. Details of publicly available activity recognition datasets were discussed earlier in this dissertation in Chapter 3. Additionally, we evaluate our technique using the Iris data set [Fis36], an explanatory small conceptual dataset, for a clear visualisation of COSTAR performance. We start the evaluation by discussing the experimental setup and performance metric. Then we evaluate COSTAR performance, in the light of our aims, on the benchmarked data.

5.10.1 Experimental setup

The setup for analysing and evaluating COSTAR aims essentially to test its efficiency in detecting the appearance of new concept(s) and its concurring instances in data streams with concept drift. In the context of activity recognition, concept drift is crystallised when comparing sensory data from different users. Therefore, in all our experiments, part of the data is used to build the BLFW while other data from a different user is applied for testing. The underlying BLFW is built from training data of multi classes. The default chunk size is 20 data instances unless otherwise stated. We handle the data stream with a continuous sliding window, using the same approach as described in Chapter 4. We also apply an online clustering for each chunk in order to separate concurrent activities appearing in a single chunk. To evaluate COSTAR, we first define the base metrics applied for measuring the performance. Tp, the number of novel instances correctly detected as novel; Tn, the number of existing instances correctly classified as existing; Fp, the number of existing instances falsely classified as novel; Fn, the number of novel instances incorrectly classified as existing; and *nInstances*, total number of instances in the stream. *nInstances* include Tp, Tn, Fp, Fn, and *unknown* instances. We refer

to total number of unknown instances as *NumUnk*. Performance measures are described in Table 5.4.

Table 5.4: Performance Measures for COSTAR

Name	Description	Formula
Acc	Accuracy as the percentage of correct classification along the stream	$\frac{Tp+Tn}{nInstances}$
ERR	Error rate in the data stream	$\frac{Fp+Fn}{nInstances}$
UnkPerc	Percentage of instances that are declared as unknown	$\frac{NumUnkn}{nInstances}$
Recall	Ratio of instances detected as novel among all true novel instances	$\frac{Tp}{Tp+Fn}$
Precision	Ratio of correctness in the examples classified as novel	$\frac{Tp}{Tp+Fp}$
Specificity	Ratio of instances classified as existing among all true existing instances	$\frac{Tn}{Tn+Fp}$
Fall-out Rate (FOR)	False alarm rate	$\frac{Fp}{Fp+Tn}$
False-discovery Rate (FDR)	Ratio of false positives among all instances classified as novel	$\frac{Fp}{Fp+Tp}$
ALRate	Number of triggered active learning inquiries per 10 thousand data instances	-
ALPoints	Percentage of total number of instances that are batch labelled with active learning inquiries	-
CPur	Percentage of the cluster purity, which is the percentage of instances with the major label within the cluster	-

COSTAR parameters: Our developed technique can be applied to any data domain when its data represents sequential data that can be also non identically distributed. Activity recognition is one of these domains. In the context of activity recognition, COSTAR is a learning technique for concept evolution that is independent on the sensing scheme. Thus, COSTAR can be

applied to a wide range of datasets for activity recognition, whenever it satisfies the dependency constraint and regardless of the scheme for collecting data, type of sensors, location of sensors, sampling rate, and other characteristics for the data collection. However, COSTAR parameters have to be tuned to suit some of these characteristics for better performance. Thus, we present the three sets of parameters that are highly attached to the sensory data characteristics. We describe the three categories of parameters that are set or tuned for COSTAR as follows:

- *Chunk size and Buffer size:* As we check movement of data along a stream, preference chunk size tends to be small to allow a reasonable time for monitoring the movement. Information about dataset sampling rate and minimum duration of any of the activities/class are essential to set a good chunk size. For instance, a small chunk size is required when the dataset has a sampling rate of 20 Hz and a minimum duration of 2 seconds for any occurrence of activities. The small duration of two seconds contains 40 instances. Thus, the chunk size for this dataset has to be much less than 40 instances to enable COSTAR to monitor the movement and hence capture the occurrence of this activity. We set the *Buffer* maximum capacity based on the chunk size. As explained in the OBSERVER methodology, a small *Buffer* capacity might increase the percentage of unknown data. On the other hand, a large *Buffer* capacity will increase the complexity. Therefore, to allow a reasonable time for monitoring the data movement, we set the capacity of the *Buffer* to accommodate the maximum of double the size of a data chunk.

- *Initial DRAB height:* This parameter refers to the initial threshold of area surrounding each cluster in the BLFW to accommodate for concept drift. While the stream evolves, the adaptation process expands or shrinks the DRAB height accordingly. However, setting a good initial value of this parameter is crucial to avoid high rates of both false alarms and active learning. This parameter is also strongly related to the decision boundary of the cluster, which is specified by $Dmax$. The initial DRAB height is manually tuned below the value of $Dmax$ to ensure stability and to avoid overlapping of slacks.

- *Stability weight of the novel concepts:* The concept is declared novel if it moves away from all existing concepts in addition to satisfying all stability criteria. Three stability criteria are applied to ensure the separation and cohesion of the suspected novel concept to be declared as novel.

These three criteria are: less movement of the cluster centroid, increased weight, and increased density. The stable weight (size) of a declared novel concept ranges in between the chunk size and the minimum number of instances representing any activity. A newborn novel concept is moving away and growing bigger over time. In addition, minimum centroid movement and increased density are required for a newborn concept to be declared stable. Both density and centroid movement criteria do not require pre-defined thresholds to be set beforehand. Alternatively, they both make a comparison between new and previous values to check increase, in the case of density, and stability, in the case of centroid movement, to satisfy the the two criteria.

In the following experiments, we tune the parameters for each dataset in order to adapt to the dataset characteristics and thus improve the performance of COSTAR. The computational complexity of COSTAR is similar to the analysis represented in STAR in Chapter 4. The main complexity arises from the online clustering. COSTAR adds the complexity of running the Cohesion Validation Test (CVT). The CVT is also another pass of online clustering runs on a maximum of n data instances, where n, in the worst case scenario, is the maximum *Buffer* capacity. The OBSERVER does not require extra processing overhead as it only performs a set of comparisons to decide on the data movement. We start COSTAR evaluation with analysing its performance using the Iris dataset, and then we show the performance on the activity recognition datasets, namely, OPPORTUNITY, WISDM, and SPAD.

5.10.2 Iris dataset

In this part, we analyse COSTAR performance closely with the well-known Iris dataset. Iris dataset contains sequential data representing different types of Iris plant. Evaluation for the Iris dataset starts with testing concept evolution of novel and recurring class instances on a multi-class underlying concept. The next step is to test the detection of the appearance of multiple novel concepts. This evaluation shows the capability of COSTAR to detect one or more novel concepts regardless of its spatial position relative to the BLFW. Figure 5.11 shows the layout of the three Iris classes: Setosa, Virginica, and Versicolor. Table 5.5 depicts the performance metrics against each experiment for the Iris dataset. In this table, the term FA indicates the number of alarms that declare the **F**irst **A**ppearance of a novel concept, while Nov_{Num} is the total number of instances detected as novel, including First Appearance instances and recurring instances that follow the First Appearance. Unk_{Num} is the total number of

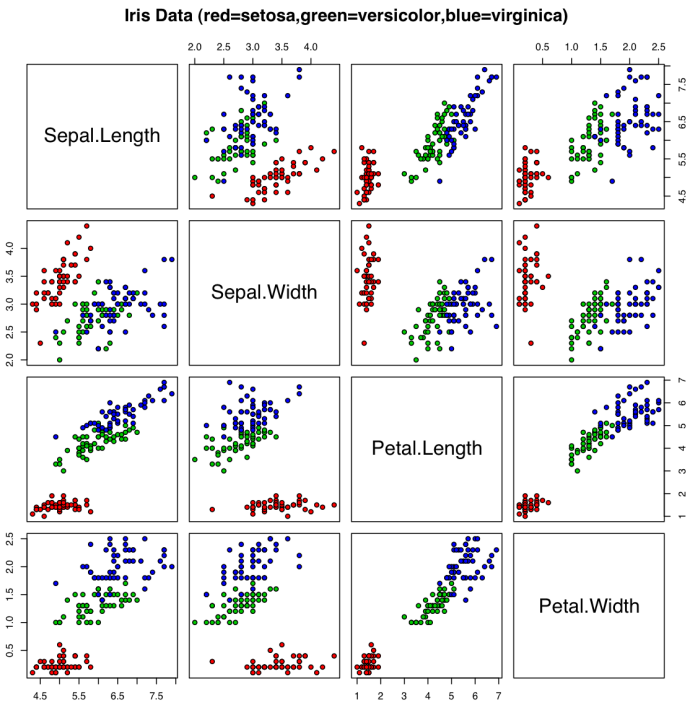


Figure 5.11: Iris Dataset Visualisation with the R Package

instances that are declared as unknown. Each class in the Iris dataset contains 50 instances. Considering the small size of the classes in this dataset, the chunk size is set to 5 in all of the experiments. The initial stable weight of a novel concept is tuned in interval (7–10 instances). From the characteristics of the BLFW, Dmax inside clusters ranges between (0.4–0.9). Therefore, DRAB height is tuned manually in the same interval. CPur is noted to be 100% in all runs. The pure clusters indicate the effectiveness of the online clustering and sliding window technique in separating activities.

As shown in Figure 5.11, the Versicolor class resides spatially inside the global decision boundary of the BLFW of Setosa and Virginica. Despite its location, COSTAR can successfully detect the appearance of the novel concept of Versicolor as well as its recurring instances. The table demonstrates the high performance of COSTAR in detecting novel concepts across all classes in the Iris dataset regardless of the position of the class from the underlying BLFW.

The second part of this table examines the COSTAR ability to detect more than one novel concept. It also shows the efficiency of the adaptation process to assimilate the newly detected concept and then detect its recurring instances. COSTAR also can distinguish between recurring instances of newborn concept and instances for another novel concept. We train the BLFW on only one class, and then we test the data that contains the other two novel concepts. Two alarms for First Appearance (FA) are triggered in these experiments for

Table 5.5: COSTAR Performance on the Iris Dataset

BLFW	Novel	FA	Nov Num	Unk Num	Acc %	Recall %	Specf %	Prec %	FOR %	FDR %
Setosa & Virginica	Versicolor	1	49	1	99.3	100	100	100	0	0
Setosa & Versicolor	Virginica	1	50	0	100	100	100	100	0	0
Virginica & Versicolor	Setosa	1	50	0	100	100	100	100	0	0
Versicolor	Setosa & Virginica	2	100	0	100	100	100	100	0	0
Virginica	Setosa & Versicolor	2	102	0	98.7	100	96	98.1	0.04	1.9
Setosa	Virginica & Versicolor	2	98	0	98.7	98	100	100	0	0

each novel concept. Once a novel concept is detected, it adapts and extends BLFW instantaneously for the further detecting of recurring instances of the novel concept. The accuracy here indicates the high performance detection of both recurring and novel concepts.

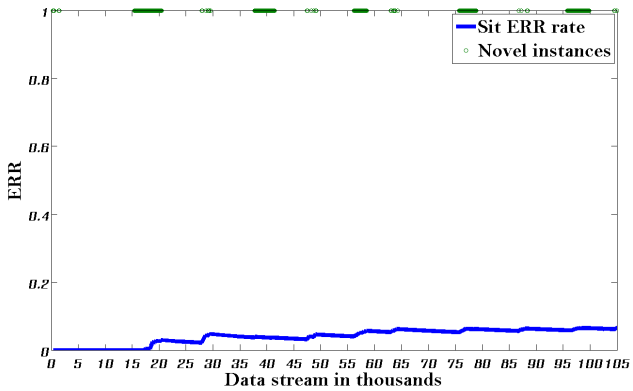
The evaluation of COSTAR on the Iris dataset shows its efficiency in detecting the first appearance of novel concepts and their recurring occurrences. It also demonstrates the ability of COSTAR to detect multiple novel concepts in the stream. We then evaluate COSTAR performance on more challenging benchmarked datasets for activity recognition that are collected from streaming sensory data.

