

Open Mobile Miner: A Toolkit for Building Situation-Aware Data Mining Applications

Pari Delir Haghighi^{a,1}, Shonali Krishnaswamy^a, Arkady Zaslavsky^b, Mohamed Medhat Gaber^c, Abhijat Sinha^a, and Brett Gillick^a

^a *Faculty of Information Technology, Monash University, Australia*

^b *CSIRO, Australia*

^c *School of Computing, University of Portsmouth, Hampshire, UK*

Corresponding author: Pari Delir Haghighi, Faculty of Information Technology, Caulfield Campus, Monash University, address: 900 Dandenong Road, Caulfield East, Victoria 3145, Australia, Tel.: +613 9903 2355; fax: +631 9903 1077, Email: Pari.DelirHaghighi@monash.edu.

¹ Corresponding author. Pari Delir Haghighi, Faculty of Information Technology, Caulfield Campus, Monash University, address: 900 Dandenong Road, Caulfield East, Victoria 3145, Australia, Tel.: +613 9903 2355; fax: +631 9903 1077, Email: Pari.DelirHaghighi@monash.edu.

Open Mobile Miner: A Toolkit for Building Situation-Aware Data Mining Applications

Abstract – In organizational computing and information systems, data mining techniques have been widely used for analyzing customer behaviour and discovering hidden patterns. Mobile Data Mining is the process of intelligently analysing continuous data streams on mobile devices. The use of mobile data mining for real-time business intelligence applications can be greatly advantageous. Past research has shown that resource-aware adaptation of data stream mining can significantly improve the continuity of data mining operations in mobile environments. The key underlying premise is that by varying the accuracy of the analysis process in accordance with changing available resource levels, the longevity and continuity of mobile data mining applications is ensured. In this paper we qualitatively extend the notion of resource-aware adaptation of mobile data mining to holistically enable situation-awareness feature for user applications. We then present a novel generic toolkit that enables building situation and resource-aware mobile data mining applications and describe along with underlying theoretical foundations of resource and situation criticality, awareness and adaptation which are entirely transparent and hidden from the user. The Open Mobile Miner (OMM) toolkit builds on our research for performing adaptive analysis of data streams on mobile/embedded devices. Finally, we describe a mobile health monitoring application as a case study and discuss the results of our conducted experimental evaluation which demonstrate the adaptation transparency and easy use of OMM for building mobile data mining applications such as stock market monitoring and real estate data analysis.

Keywords - Data stream mining, Mobile computing, Ubiquitous computing, Adaptation model, Context awareness, e-Commerce applications.

1. Introduction

Data mining techniques have been widely used in organisational computing and e-commerce to learn and discover hidden knowledge and interesting patterns from large amounts of data (Gentry et al. 2002). These techniques enable analyzing customer behaviour and predicting future trends or customer churn (Bose and Chen 2011). The popularity, ubiquity and ever increasing power of mobile devices in terms of storage and processing have led to new classes of data mining applications that enable performing real-time analysis of large amounts of data on mobile/embedded devices (Gaber 2009; Stahl et al. 2012). Examples of such application domains include healthcare, Intelligent Transportation Systems (ITS), intrusion detection, stock market monitoring and real estate data analysis. The importance and significance of data mining and processing on mobile devices can be explained as follows.

- Transmitting data to centralized servers to be analyzed could be very expensive in terms of energy consumption and communications cost. In wireless devices, communication consumes more energy than computation (Raghunathan et al. 2002). In many cases, wireless sensors are close to the mobile/embedded device and hence onboard processing of sensory data can significantly reduce the costs/overheads of data transmission.
- Mobile data mining can be used as a supporting/complementary technology that can significantly reduce the cost of data collection and transmission by performing real-time, continuous, and intelligent processing of data onboard mobile device, and sending essential information for detailed analysis.
- Currently there is an increasing reliance on the capacity and capability of the mobile phones to provide a wide range of computational support and services to the user. We are

expecting our mobile phones to provide us with the same functionality as stationary computers while we are on the move (Perich et al. 2004). This technological evolution presents an unprecedented opportunity for mobile applications including mobile data mining systems. However, it also emphasises the need for energy-efficient data analysis approaches onboard mobile devices.