5.10.3 OPPORTUNITY dataset

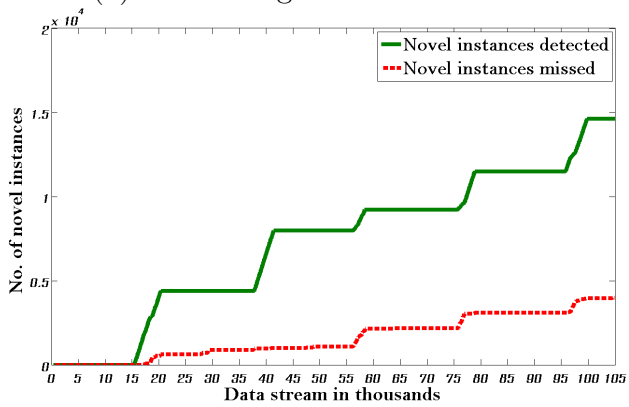
In this section, we move forward to test COSTAR on real life activity recognition data streams collected from wearable sensors. The dataset consists of data for the four atomic activities of “Sitting”, “Walking”, “Standing”, and “Lying” represented with 110 features. To be able to evaluate the COSTAR performance on data streams with concept drift, in the following experiments, the data used for building BLFW and testing is from different users. Evaluation of the detection of a novel concept is performed for each activity. The sampling rate for the OPPORTUNITY dataset is 20 Hz. We set the chunk size for all experiments to 50 instances (2.5 seconds) and the weight of stable novel concepts to 300 instances. DRAB height is tuned based on $Dmax$ values of BLFW exiting clusters in the interval (1800–3000). Figures 5.12, 5.13, 5.14, and 5.15 show various performance metrics for detecting each activity as

novel. Sub-figure (a) shows the ERR rate of COSTAR up to a certain point in the stream for each activity type. ERR is the total percentage of misclassified instances (%) in the stream that includes Fp, Fn, and unknown. For instance, in the experiment to evaluate COSTAR in detecting the "Sitting" activity in Figure 5.12(a), the average ERR rate from the beginning of the stream to 10 thousand data stream points is the value of Y at $X = 10$, around 8%. The arrival of the novel activity instances is marked at the top border in Sub-figure (a) in all figures. The number of correctly detected novel instances (Tp) and missed novel instances (Fn) are displayed in Sub-figure (b) for each activity. Whereas Sub-figure (c) shows the correlation between the classification of existing classes correctly (Tn) and false discovery of existing class as novel (Fp). In general, the activities of "Sitting" and "Lying" achieved higher performance over "Walking" and "Standing". In Figures 5.12(b) and 5.13(b), the number of novel instances detected (Tp) increases steadily over time. Whereas, the novel instances missed (Fn) are at the lower bound along the stream with a slow increase of a small constant rate. The error graph for the "Lying" activity, Figure 5.13(a), starts with a high error rate and continues until the first correct detection of the novel activity. Once the novel concept is declared, recurring instances are correctly classified and thus the error rate drops over time. The specificity relationship in Figure 5.12(c) and 5.13(c) indicates the COSTAR ability to identify existing instances correctly. The rate of FP is slightly increasing by a very small amount, along the stream which indicates a high percentage of specificity.

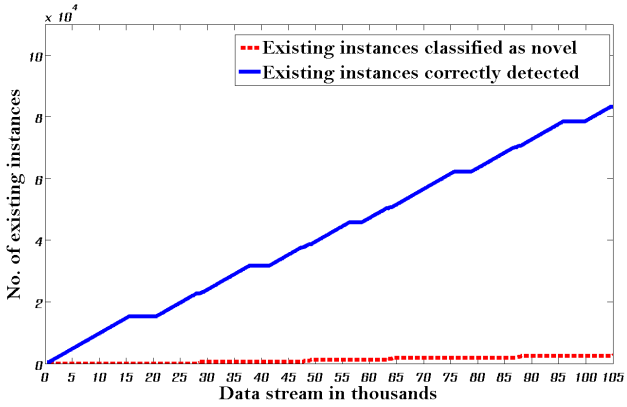
"Walking" and "Standing" activities are more challenging. Many reasons are behind the confusion between theses two activities. Both activities are strongly correlated and interleaving. That means combining and switching of the two activities are frequently occurring. Also, novel concepts that represent these activities are relatively big with diverse patterns embedded inside. For instance, the "Walking" activity might contain patterns of strolling, jogging, and normal pace walking. The strolling pattern of "Walking" is commonly interleaved with "Standing" (as pauses) while "Walking". Figure 5.14(a) shows the frequent occurrences of the "Walking" activity along the stream. In analysing the ERR rate in detecting the "Walking" activity, the figure shows a high rate at the beginning of the stream. That indicates misclassifying of novel instances (Fn). The percentage decreases once COSTAR detects the appearance of a novel concept. Figure 5.14(c) shows a high detection rate for existing instances for the "Walking" activity. However, Figure 5.14(b) shows an overlapping percentage for both detecting and missing the novel concept instances



(a) ERR along the Data Stream



(b) Detection of Novel Instances

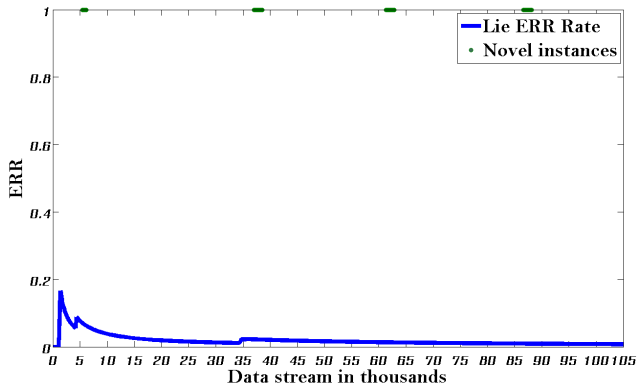


(c) Detection of Existing Instances

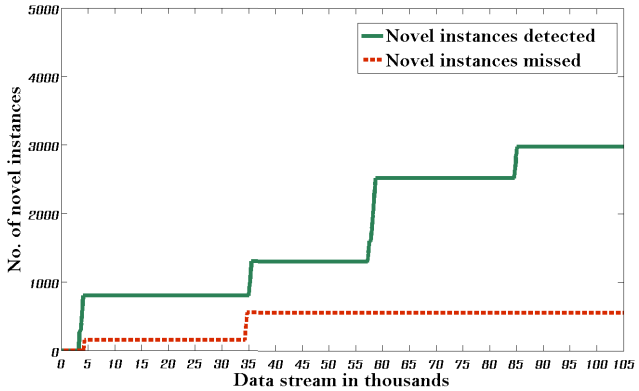
Figure 5.12: COSTAR Performance in Detecting "Sitting" Activity

along the stream. The main reason for the ERR rate in "Walking" is the Fn rate (misclassifying novel data as existing).

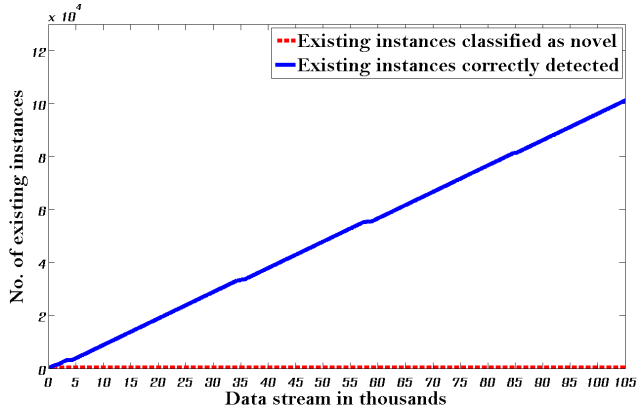
The detection of the "Standing" activity is also challenging. As shown in Figure 5.15(a), the "Standing" activity appears very frequently in the stream. Periods of "Standing" are interrupted with other activities. The increase of the Tp rate over time, Figure 5.15(b), indicates COSTAR's good performance in detecting "Standing" as novel (Recall). Figure 5.15(c) shows an opposite



(a) ERR along the Data Stream



(b) Detection of Novel Instances

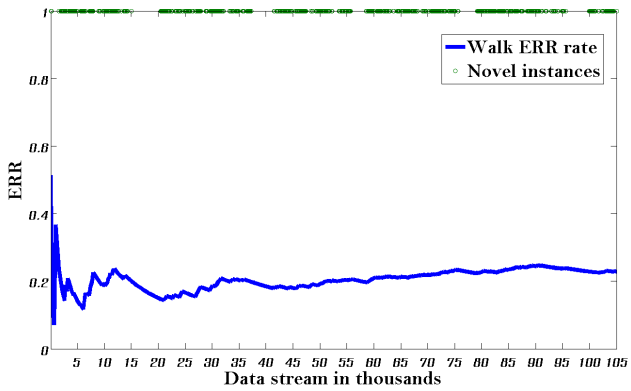


(c) Detection of Existing Instances

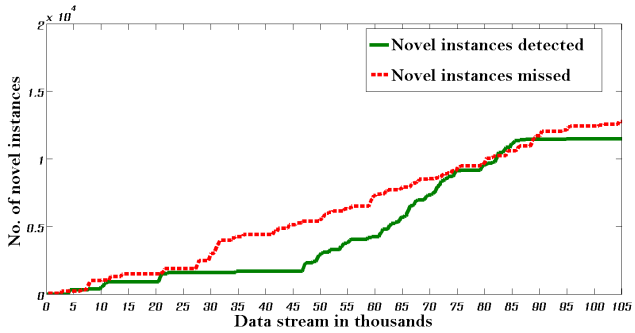
Figure 5.13: COSTAR Performance in Detecting "Lying" Activity

behaviour for the false detection of existing instances as novel. The high rate of Fp is the main cause of high ERR rate in detecting "Standing" activity.

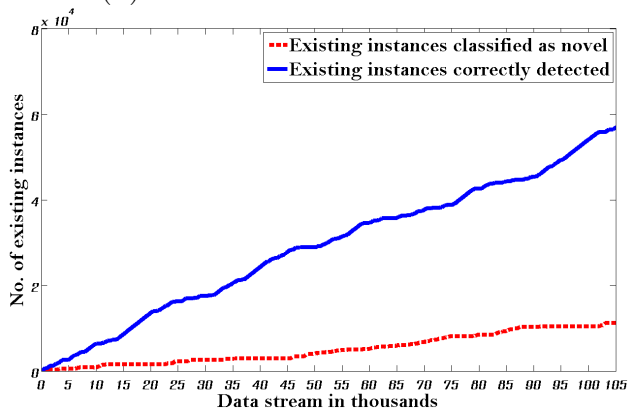
In order to visualise the OPPORTUNITY dataset, we have applied principal components analysis and transformation of the data. Dimensionality reduction is accomplished by WEKA PCA [WF05] by choosing enough eigenvectors to account for 95% of the variance in the original data. Therefore, the reduced dimension OPPORTUNITY sample data can be visualised with a 3D graph as displayed in Figure 5.16. The graph shows the clear overlapping



(a) ERR along the Data Stream



(b) Detection of Novel Instances

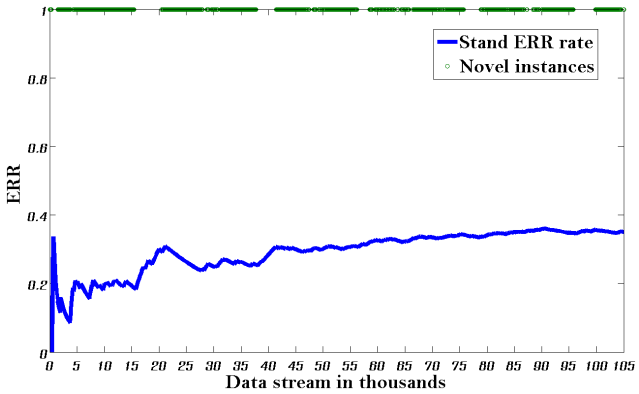


(c) Detection of Existing Instances

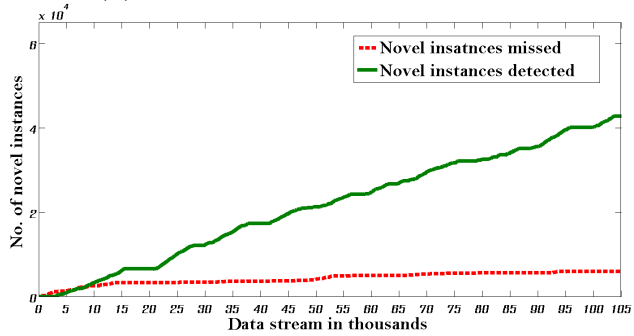
Figure 5.14: COSTAR Performance in Detecting "Walking" Activity

between "Standing" and "Walking" activities that would justify the confusion in decision between the two activities. It also shows the location of "Sitting" activity inside global decision boundary between "Lying" from one side and "Standing" and "Walking" from the other side.