Consider a scenario where a mobile business user is monitoring streaming stock market data and needs to be alerted when an important occurrence/change is detected, such as a drop in share price. A stock market data mining application can assist the mobile user with real-time analysis of stock market data and inform him/her of any changes on the mobile phone. MobiMine (Kargupta et al. 2002) and (Fu et al. 2008) are examples of mobile data mining systems for monitoring financial stock market. MineFleet (Kargupta et al. 2010) is another example of a mobile and distributed data mining application for monitoring vehicle data streams in real-time that analyzes high throughput data streams onboard the vehicle.

Current data mining applications operating on mobile devices such as a smart phone (Brezmes et al. 2009; Kargupta et al. 2010; Talia and Trunfio 2010; Hanny and Baatard 2011) recognize the implicit need for adaptation as a key feature of any effective mobile application. However, they have little consideration for resource availability. Analyzing large amounts of sensory originated data in real-time is a very challenging task. This challenge is further exacerbated when data is processed with resource-constrained devices such as mobile phones. Resource constraints include limited computational resources such as memory, processor speed, network bandwidth, battery power, and screen real-estate. Table 1 illustrates the comparison between smart phones and desktop computers with the focus on critical resources (i.e. RAM memory, CPU speed and battery lifetime). Different applications place different constraints and

requirements on resources, and also depending on the application priority the waiting for these resources can vary.

Table 1 Performance Comparison between mobile phones and desktop computers.

Resources	Smart phones (e.g. iPhone¹ and Samsung Galaxy²)	Desktop PC	Comment
RAM Memory	Up to 1 GB	Up to 16GB	These values vary according to each brand and model.
CPU Speed	Up to 1.2 GHz Dual Core	About 1.4 GHz to 2.90 GHz (e.g. Intel Core i7 Extreme Processor)	Different variations of Intel Core i5 or i7 have different clock speed and cache capacity ³ .
Battery Life	Up to 8 hours talk time, up to 400 hours stand-by time, and about 4 hours for tethering and mobile AP (Access Point)	N/A	For mobile phones the battery life is a critical resource compared to the desktop PCs that use unlimited power.

Previous studies on resource-aware adaptation (Gaber et al. 2005; 2006; Gaber 2009; Phung et al. 2007) show that dynamic adaptation to data rates and fine tuning of processing parameters can significantly enhance the longevity of continuous real-time processing of data

¹ <http://www.apple.com/au/iphone/specs.html>

² <http://www.samsung.com/au/smartphone/galaxy-s-2/specifications.html>

³ <http://www.intel.com/content/www/us/en/processor-comparison/compare-intel-processors.html>

streams in mobile environments. The Granularity-based Adaptation (GA) (Gaber et al. 2004) is a generic efficient adaptation approach that can be used with any data stream mining technique running on a resource-constrained device. This approach facilitates adaptation of data mining algorithms to varying data rates and available computational resources in mobile devices.

In addition to availability of resources, mobile data mining application's accuracy requirements vary according to the occurring situations. For example, a health monitoring application requires lower accuracy when the patient is healthy and the occurring situation is 'normal'. A situation-aware adaptation technique controls the data stream mining settings according to current situations and accuracy requirements to improve the continuity of the running application (Delir Haghighi et al. 2010). There can be other scenarios in which it is important to adjust mining algorithms considering both the current situation and resource availability. An example of such scenario is when a health monitoring application requires high accuracy because the patient's health situation is not normal but the battery level is low. In such cases, there is a need to apply a hybrid adaptation strategy that combines situation and resource-aware adaptation methods (Delir Haghighi et al. 2009; 2010).

We have developed several data stream mining algorithms for Clustering, Change Detection, Classification and Frequent Items Analysis (Gaber et al. 2004; 2005; 2006; Phung et al. 2007) that operate using the above-mentioned principles of adaptation (i.e. resource or/and situation-aware). We have also extended these principles to visualization techniques for data stream mining (Gillick et al. 2006; 2010) on mobile devices as well. There have been application specific systems for mobile data mining that have been built, and several algorithms have been developed to perform analysis on mobile devices. However, till date, an integrated toolkit for performing data stream mining on mobile devices which has a range of algorithms to facilitate

different types of applications has not been developed (Krishnaswamy et al. 2009). In this article, we present the pioneering mobile data mining toolkit: Open Mobile Miner (OMM).

The primary motivations for the development of this platform are as follows: i) to provide a platform for evaluation of new and existing mobile data stream mining techniques by the research community; ii) to encapsulate extensibility of the toolkit by easy integration of new and existing data stream mining algorithms into the toolkit that may or may not have adaptation mechanisms incorporated; iii) to interface with a range of input sources for data streams including Bluetooth-enabled sensors, previously recorded data, distributed data, and synthetic data; iv) to allow flexible, application specific visualizations to be developed; v) to enable easy deployment of mobile data mining applications on a range of mobile devices; and vi) to present case studies that show applicability of OMM to a wide range of information systems and e-commerce applications, and healthcare.

Thus, the above considerations form the requisite functionality that has driven the development of the OMM. The key unique contributions of this paper include:

- the pioneering OMM software platform and its adaptation model that controls mobile data mining algorithms by factoring in resource availability and/or occurring situations;
- formalization of the situation-aware and hybrid adaptation strategies using the notions of criticality;
- experimental evaluation which demonstrates the benefits and transparency of the situation-aware and hybrid adaptation methods;
- a case study which demonstrates the ease of developing and deploying information systems through incorporating mobile data mining applications.

The rest of the paper is organized as follows: Section 2 presents the theoretical overview of the adaptation process and situation inference for mobile data mining algorithms. Section 3 presents an overview of the architecture of the Open Mobile Miner (OMM) with a discussion on its components. Section 4 presents the implementation and operation of the Open Mobile Miner. Section 5 presents a mobile healthcare case study that uses OMM's underlying approaches for mobile data mining and applies situation- and resource-aware and hybrid adaptation methods. This is followed by the details of our experimental evaluation of this case study to validate the benefits of situation-aware adaptation. Finally, the paper is concluded in Section 6.

2. Adaptation and Situation Inference for Mobile Data Mining

This section provides an overview of the theoretical concepts underpinning the adaptation process. This is important in terms of understanding both the operation of the OMM (Open Mobile Miner) toolkit as well as the adaptive algorithms that form the core of OMM. However, in developing the platform, we have been conscious of the fact that there will be other mobile data stream mining algorithms that may or may not conform to adaptation. Furthermore, there may in the future be analysis algorithms that perform adaptation using varied strategies. Thus, the toolkit decouples the adaptation from the analysis such that algorithms can leverage the adaptation mechanisms or they can execute without adaptation.

Adaptation strategies in OMM can be categorized into two main classes: resource-aware and situation-aware strategies. To enable flexibility in OMM, adaptation can be achieved using each approach individually or by combining both approaches as a hybrid technique. The adaptation is performed transparently and is hidden from the user. The following subsections describe the underlying concepts of the resource and situation-aware adaptation.

2.1 Resource-Aware Adaptation

The dominating factor of mining stream data on mobile devices is the high input rate with regard to the available computational resources. Data Streams are generated and sent in real-time in a stream format. The input rates of data streams can range from hundreds of records per second to megabytes or terabytes of tuples per second (Gaber 2009). Given the fact that the state-of-the-art techniques in the area have only focused on data reduction or approximating the results in a low complexity of space and time, we have proposed to adapt the mining algorithm according to resource availability and data stream rate. This approach is termed granularity-based adaptation (Gaber 2009). The granularity-based adaptation approach has three different variations:

- AIG (Algorithm Input Granularity) is a process that adapts the data rates feeding into the algorithm according to the battery charge (see Figure 1).