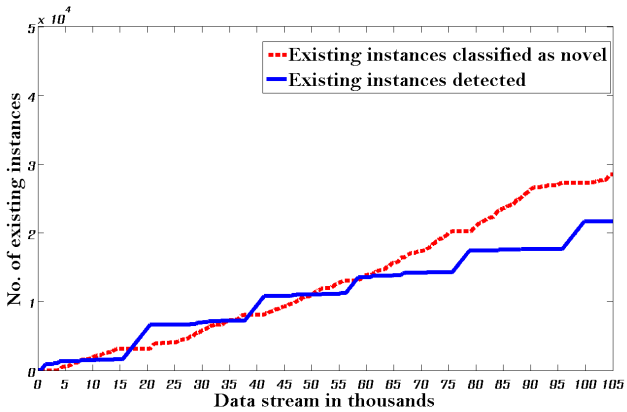
In Table 5.6, we summarise the performance of COSTAR for recognising each activity in the OPPORTUNITY dataset. The BLFW for all experiments consists of all activities except the tested novel activity. Data used to build the BLFW and test data are from different users to ensure the occurrence of



(a) ERR along the Data Stream



(b) Detection of Novel Instances



(c) Detection of Existing Instances

Figure 5.15: COSTAR Performance in Detecting "Standing" Activity

concept drift. Each activity has its own weight in the test data. The activity weight is the percentage of instances that represent the activity in the stream.

The accuracy of distinguishing between existing and novel when "Lying" is the novel activity attains the best performance with more than a 99% accuracy rate. Despite the small weight of the "Lying" class (only 5%), COSTAR can effectively detect both existing and novel activities with recurring occurrences. COSTAR also attains an accurate recognition, of 93.1%, in the case of a bigger class of "Sitting", with weight 24%. The precision in detecting the novel activity of "Sitting" is 84%, mainly because of the false detection rate. The

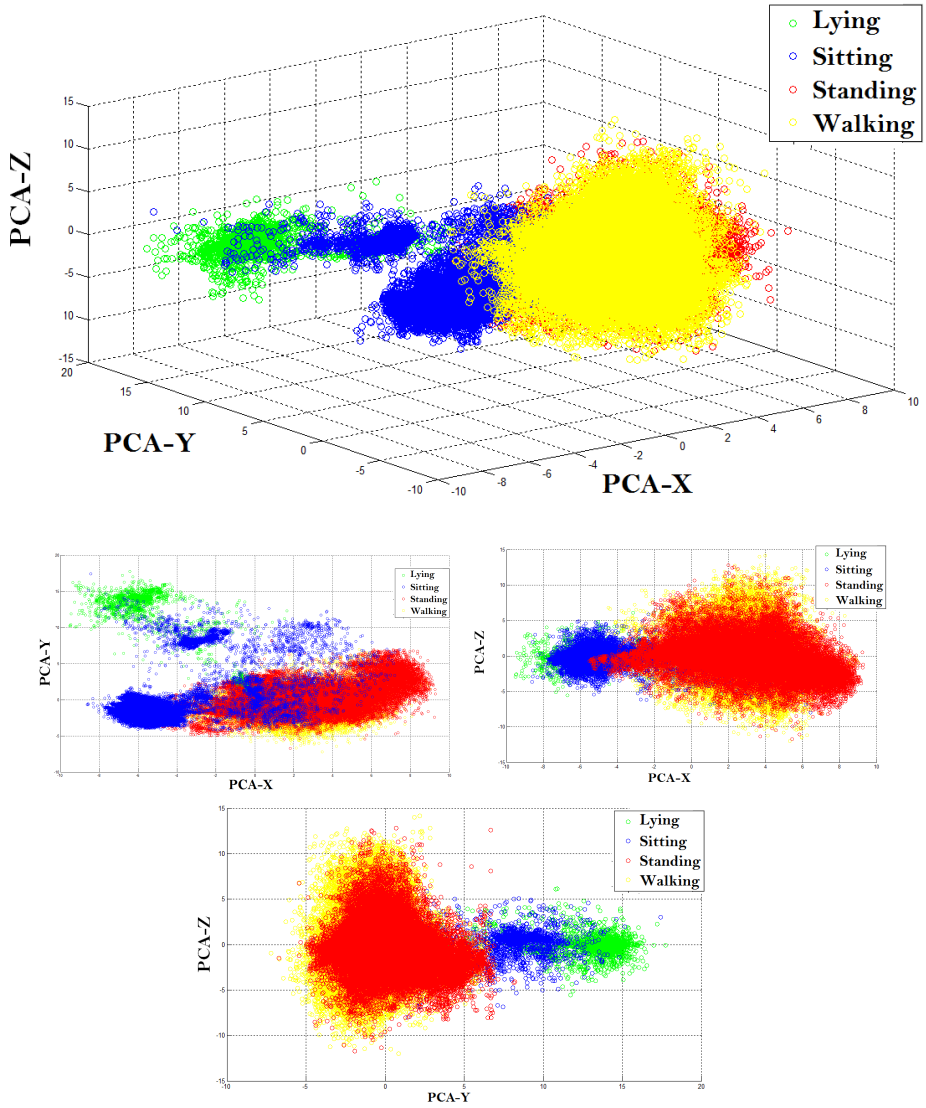


Figure 5.16: Visualisation of the OPPORTUNITY Dataset

accuracy in distinguishing between existing and novel is lower with "Standing" and "Walking" activities.

Although the "Walking" activity has a similar weight of 26% as "Sitting", its detection accuracy of 70.4% is lesser. The false detection rate of the "Walking" activity is high due to the confusion with the "Standing" activity. The low recall percentage indicates the misclassification of many novel instances as existing. Thus, the recall percentage that measures COSTAR sensitivity drops to $\simeq$ 17 % for "Walking", while the false-discovery rate (FDR) reaches 40%. The "Standing" activity in this dataset is diverse and heavy with a weight of 46%. The accuracy in detecting novel and existing activities in the case of "Standing" is 61.3%. Although, the recall percentage reaches more than 85%, opposite to "Walking" performance, specificity is dropped to $\simeq$ 40% due to the high fall-out rate of 56.8%.

Cluster purity, which measures the percentage of major labels inside declared clusters, achieves more 90% in all runs. Cluster purity indicates the average purity of the cluster when declared. The high purity percentage demonstrates the effective selection of parameters, specifically the chunk size and stable size. The purity in the"Walking" and "Standing" clusters is lower than other activities, which indicates that the overlapping between these two activities results in less pure clusters.

Table 5.6: COSTAR Performance with the OPPORTUNITY Dataset

Novel activity	Acc	Recall	Specf	Prec	FOR	FDR	CPur
Sitting (%)	93.1	78.5	96.8	84.0	3.2	16.0	98.0
Lying (%)	99.2	86.5	99.8	94.0	0.2	6.0	97.3
Walking (%)	70.4	17.2	95.7	61.9	4.3	38.1	93.2
Standing (%)	61.3	87.8	43.2	60.0	56.8	40.0	93.4

The analysis and discussion on the OPPORTUNITY dataset show the COSTAR ability to distinguish between existing and novel activities in data streams with concept drift. The error rate might be high at the beginning of the stream until the first appearance of the novel activity is successfully detected. The error rate decreases gradually afterwards, which indicates an efficient adaptation of the model to recognise recurring instances of the detected novel concept. The evaluation demonstrates COSTAR ability to recognise activities that are not well-presented in the stream, i.e., small sized activities, such as "Lying" in OPPORTUNITY. The experiments also show the trade-off between recall and precision, especially in the "Walking" and "Standing" activities. We can also conclude that COSTAR attains its best performance in detecting novel activities that are static, such as "Sitting" and "Lying". In this scenario, the BLFW might contain other static activities. Nevertheless, COSTAR can distinguish between the appearance of other novel static activities and existing ones, including static and dynamic activities. The nature of the interleaved and interrelated activities of "Walking" and "Standing" imposes more challenges in the recognition. Thus, COSTAR accuracy in recognising these activities is less than other activities.

5.10.4 WISDM dataset

In this section, we discuss COSTAR performance with another activity recognition dataset, but collected from mobile accelerometer sensor. The WISDM dataset contains five annotated activities with only three features: x, y, and z accelerometer components. Data is collected from many users and contains more than a million instances. COSTAR as a cluster-based technique processes each activity as a cluster that contains a set of sub-clusters that represent different patterns inside the cluster. Thus, we combine "Upstairs" and "Downstairs" patterns in the dataset and integrate them into one cluster of "Stairs". Table 5.7 shows data distribution among users samples with various activities performed by each.

Table 5.7: Sample of WISDM Data Distribution Across Users and Activities

Dataset	Size	Walking	Jogging	Standing	Sitting	Stairs
u2	23,525	•	•			
u8	29,761	•	•	•		
u20	21,033			•	•	
u27	34,957	•	•	•	•	•
u29	34,957	•	•	•		•
u30	12,756			•	•	•
u33	29,452		•	•	•	
u35	22,394	•	•	•	•	
u36	32,108	•	•	•	•	•

In order to deploy COSTAR on WISDM and ensure concept drift, we choose to build the BLFW with data from users who have not performed the target novel activity. For each row in Table 5.8, we aim to detect a novel activity with a combination of users. For instance, to detect the "Stairs" activity (which includes both "Upstairs" and "Downstairs"), we build a model with data from user 20 who performed "Standing" and "Sitting" activities (as per Table 5.7). Then, we evaluate COSTAR performance in detecting "Stairs" with data from user 30 who performed same BLFW activities in addition to the novel activity ("Stairs"). Performance metrics are reported in Table 5.8 for each novel activity across the different users. The set of parameters used for WISDM evaluation is as follows: chunk size=10, stable size=20, while DRAB height is in the interval (0.5–3).

COSTAR achieves its highest accuracy in detecting the "Sitting" activity with an accuracy of 99.8%. The remaining 0.2% of instances are declared as unknown. Although the weight of the "Sitting" class in the incoming stream is only 7%, COSTAR can detect the appearance of the novel concept and its recurring instances."Sitting" is a static activity that does not require body

Table 5.8: COSTAR Performance on Different Users of the WISDM Dataset

Novel	BLFW	Test	Acc	Recall	Specf	Prec	FOR	FDR
Sitting (%)	u29	u36	99.8	100	100	100	0	0
Standing (%)	u2	u8	81.1	74.2	85.4	76.3	14.5	23.7
Walking (%)	u33	u35	83.5	81.4	84.5	70.5	15.5	29.5
Stairs (%)	u20	u30	71.3	73.9	96.9	96.7	3.1	3.2
Jogging (%)	u36 (no-jogging)	u27	84.5	69.6	93.0	84.3	7.0	15.6

movement. Therefore, the detection of the "Sitting" activity is attained with high accuracy despite its small size.

The distinction between existing activities and novel activity of "Standing" is more challenging, especially when training the BLFW with data contains "Walking" and "Jogging" activities, as in the user 2 data. Both "Walking" and "Jogging" are tightly related to "Standing", which makes it hard to detect the appearance of the "Standing" activity. Periods of "Walking" or "Jogging" would be overlapped with pauses of "Standing". COSTAR can distinguish between "Standing" as a novel activity and existing "Jogging" and "Walking" activities with an accuracy of 80%. However, both the false detection rate (FDR) and fall out rate (FOR) are at high percentages (23.7 and 14.5 respectively). That shows the confusion occurs in detecting "Standing" activity as existing and also the detection of existing instances as novel. The confusion matrix of detecting "Standing" activity is illustrated in Figure 5.17(a). Novel, in all matrices listed in this figure, shows the combination between instances declared as novel with first appearance (FA) and the recurring instances. The detection of recurring instances is based on the continuously adapted BLFW that integrates the novel activity. The results show that 74% of novel instances of "Standing" are correctly detected as novel. The first reason for confusion is because of the misclassification of the "Standing" activity as existing. In this experiment, the ambiguity between "Standing" and "Walking" results in more than 23% of the novel instances misclassified as "Walking". On the other hand, the existing "Walking" instances are correctly classified as existing. Another reason of confusion is in detecting the "Jogging" activity as novel ("Standing") for more than 22% of "Jogging" data.

We face the same challenges when detecting the novel activity of "Walking". The BLFW is built with data from the user 33 that contains "Jogging", "Standing", and "Sitting" activities. Both the false detection rate and fall out

True Label	Classified as: Existing		Novel (Standing)	Unknown
Jogging	78%		22%	0%
Walking	100%		0%	0%
Novel (Standing)	3%	23%	74%	0%
	Jogging	Walking		

(a) "Standing" Activity

True Label	Classified as: Existing			Novel (Walking)	Unknown
Standing	87%			13%	0%
Sitting	87%			13%	0%
Jogging	57%			4%	39%
Novel (Walking)	24%	0%	3%	73%	0%
	Standing	Sitting	Jogging		

(b) "Walking" Activity

True Label	Classified as: Existing		Novel (Stairs)	Unknown
Sitting	91%		9%	0%
Standing	100%		0%	0%
Novel (Stairs)	5%	21%	74%	0%
	Sitting	Standing		

(c) "Stairs" Activity

True label	Classified as: Existing				Novel (Jogging)	Unknown
Walking	100%				0%	0%
Sitting	86%				2%	12%
Standing	100%				0%	0%
Stairs	94%				6%	0%
Novel (Jogging)	0%	0%	0%	59%	41%	0%
	Walking	Sitting	Standing	Stairs		

(d) "Jogging" Activity

Figure 5.17: Confusion Matrices on the WISDM Dataset

rate are high (similar to results in detecting the "Standing" activity). The confusion between "Walking" and "Standing" is the main reason for the fall out rate as shown in Figure 5.17(b). The confusion matrix also shows that 39% of the "Jogging" activity is declared unknown because of uncertainty in recognition. Also, instances of "Standing" and "Sitting" are falsely detected as novel ("Walking").

We examine COSTAR performance in detecting the "Stairs" activity. Figure 4.11 in Chapter 4 shows the overlapping between the large class of "Stairs"

and other classes. The BLFW that is built from the user 20 data contains both "Sitting" and "Standing" activities. The COSTAR ability to distinguish between the novel activity of "Stairs" and the existing activity is 71.3%. The sensitivity of detecting the novel activity is 74%, indicating a confusion with existing. In the confusion matrix of the "Stairs" activity, shown in Figure 5.17(c), it is clear that the main ambiguity occurs between "Standing" and "Stairs". This can be justified by the natural pauses of "Stand" that occur while climbing up and down the stairs.

We evaluate the detection of the novel activity of "Jogging" with a BLFW that contains all other activities: "Walking", "Sitting", "Stairs", and "Standing". The two activities of "Jogging" and "Stairs" are naturally similar. The accuracy in distinguishing between all existing activities and "Jogging" as novel is 84.5%. The sensitivity of detecting the novel activity in terms of recall percentage is 69.6%. As illustrated in the confusion matrix in Figure 5.17(d), the reason for the lower recall rate is the confusion between "Stairs" and "Jogging". Despite the efficiency of COSTAR to detect existing activities, the correct detection of a novel "Stairs" activity is only 41% of the instances representing the novel activity. In general, COSTAR can successfully distinguish between existing and novel activities in data streams with concept drift using WISDM dataset. The best performance of COSTAR is attained when recognising the appearance of "Sitting" activity and its recurring activities. The activity of "Stairs" is the most challenging to recognise because of the overlapping with the "Standing" activity.

5.10.5 SPAD dataset

We also evaluate COSTAR performance on the SPAD dataset. In all experiments, we train the model on all data except the novel one. The dataset contains four activities of "Walk", "Drive", "Run", and "Stay *still*". Table 5.9 represents the performance of COSTAR with each of the four activities. The SPAD dataset has a sampling rate of 5 Hz. We tune the window size between (1–5) sec. The DRAB height ranges from (0–0.5) based on $Dmax$ values in the BLFW. Because of the low sampling rate, we set the stable size to 5 in all experiments.

As shown in the reported results in Table 5.9, "Still" achieves the best accuracy among all activities. This is similar to the analysis of the "Sitting" activity in the WISDM and OPPORTUNITY datasets. Minor ambiguity occurs between "Drive" and "Still" as illustrated in Figure 5.18(a). It is more challenging to recognise the drive activity. "Drive" might be confused with

Table 5.9: COSTAR Performance on the SPAD Dataset

Novel	Acc	Recall	Specf	Prec	FOR	FDR
Still (%)	96.4	97.3	98.3	94.3	1.7	5.7
Walk (%)	83.8	71.7	98.5	98.3	1.5	1.7
Drive (%)	94.3	73.6	99.1	90.1	0.9	9.9
Run (%)	86.8	63.2	90.0	46.3	10.0	53.7

"Still" when collecting data from a mobile accelerometer sensor, the case in the SPAD dataset. Also in SPAD, the number of instances that represent the activity of "Drive" is relatively small (7% of the total number of instances). Despite these challenges, COSTAR can efficiently recognise the "Drive" activity with an accuracy of 94.6%. The recall percentage for detecting the "Drive" activity is low which indicates low sensitivity in classifying true novel instances as novel. The percentage of 73.6% refers to the percentage of data that is correctly classified as novel ("Drive") among all novel instances. The FDR rate indicates that 9.9% of instances are declared as novel, while they are actually existing instances and misclassified as novel. The confusion matrix for detecting "Drive" is shown in Figure 5.18(b). The results in this matrix show the confusion that mainly occurs between "Still" and "Drive". 21% of the instances that represent the novel "Drive" activity is misclassified as the "Still" activity.

The "Run" and "Walk" activities are more difficult to recognise. The "Run" activity has also a relatively small size with a weight of 12% along the stream. Moreover, it spans a large area and resides in the middle of all other activities, as shown in Figure 4.12 in Chapter 4. From Table 5.9, the FDR in "Run" indicates more than half of the instances that were declared as novel are existing. The matrix in Figure 5.18(c) confirms this by indicating the main ambiguity that happens between "Run" and "Walk". Specificity, 16% of the "Walk" activity is classified as novel ("Run"), while 37% of the novel instances of "Run" are misclassified as "Walk".

When detecting the novel"Walk" activity, we found that COSTAR can efficiently handle the recognition of existing activities. The overall accuracy in detecting novel and existing is 83.3% as shown in Table 5.9. COSTAR can recognise the appearance of the "Walk" activity and its recurring instances with an accuracy of 72% as shown in the confusion matrix in Figure 5.18(d). There are still 28% of instances that are confused with the "Run" activity.

True Label \ Classified as	Existing			Novel (Still)	Unknown
Walk	97%			1%	2%
Run	99%			0%	1%
Drive	93%			7%	0%
Novel (Still)	0%	0%	3%	97%	0%
	Walk	Run	Drive		

(a) "Still" Activity

True Label \ Classified as	Existing			Novel (Drive)	Unknown
Walk	98%			0%	2%
Run	99%			0%	1%
Still	93%			3%	4%
Novel (Drive)	3%	0%	21%	76%	0%
	Walk	Run	Still		

(b) "Drive" Activity

True Label \ Classified as	Existing			Novel (Run)	Unknown
Walk	84%			16%	0%
Still	100%			0%	0%
Drive	100%			0%	0%
Novel (Run)	37%	0%	0%	63%	0%
	Walk	Still	Drive		

(c) "Run" Activity

True Label \ Classified as	Existing			Novel (Walk)	Unknown
Run	100%			0%	0%
Still	98%			2%	0%
Drive	98%			2%	0%
Novel (Walk)	28%	0%	0%	72%	0%
	Run	Still	Drive		

(d) "Walk" Activity

Figure 5.18: Confusion Matrices on the SPAD Dataset

From the analysis of COSTAR on the SPAD dataset, the results show an accurate recognition of existing activities in all experiments. The detection of novel activities achieves the best accuracy with "Still" and "Drive" activities. The detection of the dynamic activities of "Run" and "Walk" achieves a good accuracy as well. These two activities are tightly related, therefore they cause more difficulty in recognition than other activities.

After discussing the overall performance of COSTAR with the benchmarked datasets, in the following we will discuss the parameters and reasons for active learning that is triggered to adapt the BLFW.

5.10.6 Active learning

Active learning in COSTAR is provoked in two cases, *unknown* data and the *first appearance* of a novel activity (FA). The First Appearance includes cases of Tp for first occurrence of the novel concept, Fp for existing instances misclassified as novel, and recurring instances that are incorrectly recognised as entirely novel concepts. In Figure 5.19, the results show the two metrics that evaluate active learning performance. ALRate represents the average number of inquiries triggered for every 10k instances in the stream. The smaller the ALRate, the more efficient is our technique. ALPoints refers to the total number of instances that require active learning and thus feed back to adapt the BLFW. The two metrics include the two cases of active learning.

In the OPPORTUNITY dataset, the rate of active learning needed is very low in the "Lying" and "Sitting" activities. There is a small percentage of *unknown* instances noted in these two activities. Thus, the low active learning rate and small percentage of ALPoints correspond mostly to the FA case. The rate is slightly higher in the "Walking" and "Standing" activities as they are more challenging activities to be recognised. Yet, the ALRate is very small at 3.8 and 2.6 inquiries per 10k instances in "Walking" and "Standing" respectively.

Activity	ALRate	ALPoints
Sitting	1.7	3.4%
Lying	0.3	0.4%
Walking	3.8	7%
Standing	2.6	7%

(a) OPOORTUNITY dataset

Activity	ALRate	ALPoints
Still	1.5	1.7%
Run	1.5	0.1%
Drive	3	3.2%
Walk	12	1.7%

(b) SPAD dataset

Activity	ALRate	ALPoints
Stairs	6.5	6.7%
Jogging	0.3	9.8 %
Walking	3.1	4.4%
Standing	22.8	9%
Sitting	0.3	0.1%

(c) WISDM dataset

Figure 5.19: Active Learning with Different Datasets

In the SPAD dataset, the "Run" activity provokes active learning only for the FA of a novel concept when it arrives. We also noted COSTAR does not declare any data as *unknown* when detecting the "Walk" activity. More than half of the ALRate in detecting "Walk" is because of cases of Fp, i.e., when existing instances are misclassified as novel. The next reason is recurring instances that are defined as new novel concepts.

The rate of active learning in the "Walk" activity is higher than others, 12 inquiries in 10k data instances, which is about 33 minutes for the SPAD dataset. Despite the high rate, the percentage of points that require active learning is still low. The active learning technique in COSTAR implements a batch approach. That means, whenever conditions are satisfied, the batch of instances triggers active learning. Therefore, the size of the batch of instances for active learning is either the stable size or the buffer size. In detecting the "Walk" activity in SPAD, many inquiries are triggered, all with small size. Therefore, the total amount of points remain small despite the high ALRate.

The WISDM dataset results show a low ALRate in "Jogging", "Sitting", and "Walking" activities. The high percentage of the ALPoints rate in "Standing" is because of the high percentage of Fp as per the aforementioned discussion on the confusion matrix. Frequent "Standing" triggers are caused by the frequent alternation between "Standing" and other activities. The alternation between activities is a reason for the Replace and Reset SPRK operations in the CVC. When the data is released with incomplete monitoring information, the declared decision is inaccurate because of the incomplete information. The activity of "Jogging" does not have the same characteristics as "Standing". Despite the low rate of active learning in "Jogging", the size of batches that require active learning is big and therefore the ALPoints are as high as 9.8%. We can conclude that, most results show an efficient performance of the batch active learning technique with a low ALRate in most of the cases.

In conclusion, we analysed the performance of COSTAR in detecting novel concepts and their recurring occurrences on benchmarked datasets. The analysis shows COSTAR ability to distinguish between existing and novel activities evolving in the stream. In general, activities that are static are easier to be detected even if they appear less frequent in the stream. Interrelated and interleaved activities are more difficult to recognise. Nevertheless, COSTAR still can recognise these activities with a good accuracy. The low percentage of active learning needed depicts the efficiency of COSTAR to ask user for a small subset of data that is most informative.

5.11 Summary

The number of activities in the data streams changes over time. New activities might emerge; irrelevant activities might disappear. Training the learning model in activity recognition with a static number of activities is impractical and inefficient especially in a streaming environment. The capability of the learning model to adapt for extension and prune continuously to reflect

the changes of activities in data streams is essential for accurate and efficient recognition performance.

In this chapter, we have proposed, developed, and evaluated our COSTAR technique for detecting concept evolution in evolving activities. COSTAR applies a continuous learning approach for monitoring the evolution of activities in the stream and thus detects the expected changes. The detected changes in COSTAR include novel activities, recurring novel activities, and abandoned activities. COSTAR comprises two components: the Cohesion Validation Component (CVC) and OBSERVER. While the CVC checks the dependency in the stream, OBSERVER monitors the movement of data over time. The learning model is adapted incrementally and continuously to reflect the detected changes. COSTAR also addresses the scarcity of labeled data by incorporating an active learning approach for labelling only the most uncertain data in the steam.

The evaluation of COSTAR on benchmarked datasets demonstrated its ability to detect novel concepts and its recurring instances with learning model that contains multi-concepts. The recognition performance showed its efficiency in detecting the small sized classes that are mostly overlooked and thus misclassified. The results also showed the efficiency of the model adaptation technique to refine the model with the expected changes. COSTAR can distinguish with good accuracy between concept drift and novel concepts. We discussed also in the results the active learning percentage for different activities. The rate of active learning inquiry is noted to be small with different experiments. That indicates efficient performance of the batch active learning approach.