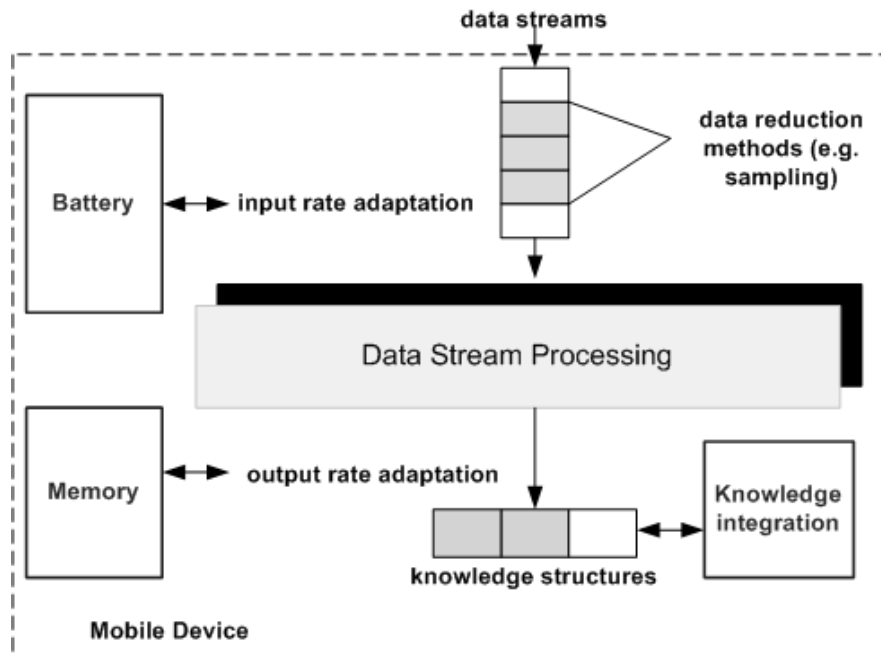


Figure 1 Input and output rate adaptation based on resource levels using AIG and AOG.

- AOG (Algorithm Output Granularity) provides adaptability by adjusting the algorithm output rate (e.g. the number of clusters) (see Figure 1).
- APG (Algorithm Processing Granularity) performs adaptation of the processing settings of the algorithm with respect to the CPU usage.

Resource-aware adaptation focuses on resources (i.e. memory, battery and CPU). Yet, the mobile data mining algorithm's cost-efficiency with regards to resource utilization can be improved further by factoring in the entire situational context of the application. This is due to the fact that the data mining application's requirements in terms of the accuracy (and therefore resource consumption) vary according to the current situations. The next subsection discusses the concept of situation-aware adaptation and the situation inference model that it applies.

2.2 Situation-Aware Adaptation

Resource-aware adaptation aims to adjust the algorithm input and output rates (i.e. the algorithm accuracy) according to the resource levels of mobile devices to preserve resources. When the resource levels are low, a resource-aware adaptation moderately reduces the algorithm accuracy by decreasing the input or output rates. A high level of accuracy (without using adaptation) consumes resources quickly and can result in the mobile application failure.

The accuracy requirements of a mobile data mining application can change based on the occurring situations. By situations we mean real-life situations such 'fire_threat' or 'driving'. There are certain situations in which applications do not need high accuracy such as the 'healthy/normal' situation in a health monitoring application. However, there are other situations like 'hypertension' (caused by high blood pressure) which will require a higher level of accuracy. A situation-aware approach can increase the accuracy during critical situations where

there is a need for closer monitoring and detailed output. However, when the current situation requires less frequent data analysis and less detailed mining results (i.e. low accuracy), this adaptation technique can decrease the algorithm accuracy to preserve resources.

2.2.1 Situation Inference

To provide situation-awareness, there is a need for a context modeling and reasoning technique that can represent the current situation and more importantly is able to infer the situations from low level context. Individual contextual parameters provide a limited view of the real-world and a partial understanding of the environment (Padovitz et al. 2004). Multiple contextual parameters can be aggregated by employing reasoning techniques and used for inferring situations (Padovitz et al. 2004). Fuzzy Situation Inference (FSI) technique (Delir Haghighi et al. 2008) is a novel context modeling and reasoning approach that we have developed to identify and represent real-world situations as well as the uncertainty associated with these situations. The inferred situations are used to enable a smooth and fine-grained adaptation of data mining algorithms' settings according to application constraints.