COSTAR parameter tuning is crucial for obtaining a good performance. We aim in our future work to automate the selection of COSTAR parameters for more efficient recognition. Moreover, we aim to combine STAR and COSTAR in a single framework, for both personalisation and adaptation in a single recognition framework.

Chapter 6

Conclusion

Activity recognition has been a progressive research area in recent years. Researchers have addressed many challenges in activity recognition, while many other challenges are yet to be resolved. We address in this dissertation a set of crucial and paramount issues that arose from activity recognition. The main focus of this dissertation is on activity recognition in ubiquitous computing and streaming environments. We contributed to the research by introducing adaptive techniques to cope with the evolving nature of sensory data in streaming and ubiquitous environments. In the previous chapters in this dissertation, we have proposed, developed, and evaluated our adaptive techniques for personalisation and adaptation, for activity recognition in evolving data streams. In this chapter, we conclude this dissertation with a summary of the research and and its contributions. Then we discuss possible future directions based on the thesis contributions.

6.1 Research Summary

Activity recognition has become an emerging field in the areas of pervasive sensory data processing and ubiquitous computing. Recognition of activities opened the door to many applications in wellness such as elder care support and fitness monitoring, surveillance, marketing, and crowdsourcing applications. There is a large variety of low cost sensors that are embedded on mobile phones, body worn, or ambient. The availability of these sensors gives rise to the development of techniques that can effectively recognise users' activities and thus contribute to many applications in activity recognition. Activity recognition intuitively deals with sensory data that is continuously streaming from different sensing sources over time. The process of recognising activities includes three key components: data collection, modelling, and actual recognition. Historical data is collected from different sensors. The sensory data

is modelled to represent activities. Then, the model is deployed for recognising incoming activities. Although much research has been directed towards activity recognition, many research gaps are still open and yet to be addressed.

Data applied for building the learning model in activity recognition is typically collected from a small group of users. Learning techniques in activity recognition rely strongly on the prior knowledge of the collected and modelled data. However, people perform activities in different ways. Thus, the learning model, which represents activities based on data collected from some users, has to be tuned and refined to suit other users. Moreover, it is impractical to assume that the number of activities is static across users especially with evolving data streams. Novel activities may emerge over time, while irrelevant activities might be abandoned. The learning model has to adapt continuously to reflect the appearance and disappearance of activities along the evolving streams and therefore demonstrate an accurate representation of the activities. The performance of state-of-the art techniques in activity recognition are also tightly related to the sensing scheme. Thus, changes in sensor location, type, and orientation would directly affect the learning performance. Lastly, techniques in activity recognition have to consider essentially the known-issue of the scarcity of labelled data especially when dealing with data streams.

The focus of the research presented in this dissertation is to develop learning techniques that address the aforementioned challenges. Thus, we proposed, developed, and evaluated adaptive techniques that refine and adapt the learning model contentiously while sensory data evolves. The developed techniques address the personalisation and adaptation challenges in activity recognition by applying incremental and continuous learning for refining and adapting the learning model. Our techniques have to be accurate, robust, flexible, and efficient to achieve real time accurate recognition in a streaming and ubiquitous environments. The techniques are independent on the type and location of sensors that are deployed for data collection. In evaluating the performance of our techniques, we use various publicly available datasets. Each dataset has its own scheme and characteristics. The data in these datasets is collected from various sensors including mobile sensors, body worn sensors, and ambient sensors. The developed techniques address the scarcity of labelled data by incorporating active learning to label only a small subset of data that is most informative and with lowest cost.

We first introduced, in Chapter 3, our baseline framework that is built offline for modelling different activities. The modelling component is responsible for building the baseline framework. Thus, the modelling component processes the historical data to produce a flexible, compact, accurate, and

lightweight representation of activities for future recognition. A key characteristic of the baseline framework is its flexibility to be refined and adapted beyond modelling. The flexibility enables both personalisation and adaptation. The baseline framework presents the core for our three techniques for activity recognition represented in this dissertation.

The chapter also introduces CBARS, our first technique for activity recognition. The developed technique combines the modelling component with a recognition component. The recognition component builds an ensemble classifier, based on a hybrid similarity measure approach, for recognising activities from incoming data. The hybrid similarity measure approach brings different measures' perspectives together for enhancing the recognition accuracy across users. CBARS, comprising both modelling and recognition components, demonstrates its efficient recognition performance when evaluated on benchmarked activity recognition datasets. CBARS achieves its best accuracy enhancement, of other standard recognition techniques, when applied to noisy datasets across different users.

In Chapter 4 we introduce STAR, which extends CBARS by adding a personalisation component to the learning process. STAR enables incremental and active learning for model personalisation in streaming environments. The personalisation component implements incremental learning for refining the learning model with recent data streams to suit a user's personalised way of performing an activity. STAR is also capable of handling the streaming data in a ubiquitous environment. Given the scarcity of labelled data, especially in a streaming environment, STAR incorporates an active learning approach for labelling only a small amount of the most informative data to enhance the recognition performance. We deploy STAR on a mobile device to demonstrate its efficiency for real life applications in ubiquitous environment.

The adaptability of the recognition techniques is expanded in Chapter 5. While STAR performs model personalisation for existing activities to fit a specific user, COSTAR is developed for detecting completely new activities and forgetting abandoned ones. COSTAR enables continuous and active learning to monitor the evolution of concepts along data streams and adapts the model accordingly with the detected changes. The learning model in COSTAR is extended by assimilating detected novel activities, or pruned by removing no longer relevant activities. The continuous adaptation of the model enables the recognition of recurring occurrences of detected novel activities and reflects changes in data streams.

Experimental analysis is conducted for evaluating the efficacy of techniques proposed in Chapters 3, 4, and 5. The performance for CBARS, STAR, and

COSTAR is evaluated in terms of their accuracy, efficiency, robustness, and flexibility. We developed an efficient metric for different performance aspects and applied the evaluation to benchmarked datasets for activity recognition. The following section concludes the thesis by presenting a summary of the research contributions and outlining future research directions.

6.2 Research Contributions

The key objective of our research is to build adaptive techniques in activity recognition that enable continuous personalisation and adaptation with evolving data streams. We summarise the contribution of this thesis in fulfilling our research objectives and goals as follows:

- *Build an accurate, flexible, robust, and efficient learning model:* The initial process in any recognition technique is to build the learning model from historical data. In this dissertation, we build a fine-grained learning model that represents activities accurately. The representation of the activities is flexible to be updated and expanded, therefore our learning model enables incremental and continuous learning after building the model. It is equally important that the learning model is computationally efficient to perform in streaming environments for real time recognition. We demonstrated the efficiency of the learning model by deploying it in a ubiquitous and streaming environment. Dealing with sensory data increases the chance of contained noise. Thus, we developed approaches that aim to build a robust learning model that can efficiently handle noisy data and attain efficient recognition.

- *Develop an efficient recognition technique:* The recognition technique developed in this dissertation is an ensemble classifier based on a hybrid similarity measure approach. The merit in combining different measures is presenting a more comprehensive view of data by bringing together different perspectives. By applying this approach, the ensemble classifier can achieve an accurate recognition of activities across different users. The measures are also simple to calculate and update, thus enabling recognition of activities in streaming environment.

- *Personalise the learning model with evolving data streams:* We presented in this thesis an innovative technique that uses incremental learning for model personalisation while a stream evolves, STAR. The technique benefits from the learning model flexibility by enabling incremental learning

for model personalisation. It handles the streaming nature of data and tunes the model automatically to reflect the changes in data streams arising from personalisation.

- *Detect novel activities and adapt model accordingly:* In addition to tuning the model for personalisation, we propose, develop, and evaluate COSTAR that extends the dynamics beyond personalisation. COSTAR enables continuous learning for recognising novel activities and forgetting abandoned ones. The technique adapts the model continuously with the evolving stream, to accommodate for the detected changes. Model adaptation enables accurate representation of the most recent activities and thus enhances recognition performance.

- *Incorporating active learning:* The three techniques developed in this dissertation address the known challenge of the scarcity of labelled data. Thus, they incorporate an active learning approach that automatically selects the most informative data to be labelled. Therefore, active learning can effectively enhance the recognition performance with low labelling cost.

The results of this thesis have been validated and published in six peer-reviewed conference papers [AGSK12b,AGSK12a,AG09,AG11,BZA12,JGN+14], one journal article [AGSK15], and another journal article that is subject to a final revision [AG15]. After highlighting the key contributions of our research, the next section outlines the research directions and future work.

6.3 Future Research Directions

The research work implemented in this dissertation sheds light on different areas of research directions for extensions. We highlight in this section some of these key directions as follows:

- Enabling *personalisation* in activity recognition benefits the research area for accurate and efficient recognition. With an efficient personalisation, activity recognition becomes widely available and accurate across users, as there is no need to train the recognition system on each user's way of performing activities. General models with our approach are automatically customised for a user's personalised activities at real time and on limited resource devices. The capability of personalisation supports an

improved and accurate recognition across many applications. These applications include personalised advertisements, lifestyle monitoring, and elderly people monitoring.

- The capturing of novel activities and dynamic *adaptation* with evolving activities enables the development of reliable systems for activity recognition that can learn new knowledge and forget outdated knowledge. With the adaptation technique, there is no need to define a set of activities to recognise. The automatically adaptable technique is a self-learner that monitors the stream and understands the evolving data. This approach is a significant contribution in activity recognition as it creates less dependency on the historical annotated data. The future direction of this feature is to enable an independent technique that requires almost no prior knowledge to recognise activities. The feature can be extended for detecting and recognising activities on the fly with evolving streams. The dynamic adaptation feature can incrementally create the model with detected activities once they arrive and without prior training.

- Future work of this dissertation includes also building a *holistic approach* for activity recognition that integrates the personalisation of existing activities with the adaptation for novel detected and forgotten activities. The holistic approach extends our techniques by creating a single framework for detecting both major and minor changes in activities while a stream evolves.

- A big challenge in activity recognition is to collect a sufficient amount of labeled data to train the learning model. The annotating process is expensive, time-consuming and erroneous. We introduce in this dissertation an incremental and active learning approach that addresses this problem. The initial model in the developed approach is built with a small amount of labeled data. The model is continuously accumulated through incremental learning. Active learning asks only for labels of informative samples. This approach opens the door for a wider range of applications in activity recognition as it allows learning from unlabeled data, which is pervasively available. The research in this direction is still new and promising as it aims at less supervision in the recognition process.

- The developed adaptive techniques are capable of recognising activities accurately. The accurate recognition of activities can be combined with a *context aware* framework for recognising higher level and more complex

activities. A future activity recognition tiered approach has to leverage all available information for context aware recognition. Building a context aware technique for activity recognition enables a wide range of applications that concerns not only the activity of the user, but also the context of surrounding environment, to be able to provide the user with accurate and opportune information. We have presented an early work in this direction in [BZA12].

- Benefiting from the dynamic and adaptive recognition, the recognition technique could be applied to a wide range of users, to provide crowdsourcing information based on their performed activities. The computationally efficient and real time recognition approach developed in this thesis allows implementation of the recognition technique on a user's individual mobile devices. Aggregating on the recognised activities from users' devices on a higher level platform, such as the cloud, enables a wider perspective of understanding the performed activities based on different criteria, such as location based activity recognition on the cloud. Our work in [JGN+14] presents a step in this direction for analysing activities in the fog.

In conclusion, this thesis takes a step forward and opens up new opportunities in realising the potential of activities enhanced with compositional adaptation for pervasive and ubiquitous computing.

Appendix A

Table A.1 describes the distribution of data in PAMAP2 across users and activities.

Table A.1: PAMAP2 Description of Performed Activities for each User (in seconds)

User	**S 101**	**S 102**	**S 103**	**S 104**	**S 105**	**S 106**	**S 107**	**S 108**	**S 109**
lying	271.86	234.29	220.43	230.46	236.98	233.39	256.1	241.64	0
sitting	234.79	223.44	287.6	254.91	268.63	230.4	122.81	229.22	0
standing	217.16	255.75	205.32	247.05	221.31	243.55	257.5	251.59	0
walking	222.52	325.32	290.35	319.31	320.32	257.2	337.19	315.32	0
running	212.64	92.37	0	0	246.45	228.24	36.91	165.31	0
cycling	235.74	251.07	0	226.98	245.76	204.85	226.79	254.74	0
nordic walking	202.64	297.38	0	275.32	262.7	266.85	287.24	288.87	0
ascending stairs	158.88	173.4	103.87	166.92	142.79	132.89	176.44	116.81	0
descending stairs	148.97	152.11	152.72	142.83	127.25	112.7	116.16	96.53	0
vacuum cleaning	229.4	206.82	203.24	200.36	244.44	210.77	215.51	242.91	0
ironing	235.72	288.79	279.74	249.94	330.33	377.43	294.98	329.89	0
rope jumping	129.11	132.61	0	0	77.32	2.55	0	88.05	63.9

Table A.2 describes the total time in seconds for each activity.