FSI integrates fuzzy logic into the Context Spaces (CS) model (Padovitz et al. 2004). It uses the benefits of the CS model for supporting pervasive computing environments while incorporating fuzzy logic to deal with uncertainty associated with real-world situations. In FSI, fuzzy rules can be specified for situations of interest by domain experts in the process of knowledge acquisition by FSI developers or designers. The rules can be extracted by some tools or manually. Once the rule repository has been developed it can be maintained and updated by domain experts in the same way that they are initially acquired. Additionally rules can be generated by data stream mining algorithms and based on their extracted knowledge (Gaber et al. 2004), and then validated by domain experts. Throughout the lifetime of rule repository, rules

can also be refined using data stream mining algorithms such as clustering or Detect Change algorithms (Gaber et al. 2005). It is worth mentioning that development of such a tool (or approach) for rule acquisition and maintenance goes beyond the scope of this paper. However such a tool will be very useful and is considered as our future research effort.

To model the importance of conditions, we assign a weight w to each condition with a value ranging between 0 and 1. The sum of weights is 1 per rule. A weight represents the importance of its assigned condition relative to other conditions in defining a situation. An example of a FSI rule is as follows:

IF systolic_blood_pressure is 'high' AND diastolic_blood_pressure is 'high' AND heart_rate is 'fast' THEN situation is 'hypertension'

To reason about a situation, rules need to be evaluated to produce a single output that determines the membership degree of the consequent (Zimmerman 1996). Using fuzzy logic, the FSI model is able to compute the individual contribution levels of context values using the trapezoidal membership function. The membership degree of an element represents its contribution level according to the definition of the CS model. The FSI proposes a basic technique for evaluation of FSI rules and conditions joined with the AND operators:

$$Confidence = \sum_{i=1}^n w_i \mu(x_i) \quad (1)$$

where w_i represents a weight assigned to a linguistic variable such as heart rate, and $\mu(x_i)$ denotes the membership degree of the element x_i given that it belongs to an associated fuzzy set. The membership degree represents the contribution level (i.e. c_i). The result of

$w_i\mu(x_i)$ represents a weighted membership degree of x_i and n represents the number of conditions in a rule ($1 \leq i \leq n$).

2.3 Adaptation Strategies and Underlying Concepts

The inferred situations by FSI and their membership degrees are used by situation-aware and hybrid strategies for adaptation of data mining algorithm settings.

- **Situation-Aware Strategy** - A situation-aware adaptation technique controls the data stream mining settings (i.e. input and output rates) according to the occurring situations and accuracy requirements of the running application. During adaptation, the pre-initialized parameters of mining algorithms such as sampling rate are adjusted according to the degree of membership (i.e. a value between 0 and 1) of occurring situations. The pre-initialized parameters are defined for each situation and reflect the accuracy needs of application during occurrence of that situation.
- **Hybrid strategy** - In the cases where both resources and situations are critical and there is a need for high accuracy, the situation-aware approach can result in draining the resources as it does not consider the resource availability. To address the issue and factor in both occurring situations and levels of resources, the hybrid strategy computes each algorithm's parameter value by considering the criticality values of situations and resources (i.e. battery and memory).

Since our adaptation approach includes situation- and resource-aware and hybrid strategies, it is important that the appropriate strategy is selected at run-time. The selection is performed based on the concepts of situation and resource criticality:

- **Situation criticality** - We model the application's accuracy requirement (and resource consumption) for a situation by the concept of situation criticality. The criticality of a situation can be expressed by a value between 0 and 1. If a situation requires closer monitoring and more detailed data analysis output, it should be given a higher criticality value (closer to 1), and if the situation needs a lower level of accuracy, it should be assigned a lower criticality value (closer to 0).
- **Resource Criticality** - Resource criticality is used to model the availability of resources and expressed as a value between 0 and 1. When a resource such as memory is fully available (i.e. 100%), its criticality value is 0 which implies it is not critical.

To define the low and high criticality levels, there is a need for using a point of reference that values can be compared to. This is achieved by assigning thresholds (i.e. a value between 0 and 1) to resource and situation criticality. These thresholds are application-specific and determined by system designers and application domain experts. For example, the situations above the upper bound threshold with a value of 0.7 can be considered as critical situations requiring high accuracy. The situations below the lower bound threshold which is assigned a value of 0.3 can be regarded as non-critical. Non-critical situations do not need high accuracy.