Table A.2: Total Duration of Activities in PAMAP2

Activity	**Duration (in seconds)**
lying	1925.15
sitting	1851.8
standing	1899.23
walking	2387.53
running	981.92
cycling	1645.93
nordic walking	1881
ascending stairs	1172
descending stairs	1049.27
vacuum cleaning	1753.45
ironing	2386.82
rope jumping	493.54

References

[AAR96] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *Knowledge Discovery and Data Mining (KDD)*, pages 164–169, 1996.

[AB13] Iris Adä and Michael R Berthold. Eve: a framework for event detection. *Evolving systems*, 4(1):61–70, 2013.

[ABD00] Tim Armstrong, Adrian Ernest Bauman, and Joanne Davies. Physical activity patterns of australian adults: results of the 1999 national physical activity survey. Technical report, Australian Institute of Health and Welfare, Canberra, 2000. `http://aihw.gov.au/WorkArea/DownloadAsset.aspx?id=6442454841`.

[ABKS99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *SIGMOD Record*, 28(2):49–60, 1999.

[act] Actitracker application. `https://actitracker.com/`. Accessed: 2014-12-15.

[AF07] Fabrizio Angiulli and Fabio Fassetti. Detecting distance-based outliers in streams of data. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 811–820, New York, NY, USA, 2007. ACM.

[AG09] Zahraa Said Abdallah and Mohamed Medhat Gaber. DDG-Clustering : A novel technique for highly accurate results. In *Proceedings of the IADIS European Conference on Data Mining*, pages 163–167, 2009.

[AG11] Zahraa Said Abdallah and Mohamed Medhat Gaber. KB-CB-N classification: Towards unsupervised approach for supervised

learning. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence*, pages 283–290, 2011.

[AG15] Zahraa Said Abdallah and Mohamed Medhat a Gaber. A novel ensemble similarity-based approach to machine learning. *International Journal of Information Technology and Decision Making*, Accepted, subject to final revision, 2015.

[Agg13] Charu C. Aggarwal. *Outlier Analysis*. Springer New York, 2013.

[AGPV13] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 13–24, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[AGSK12a] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. CBARS: cluster based classification for activity recognition systems. In *Advanced Machine Learning Technologies and Applications*, volume 322 of *Communications in Computer and Information Science*, pages 82–91. Springer Berlin Heidelberg, 2012.

[AGSK12b] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. StreamAR: incremental and active learning with evolving sensory data for activity recognition. In *IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, volume 1, pages 1163–1170. IEEE, 2012.

[AGSK15] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, 150, Part A:304 – 317, 2015.

[AHWY03] Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment, 2003.

[AHWY04] Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu. On demand classification of data streams. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 503–508. ACM, 2004.

[AKBS12] Ira Assent, Philipp Kranen, Corinna Baldauf, and Thomas Seidl. Anyout: Anytime outlier detection on streaming data. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications - Volume Part I*, DASFAA'12, pages 228–242, Berlin, Heidelberg, 2012. Springer-Verlag.

[AKMK+12] Tahseen Al-Khateeb, Mohammad M Masud, Latifur Khan, Charu C Aggarwal, Jiawei Han, and Bhavani M Thuraisingham. Stream classification with recurring and novel class detection using class-based ensemble. In *ICDM*, pages 31–40, 2012.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[BDM03] Brian Babcock, Mayur Datar, and Rajeev Motwani. Load shedding techniques for data stream systems. In *Workshop on Management and Processing of Data Streams*, 2003.

[BDMO03] Brain Babcock, Mayur Datar, Rajeev Motwani, and Liadan O'Callaghan. Maintaining variance and k-medians over data stream windows. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 234–243. ACM, 2003.

[BG06] Albert Bifet and Ricard Gavaldà. Kalman filters and adaptive windows for learning in data streams. In *Proceedings of the 9th International Conference on Discovery Science*, DS'06, pages 29–40, Berlin, Heidelberg, 2006. Springer-Verlag.

[BG07] Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *Proceedings of the SIAM International Conference on Data Mining*, pages 443–448, 2007.

[BGC09] Tomas Brezmes, Juan-Luis Gorricho, and Josep Cotrina. Activity recognition from accelerometer data on a mobile phone.

In *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, volume 5518 of *Lecture Notes in Computer Science*, pages 796–799. Springer Berlin Heidelberg, 2009.

[BI04] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2004.

[BJPT03] Robert Bodor, Bennett Jackson, Nikolaos Papanikolopoulos, and Human Tracking. Vision-based human tracking and activity recognition. In *Proceedings of the 11th Mediterranean Conference on Control and Automation*, pages 18–20. Kostrzewa Joseph, 2003.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM.

[BNpBS12] Abbas Bahrololoum, Hossein Nezamabadi-pour, Hamid Bahrololoum, and Masoud Saeed. A prototype classifier based on gravitational search algorithm. *Applied Soft Computing*, 12(2):819–825, 2012.

[Bou11] Abdelhamid Bouchachia. Incremental learning with multi-level adaptation. *Neurocomputing*, 74(11):1785–1799, 2011.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BZA12] Andrey Boytsov, Arkady Zaslavsky, and Zahraa Abdallah. Where have you been? using location clustering and context awareness to understand places of interest. In *Internet of Things, Smart Spaces, and Next Generation Networking*, volume 7469 of *Lecture Notes in Computer Science*, pages 51–62. Springer Berlin Heidelberg, 2012.

[CCH+08] Tanzeem Choudhury, Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Anthony Lamarca, Louis Legrand, Ali Rahimi, Adam Rea, Gaetano Bordello, Bruce Hemingway, Predrag Klasnja, Karl Koscher, James.A. Landay, Jonathan Lester, Danny Wyatt, and

Dirk Haehnel. The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, 7(2):32–41, 2008.

[CEQZ06] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the SIAM International Conference on Data Mining*, pages 328–339, 2006.

[CFK13] Diane Cook, KyleD. Feuz, and NarayananC. Krishnan. Transfer learning for activity recognition: a survey. *Knowledge and Information Systems*, 36(3):537–556, 2013.

[CGL83] Tony F. Chan, Gene H. Golub, and Randall J. Leveque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983.

[CK11] Liming Chen and Ismail Khalil. Activity recognition: Approaches, practices and trends. In *Activity Recognition in Pervasive Intelligent Environments*, volume 4 of *Atlantis Ambient and Pervasive Intelligence*, pages 1–31. Atlantis Press, 2011.

[CM14] Ling Chen and Qingling Mei. Mining frequent items in data stream using time fading model. *Information Sciences*, 257:54–69, 2014.

[CMT+08] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in the wild: A field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1797–1806, New York, NY, USA, 2008. ACM.

[CNCC08] Yongwon Cho, Yunyoung Nam, Yoo-Joo Choi, and We-Duke Cho. Smartbuckle: Human activity recognition using a 3-axis accelerometer and a wearable camera. In *Proceedings of the 2Nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, HealthNet '08, pages 7:1–7:3, New York, NY, USA, 2008. ACM.

[CP01] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, pages 409–415, 2001.

[CSC+13] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José Del R. Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[DH00] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80. ACM, 2000.

[DLL13] Thang M. Do, Seng W. Loke, and Fei Liu. Healthylife: An activity recognition system with smartphone using logic-based stream reasoning. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 120 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 188–199. Springer Berlin Heidelberg, 2013.

[DND77] Arthur P. Dempster, M. Laird Nan, and B. Rubin Donald. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DR09] Anton Dries and Ulrich Rückert. Adaptive concept drift detection. *Statistical Analysis and Data Mining*, 2(56):311–327, 2009.

[EFWD88] George Efstathiou, Carlos S Frenk, Simon DM White, and Marc Davis. Gravitational clustering from scale-free initial conditions. *Monthly Notices of the Royal Astronomical Society*, 235(3):715–748, 1988.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Conference on Knowledge Discovery and Data Mining (KDD)*, volume 96, pages 226–231, 1996.

[EPMK08] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[FF00] Friedrich Foerster and Jochen Fahrenberg. Motion pattern and posture: Correctly assessed by calibrated accelerometers. *Behavior Research Methods, Instruments, & Computers*, 32(3):450–457, 2000.

[FGC13] Elaine R. Faria, João Gama, and André C. P. L. F. Carvalho. Novelty detection algorithm for data streams multi-class problems. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 795–800, New York, NY, USA, 2013. ACM.

[FHWP04] Wei Fan, Yi-an Huang, Haixun Wang, and S Yu Philip. Active mining of data streams. In *SDM*. SIAM, 2004.

[Fis36] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[fit] Fitbit products. 2012. `http://www.fitbit.com`. Accessed: 2014-12-15.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[FTARR04] Francisco Ferrer-Troyano, Jesús S. Aguilar-Ruiz, and José C. Riquelme. Discovering decision rules from numerical data streams. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, SAC '04, pages 649–653, New York, NY, USA, 2004. ACM.

[Gam12] João Gama. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence*, 1(1):45–55, 2012.

[Gav99] Dariu M. Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 73(1):82–98, 1999.

[GFH09] Norbert Györbíró, Ákos Fábián, and Gergely Hományi. An activity recognition system for mobile phones. *Mobile Networks and Applications.*, 14(1):82–91, 2009.

[GFJH08] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 283–291, New York, NY, USA, 2008. ACM.

[GGI+02] Anna C Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, S Muthukrishnan, and Martin J Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 389–398. ACM, 2002.

[GGK+14] Mohamed Medhat Gaber, João Gama, Shonali Krishnaswamy, João Bártolo Gomes, and Frederic Stahl. Data stream mining in ubiquitous environments: state-of-the-art and current directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):116–138, 2014.

[GHP+03] Chris Giannella, Jiawei Han, Jian Pei, Xifeng Yan, and Philip S Yu. Mining frequent patterns in data streams at multiple time granularities. *Next generation data mining*, 212:191–212, 2003.

[GJAS06] Raghu K. Ganti, Praveen Jayachandran, Tarek F. Abdelzaher, and John A. Stankovic. Satire: A software architecture for smart attire. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, MobiSys '06, pages 110–123, New York, NY, USA, 2006. ACM.

[GKG+12a] João Bártolo Gomes, Shonali Krishnaswamy, Mohamed M. Gaber, Pedro A. C. Sousa, and Ernestina Menasalvas. Mobile activity recognition using ubiquitous data stream mining. In *Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery*, DaWaK'12, pages 130–141. Springer-Verlag, Berlin, Heidelberg, 2012.

[GKG+12b] Joao Bartolo Gomes, Shonali Krishnaswamy, Mohamed Medhat Gaber, Pedro A. C. Sousa, and Ernestina Menasalvas. Mars: A personalised mobile activity recognition system. In *Proceedings*

of the 2012 IEEE 13th International Conference on Mobile Data Management (Mdm 2012), MDM '12, pages 316–319, Washington, DC, USA, 2012. IEEE Computer Society.

[GKZ04] Mohamed Medhat Gaber, Shonali Krishnaswamy, and Arkady Zaslavsky. Cost-efficient mining techniques for data streams. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation-Volume 32*, pages 109–114. Australian Computer Society, Inc., 2004.

[GKZ05] Mohamed Medhat Gaber, Shonali Krishnaswamy, and Arkady Zaslavsky. On-board mining of data streams in sensor networks. In *Advanced methods for knowledge discovery from complex data*, pages 307–335. Springer, 2005.

[GMCR04] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence–SBIA 2004*, pages 286–295. Springer, 2004.

[GMM+03] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *Knowledge and Data Engineering, IEEE Transactions on*, 15(3):515–528, 2003.

[GMMO00] Sudipto Guha, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams. In *Foundations of computer science, proceedings. 41st annual symposium on*, pages 359–366. IEEE, 2000.

[GQ12] Thomas Guyet and René Quiniou. Incremental mining of frequent sequences from a window sliding over a stream of itemsets. In *Journées Intelligence Artificielle Fondamentale*, pages 153–162, France, 2012.

[GR09] João Gama and Pedro Pereira Rodrigues. An overview on mining data streams. In *Foundations of Computational, IntelligenceVolume 6*, pages 29–45. Springer, 2009.

[GRSdC10] João Gama, Pedro Pereira Rodrigues, Eduardo J Spinosa, and André Carlos Ponce Leon Ferreira de Carvalho. *Knowledge discovery from data streams*. Chapman & Hall/CRC Boca Raton, 2010.

[GS99] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 33–42, New York, NY, USA, 1999. ACM.

[GvB+14] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

[GWT+09] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, PERCOM '09, pages 1–9, Washington, DC, USA, 2009. IEEE Computer Society.

[GY06] Mohamed Medhat Gaber and Philip S Yu. Detection and classification of changes in evolving data streams. *International Journal of Information Technology & Decision Making*, 5(04):659–670, 2006.

[GYL+07] Donghai Guan, Weiwei Yuan, Young-Koo Lee, A. Gavrilov, and Sungyoung Lee. Activity recognition based on semi-supervised learning. In *Embedded and Real-Time Computing Systems and Applications, RTCSA. 13th IEEE International Conference on*, pages 469–475, 2007.

[GZK05] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.

[GZK07] MohamedMedhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. A survey of classification methods in data streams. In *Data Streams*, volume 31 of *Advances in Database Systems*, pages 39–59. Springer US, 2007.

[Hal99] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[Han05] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[HANp12] Abdolreza Hatamlou, Salwani Abdullah, and Hossein Nezamabadi-pour. A combined approach for clustering based on k-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, 6:47 – 52, 2012.

[Har75] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[Haw80] Douglas M Hawkins. *Identification of outliers*. Chapman and Hall, London, 1980.

[HD07] Shucheng Huang and Yisheng Dong. An active learning system for mining time-changing data streams. *Intelligent Data Analysis*, 11(4):401–419, 2007.

[HFS08] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 10–19, New York, NY, USA, 2008. ACM.

[HFW08] Kathryn Hempstalk, Eibe Frank, and Ian H. Witten. One-class classification by combining density and class probability estimation. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, ECML PKDD '08, pages 505–519. Springer-Verlag, Berlin, Heidelberg, 2008.

[HH10] Morteza Zi Hayat and Mahmoud Reza Hashemi. A dct based approach for detecting novelty and concept drift in data streams. In *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, pages 373–378, 2010.

[HKAK10] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Mobile health monitoring system based on activity recognition using accelerometer. *Simulation Modelling Practice and Theory*, 18(4):446–455, 2010.

[HKM+01] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmäki, and Hannu Toivonen. Time series segmentation for context recognition in mobile devices. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 203–210, Washington, DC, USA, 2001. IEEE Computer Society.

[hLKK+09] Mi hee Lee, Jungchae Kim, Kwangsoo Kim, Inho Lee, Sun Ha Jee, and Sun Kook Yoo. Physical activity recognition using a single tri-axis accelerometer. In *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I, WCECS '09, October 20 - 22, 2009, San Francisco, USA*, Lecture Notes in Engineering and Computer Science, pages 14–17. International Association of Engineers, Newswood Limited, 2009.

[HS06] Tâm Huynh and Bernt Schiele. Towards less supervision in activity recognition from wearable sensors. In *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 3–10. IEEE, 2006.

[HSD01] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 97–106, New York, NY, USA, 2001. ACM.

[HSG+07] Jiayuan Huang, Alexander J Smola, Arthur Gretton, Karsten M Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2007.

[HSU12] Hirotaka Hachiya, Masashi Sugiyama, and Naonori Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101, 2012.

[HY11] Derek Hao Hu and Qiang Yang. Transfer learning for activity recognition via sensor mapping. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1962–1967. AAAI Press, 2011.

[Int04] Stephen S Intille. Ubiquitous computing technology for just-in-time motivation of behavior change. *Stud Health Technol Inform*, 107(Pt 2):1434–1437, 2004.

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[JGN+14] Prem Prakash Jayaraman, João Bártolo Gomes, Hai Long Nguyen, Zahraa Said Abdallah, Shonali Krishnaswamy, and

Arkady Zaslavsky. Cardap: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog. In *Advances in Databases and Information Systems*, pages 192–206. Springer, 2014.

[JL95] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[KBDG04] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 180–191. VLDB Endowment, 2004.

[KC14] Narayanan C. Krishnan and Diane J. Cook. Activity recognition on streaming sensor data. *Pervasive Mobile Computing*, 10:138–154, 2014.

[KH08] Ashish Kapoor and Eric Horvitz. Experience sampling for building predictive user models: A comparative study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 657–666, New York, NY, USA, 2008. ACM.

[KHF+11] Marc Kurz, Gerold Hölzl, Alois Ferscha, Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system. In *The Third International Conference on Adaptive and Self-Adaptive Systems and Applications*, pages 73–78, 2011.

[KJ00] Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. In *ICML*, pages 487–494, 2000.

[KLLK10] Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y. Lee, and Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *Information Technology in Biomedicine, IEEE Transactions on*, 14(5):1166–1172, 2010.

[KNM+06a] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for

ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, 2006.

[KNM+06b] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, 2006.

[Knu97] Donald E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3 edition, 1997.

[KR98] Ralf Klinkenberg and Ingrid Renz. Adaptive information filtering: Learning drifting concepts. In *Workshop Learning for Text Categorization AAAI98/ICML-98*, pages 33–40. Press, 1998.

[KRT+05] N. Krahnstoever, J. Rittscher, P. Tu, K. Chean, and T. Tomlinson. Activity recognition using visual tracking and rfid. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1 - Volume 01*, WACV-MOTION '05, pages 494–500, Washington, DC, USA, 2005. IEEE Computer Society.

[KS13] Vrushali Y Kulkarni and Pradeep K Sinha. Random forest classifiers: A survey and future research directions. *International Journal of Advanced Computing*, 36(1):1144–1153, 2013.

[Kun04] Ludmila I Kuncheva. Classifier ensembles for changing environments. In *Multiple classifier systems*, pages 1–15. Springer, 2004.

[KWM11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[KZX+11] Matthew Keally, Gang Zhou, Guoliang Xing, Jianxin Wu, and Andrew Pyles. Pbn: Towards practical activity recognition using smartphone-based body sensor networks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys '11, pages 246–259, New York, NY, USA, 2011. ACM.

[Las02] Mark Last. Online classification of nonstationary data streams. *Intelligent Data Analysis*, 6(2):129–147, 2002.

[LB69] M. Powell Lawton and Elaine M. Brody. Assessment of older people: Self-maintaining and instrumental activities of daily living. *The Gerontologist*, 9(3):179–186, 1969.

[LC14] Young-Seol Lee and Sung-Bae Cho. Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. *Neurocomputing*, 126:106–115, 2014.

[LCB06] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Proceedings of the 4th International Conference on Pervasive Computing*, PERVASIVE'06, pages 1–16. Springer-Verlag, Berlin, Heidelberg, 2006.

[LCK+05] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 766–772, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[LD11] Fei Li and Schahram Dustdar. Incorporating unsupervised learning in activity recognition. In *Activity Context Representation*, 2011.

[LFK05] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition. In *Advances in Neural Information Processing Systems (NIPS*, pages 787–794. MIT Press, 2005.

[LHP+07] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, UbiComp '07, pages 483–500, Berlin, Heidelberg, 2007. Springer-Verlag.

[LK09] Mitja Luštrek and Boštjan Kaluža. Fall detection and activity recognition with machine learning. *Informatica (Slovenia)*, 33(2):197–204, 2009.

[LKLC03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD*

Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM.

[LL13] Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys Tutorials, IEEE*, 15(3):1192–1209, 2013.

[LLPK09] Min-Seok Lee, Jong-Gwan Lim, Ki-Ru Park, and Dong-Soo Kwon. Unsupervised clustering for abnormality detection based on the tri-axial accelerometer. In *ICCAS-SICE, 2009*, pages 134–137, 2009.

[LM02] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24–32, 2002.

[LML+10] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.

[LMND13] Patrick Lindstrom, Brian Mac Namee, and Sarah Jane Delany. Drift detection using uncertainty distribution divergence. *Evolving Systems*, 4(1):13–25, 2013.

[LP04] Neil D. Lawrence and John C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 65–, New York, NY, USA, 2004. ACM.

[LPW12] Jeffrey W. Lockhart, Tony Pulickal, and Gary M. Weiss. Applications of mobile activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 1054–1058, New York, NY, USA, 2012. ACM.

[LRE10] Brent Longstaff, Sasank Reddy, and Deborah Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 4th International Conference on-NO PERMISSIONS*, pages 1–7. IEEE, 2010.

[LW14] Jeffrey W. Lockhart and Gary M. Weiss. The benefits of personalized smartphone-based activity recognition models. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 614–622, 2014.

[LWFF07] Li-Jia Li, Gang Wang, and Li Fei-Fei. OPTIMOL: automatic online picture collection via incremental model learning. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8, 2007.

[LWH12] Peipei Li, Xindong Wu, and Xuegang Hu. Mining recurring concept drifts with limited labeled streaming data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):29, 2012.

[LYA09] Xi Long, Bin Yin, and R.M. Aarts. Single-accelerometer-based daily physical activity classification. In *Engineering in Medicine and Biology Society, EMBC. Annual International Conference of the IEEE*, pages 6107–6110, 2009.

[MAKK+11] Mohammad M. Masud, Tahseen M. Al-Khateeb, Latifur Khan, Charu Aggarwal, Jing Gao, Jiawei Han, and Bhavani Thuraisingham. Detecting recurring and novel classes in concept-drifting data streams. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 1176–1181, Washington, DC, USA, 2011. IEEE Computer Society.

[MCK+13] Mohammad M Masud, Qing Chen, Latifur Khan, Charu C Aggarwal, Jing Gao, Jiawei Han, Ashok Srivastava, and Nikunj C Oza. Classification and adaptive novel class detection of feature-evolving data streams. *Knowledge and Data Engineering, IEEE Transactions on*, 25(7):1484–1497, 2013.

[MCP98] Florent Masseglia, Fabienne Cathala, and Pascal Poncelet. The psp approach for mining sequential patterns. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '98, pages 176–184, London, UK, UK, 1998. Springer-Verlag.

[MGK+11] Mohammad M. Masud, Jing Gao, L. Khan, Jiawei Han, and Bhavani Thuraisingham. Classification and novel class detection

in concept-drifting data streams under time constraints. *Knowledge and Data Engineering, IEEE Transactions on*, 23(6):859–874, 2011.

[MLF+08] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 337–350, New York, NY, USA, 2008. ACM.

[MM02] Gurmeet Singh Manku and Rajeev Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 346–357. VLDB Endowment, 2002.

[MM06] Alice Marascu and Florent Masseglia. Mining sequential patterns from data streams: a centroid approach. *Journal of Intelligent Information Systems*, 27(3):291–307, 2006.

[MM08] Lilyana Mihalkova and Raymond J Mooney. Transfer learning by mapping with minimal target data. In *Proceedings of the AAAI-08 workshop on transfer learning for complex tasks*, 2008.

[MMEJ08] Iqbal Mohomed, Archan Misra, Maria Ebling, and William Jerome. Harmoni: Context-aware filtering of sensor data for continuous remote health monitoring. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '08, pages 248–251, Washington, DC, USA, 2008. IEEE Computer Society.

[MMK00] Ion Muslea, Steven Minton, and Craig A. Knoblock. Selective sampling with redundant views. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 621–626. AAAI Press, 2000.

[MRS+09] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Péter Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th International Conference on Mobile Systems,*

Applications, and Services, MobiSys '09, pages 55–68, New York, NY, USA, 2009. ACM.

[MS10] Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[MSS13] Muhammad Mubashir, Ling Shao, and Luke Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.

[MSSD06] Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, BSN '06, pages 113–116, Washington, DC, USA, 2006. IEEE Computer Society.

[MWG+12] Mohammad M Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W Hamlen, and Nikunj C Oza. Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowledge and information systems*, 33(1):213–244, 2012.

[NCM04] Hammadi Nait-Charif and Stephen J. McKenna. Activity summarisation and fall detection in a supportive home environment. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04*, ICPR '04, pages 323–326, Washington, DC, USA, 2004. IEEE Computer Society.

[NKR10] Vit Niennattrakul, Eamonn Keogh, and Chotirat Ann Ratanamahatana. Data editing techniques to allow the application of distance-based outlier detection to streams. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 947–952, Washington, DC, USA, 2010. IEEE Computer Society.

[NLHW04] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang. Human activity detection and recognition for video surveillance. In *Multimedia and Expo. ICME '04. IEEE International Conference on*, volume 1, pages 719–722, 2004.

[OCY01] Yen-Jen Oyang, Chien-Yu Chen, and Tsui-Wei Yang. A study on the hierarchical data clustering algorithm based on gravity theory. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '01, pages 350–361. Springer-Verlag, London, UK, UK, 2001.

[OMG+02] Liadan O'callaghan, Nina Mishra, Sudipto Guha, Adam Meyerson, and Rajeev Motwani. Streaming-data algorithms for high-quality clustering. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 0685–0685. IEEE Computer Society, 2002.

[PCE10] Juha Pärkkä, Luc Cluitmans, and Miikka Ermes. Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree. *Information Technology in Biomedicine, IEEE Transactions on*, 14(5):1211–1215, 2010.

[Pen00] Alex Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, 2000.

[PFKP05] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, ISWC '05, pages 44–51, Washington, DC, USA, 2005. IEEE Computer Society.

[PFP+04] Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[PGK+09] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity identification using body-mounted sensorsa review of classification techniques. *Physiological measurement*, 30(4):1–33, 2009.

[PHW07] Jian Pei, Jiawei Han, and Wei Wang. Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems*, 28(2):133–160, 2007.

[PLL07] Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Latecki. Incremental local outlier detection for data streams. In *Computational Intelligence and Data Mining, CIDM. IEEE Symposium on*, pages 504–515. IEEE, 2007.

[PPGD14] Tom Peterek, Marek Penhaker, Petr Gajdo, and Pavel Dohnlek. Comparison of classification algorithms for physical activity recognition. In *Innovations in Bio-inspired Computing and Applications*, volume 237 of *Advances in Intelligent Systems and Computing*, pages 123–131. Springer International Publishing, 2014.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RB11] Daniele Riboni and Claudio Bettini. Cosar: Hybrid reasoning for context-aware activity recognition. *Personal Ubiquitous Computing*, 15(3):271–289, 2011.

[RBL+07] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 759–766, New York, NY, USA, 2007. ACM.

[RC09] Parisa Rashidi and Diane J Cook. Transferring learned activities in smart environments. In *Intelligent Environments*, pages 185–192, 2009.

[RC10] Parisa Rashidi and Diane J. Cook. Mining sensor streams for discovering human activity patterns over time. In *IEEE 10th International Conference on Data Mining (ICDM)*, pages 431–440, 2010.

[RCHSE11] Parisa Rashidi, Diane J Cook, Lawrence B Holder, and Maureen Schmitter-Edgecombe. Discovering activities to recognize and track in a smart environment. *Knowledge and Data Engineering, IEEE Transactions on*, 23(4):527–539, 2011.

[RCR+10] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Forster, Gerhard Troster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. del R Millan. Collecting complex activity datasets in highly rich networked sensor environments. In *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, pages 233–240, 2010.

[RDML05] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, pages 1541–1546. AAAI Press, 2005.

[RFC+09] Daniel Roggen, Kilian Förster, Alberto Calatroni, Thomas Holleczek, Y. Fang, Gerhard Tröster, Paul Lukowicz, Gerald Pirkl, David Bannach, Kai Kunze, Alois Ferscha, Clemens Holzmann, Andreas Riener, R. Chavarriaga, and J. del R. Millán. Opportunity: Towards opportunistic activity and context recognition systems. In *World of Wireless, Mobile and Multimedia Networks Workshops. WoWMoM. IEEE International Symposium on a*, pages 1–6, 2009.

[RFCT13] Daniel Roggen, Kilian Förster, Alberto Calatroni, and Gerhard Tröster. The adarc pattern analysis architecture for adaptive human activity recognition systems. *Journal of Ambient Intelligence and Humanized Computing*, 4(2):169–186, 2013.

[RGP08] Pedro Pereira Rodrigues, João Gama, and Joao Pedroso. Hierarchical clustering of time-series data streams. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):615–627, 2008.

[RH+97] Dan Y. Rubinstein, Trevor Hastie, et al. Discriminative vs informative learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 49–53. AAAI Press, 1997.

[RNp14] Mohadeseh Rezaei and Hossein Nezamabadi-pour. A prototype optimization method for nearest neighbor classification by gravitational search algorithm. In *Intelligent Systems (ICIS), 2014 Iranian Conference on*, pages 1–4, 2014.

[RNpS09] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. Gsa: A gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[RPT08] Chedi Raıssi, Pascal Poncelet, and Maguelonne Teisseire. Need for speed: Mining sequential patterns in data streams. In *BDA'05: Bases de données Avancées 2005*, page 12, Palais du Grand Large, Saint-Malo, 2008.

[RS12a] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, pages 40:1–40:8, New York, NY, USA, 2012. ACM.

[RS12b] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 108–109, 2012.

[RS13] Attila Reiss and Didier Stricker. Personalized mobile physical activity recognition. In *Proceedings of the 2013 International Symposium on Wearable Computers*, ISWC '13, pages 25–28, New York, NY, USA, 2013. ACM.

[SAT+99] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 89–101, London, UK, 1999. Springer Verlag.

[SCG07] Eduardo J. Spinosa, André C. P. L. F. Carvalho, and João Gama. Olindda: A cluster-based approach for detecting novelty and concept drift in data streams. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 448–452, New York, NY, USA, 2007. ACM.

[SG00] Chris Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, 2000.

[SGJ86] Jeffrey C Schlimmer and Richard H Granger Jr. Incremental learning from noisy data. *Machine learning*, 1(3):317–354, 1986.

[SK01] W Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM, 2001.

[SRBF06] Amar Subramanya, Alvin Raj, Jeff A. Bilmes, and Dieter Fox. Recognizing activities and spatial context using wearable sensors. In *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA*, 2006.

[SRM12] Patrick Santos, Fernando Reinaldo Ribeiro, and José Metrolho. Using pervasive computing technologies to deliver personal and public ads in public spaces. In *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, pages 1–6, 2012.

[STF08] Dairazalia Sánchez, Monica Tentori, and Jesús Favela. Activity recognition for the smart hospital. *IEEE Intelligent Systems*, 23(2):50–57, 2008.

[SVA06] M.S. Sricharan, V. Vaidehi, and P.P. Arun. An activity based mobility prediction strategy for next generation wireless network. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 5–10, 2006.

[SVLS08] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*, ISWC '08, pages 81–88, Washington, DC, USA, 2008. IEEE Computer Society.

[SZC13] Saguna Saguna, Arkady Zaslavsky, and Dipanjan Chakraborty. Complex activity recognition using context-driven activity theory and activity signatures. *ACM Transactions on Computer-Human Interaction*, 20(6):32:1–32:34, 2013.

[TCY03] Wei-Guang Teng, Ming-Syan Chen, and Philip S. Yu. A regression-based temporal pattern mining scheme for data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases*, volume 29 of *VLDB '03*, pages 93–104. VLDB Endowment, 2003.

[TÇZ+03] Nesime Tatbul, Uğur Çetintemel, Stan Zdonik, Mitch Cherniack, and Michael Stonebraker. Load shedding on data streams. In *Proceedings of the Workshop on Management and Processing of Data Streams (MPDS 03), San Diego, CA, USA*, 2003.

[TF08] Monica Tentori and Jesus Favela. Activity-aware computing for healthcare. *Pervasive Computing, IEEE*, 7(2):51–57, 2008.

[TIL04] Emmanuel Munguia Tapia, StephenS. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175. Springer Berlin Heidelberg, 2004.

[VHC13] Quang Viet Vo, Minh Thang Hoang, and Deokjai Choi. Personalization in mobile activity recognition system using-medoids clustering algorithm. *International Journal of Distributed Sensor Networks*, 2013.

[VJ01] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

[VKP10] Ashok Venkatesan, Narayanan C Krishnan, and Sethuraman Panchanathan. Cost-sensitive boosting for concept drift. In *International workshop on handling concept drift in adaptive information systems*, pages 41–47, 2010.

[WA05] Daniel H. Wilson and Chris Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous binary sensors. In *Proceedings of the Third International Conference on Pervasive Computing*, PERVASIVE'05, pages 62–79. Springer-Verlag, Berlin, Heidelberg, 2005.

[WCZZ08] Wanhong Wu, Jiannong Cao, Yuan Zheng, and Yong-Ping Zheng. Waiter: A wearable personal healthcare and emergency aid system. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '08, pages 680–685, Washington, DC, USA, 2008. IEEE Computer Society.

[Web99] Geoffrey I. Webb. Decision tree grafting from the all-tests-but-one partition. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, pages 702–707, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2. edition, 2005.

[WFYH03] Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 226–235, New York, NY, USA, 2003. ACM.

[WG96] Robert B Woodruff and Sarah Gardial. *Know your customer: new approaches to customer value and satisfaction*. Blackwell Business Cambridge, MA, 1996.

[WK96] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[WL12] Gary M Weiss and Jeffrey W Lockhart. The impact of personalization on smartphone-based activity recognition. In *AAAI Workshop on Activity Context Representation: Techniques and Languages*, 2012.

[WLP+14] Gary M Weiss, Jeffrey W Lockhart, Tony T Pulickal, Paul T McHugh, Isaac H Ronan, and Jessica L Timko. Actitracker: A smartphone-based activity recognition system for improving health and well-being. In *SIGKDD Exploration Newsletter, New York, NY, USA*, August 2014.

[WLT05] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, sOc-EUSAI '05, pages 99–104, New York, NY, USA, 2005. ACM.

[WLTS06] Jamie A. Ward, Paul Lukowicz, Gerhard Troster, and Thad E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1553–1567, 2006.

[WP11] Bin Wei and Christopher J Pal. Heterogeneous transfer learning with RBMs. In *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

[WPC05] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*, AAAI'05, pages 21–27. AAAI Press, 2005.

[WY05] Dwi H Widyantoro and John Yen. Relevant data expansion for learning concept drift from sparsely labeled data. *Knowledge and Data Engineering, IEEE Transactions on*, 17(3):401–412, 2005.

[XHW13] Yao Xie, Jiaji Huang, and R. Willett. Change point detection for high-dimensional time series with missing data. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):12–27, 2013.

[Yan09] Jun Yang. Toward physical activity diary: Motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics*, IMCE '09, pages 1–10, New York, NY, USA, 2009. ACM.

[Yar95] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[YC02] Dit-Yan Yeung and C. Chow. Parzen-window network intrusion detectors. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 385–388, 2002.

[YH14] Bingchuan Yuan and John Herbert. Smartphone-based activity recognition using hybrid classifier. In *Proceeding of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems*, 2014.

[YHY+11] Minghao Yin, Yanmei Hu, Fengqin Yang, Xiangtao Li, and Wenxiang Gu. A novel hybrid k-harmonic means and gravitational search algorithm approach for clustering. *Expert Systems Applications*, 38(8):9319–9324, 2011.

[YJ06] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Technical report, Michigan State Universiy, 2006. `https://www.cs.cmu.edu/~liuy/frame_survey_v2.pdf`.

[YLLB10] Jun Yang, Hong Lu, Zhigang Liu, and Péter Pál Boda. Physical activity recognition with mobile phones: challenges, methods, and applications. In *Multimedia Interaction and Intelligent User Interfaces*, pages 185–213. Springer, 2010.

[YZCJ02] Yiming Yang, Jian Zhang, Jaime Carbonell, and Chun Jin. Topic-conditioned novelty detection. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 688–693, New York, NY, USA, 2002. ACM.

[Zad04] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 114–122, New York, NY, USA, 2004. ACM.

[Zak01] Mohammed J Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

[ZBG+12] Indre Zliobaite, Albert Bifet, Mohamed Gaber, Bogdan Gabrys, João Gama, Leandro Minku, and Katarzyna Musial. Next challenges for adaptive learning systems. *ACM SIGKDD Explorations Newsletter*, 14(1):48–55, 2012.

[ZBPH14] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(1):27–39, 2014.

[ZCL+11] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. Cross-people mobile-phone based activity recognition. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2545–2550. AAAI Press, 2011.

[ZHM08] Daqing Zhang, Mossaab Hariz, and Mounir Mokhtari. Assisting elders with mild dementia staying at home. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '08, pages 692–697, Washington, DC, USA, 2008. IEEE Computer Society.

[ZHY09] Vincent Wenchen Zheng, Derek Hao Hu, and Qiang Yang. Cross-domain activity recognition. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 61–70. ACM, 2009.

[ZLS+08] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection. In *Proceedings of the 5th European Conference on Wireless Sensor Networks*, EWSN'08, pages 17–33, Berlin, Heidelberg, 2008. Springer-Verlag.

[ZPYP08] Vincent Wenchen Zheng, Sinno Jialin Pan, Qiang Yang, and Jeffrey Junfeng Pan. Transferring multi-device localization models using latent multi-task learning. In *AAAI*, volume 8, pages 1427–1432, 2008.

[ZS02] Yunyue Zhu and Dennis Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 358–369. VLDB Endowment, 2002.

[ZS03] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 336–345. ACM, 2003.