Using criticality values and thresholds enables the Controller to compare resources according to their levels and situations with regard to the application's accuracy requirement, and determine which strategy can achieve the required accuracy while using resources efficiently.

2.2.3 Criticality variations and the selection technique

The controller component of the situation and resource-aware adaptation framework is developed according to the four main variations. Table 2 presents these cases that can occur

according to the criticality of resources and situations during the application run, and shows the level of accuracy that is achieved by each adaptation strategy.

Table 2 Adaptation results considering criticality variations.

S. C	R.C	S.A. method	R.A. method	Hybrid method
Low	Low	Low accuracy	High accuracy	Moderate accuracy
Low	High	Low accuracy	Low accuracy	Low accuracy
High	Low	High accuracy	High accuracy	High accuracy
High	High	High accuracy	Low accuracy	Moderate accuracy

S.C., R.C., S.A. and R.A. stand for situation criticality, resource criticality, situation-aware, and resource-aware.

We now discuss the selection process based on the assumption that the low accuracy results in less resource consumption and the high accuracy increases the resource consumption.

- 1) *When the criticality values of both resources and occurring situation are low.* In such cases, the situation-aware technique aims to preserve resources by decreasing the accuracy because the criticality value of the occurring situation is low. Conversely, the resource-aware approach aims to increase the accuracy because of the resource availability. The hybrid method combines both situation and resource-aware methods, and therefore it attempts to maintain a moderate level of accuracy but higher than the situation-aware adaptation which is not needed by the application. Therefore, in such scenarios, the situation-aware technique can be considered a better choice.
- 2) *When the criticality value of occurring situation is low but the resource criticality is high.* In this case, the situation-aware strategy reduces the accuracy and resource consumption. Meanwhile, in such cases, since the resource criticality is high, the

resource-aware method also decreases accuracy to preserve resources. The hybrid strategy considers both resource availability and occurring situation and attempts to decrease the algorithm accuracy. Therefore, in this case either of the strategies (situation-aware or hybrid) can be selected. However, the hybrid technique requires more computation because it executes both resource and situation-aware methods. Hence, the situation-aware approach is preferred to the hybrid technique.

3) *When the criticality value of the occurring situation is high but the resource criticality is low.* In this scenario, the situation-aware strategy increases the input and output rates to meet the application's requirement for high accuracy. Since resources are available, the resource-aware method also aims to increase the accuracy. Hence the hybrid method that integrates the situation and resource-aware strategies results in high accuracy. With regards to this variation, the results of situation-aware and hybrid methods are similar but the situation-aware technique requires less computation and is considered a better choice.

4) *When the criticality values of resources and occurring situation are high.* In this scenario, the hybrid technique is a better choice as the situation-aware method will drain the resources to maintain high accuracy. The hybrid method considers both occurring situations and resource levels, and enables the algorithm to use resources efficiently while providing an acceptable level of accuracy that is required by the current situation.

Table 3 presents the notation used in the algorithm for selecting the adaptation strategies.

Table 3 Symbols used in the strategy selection algorithm.

Symbol	Meaning
R	Vector of resources $R = \{r_1, r_2, \dots, r_j\}$
S	Vector of inferred situations $S = \{s_1, s_2, \dots, s_i\}$
$\mu(s_i^{highest})$	Function returning the situation with highest degree of membership
$C(s_i^{highest})$	Criticality of the situation with the highest membership degree
$C(r_j)$	Criticality of a resource r_j

Figure 2 shows the algorithm used for selection of adaptation strategies. The Adaptation Engine (AE) periodically obtains resource levels and inferred situations. At the beginning of each time interval, the AE checks the criticality level of each resource. If all the resources are available (i.e. criticality value is low), AE triggers the situation-aware strategy. Situation-aware adaptation adjusts all the parameter values according to the occurring situation and returns the adjusted values of parameters used for controlling the mining algorithm settings.

```

1. Get  $R$  from RM
2. Get  $S$  from FSIE
3. Get the highest situation  $s_i^{highest}$ 
4. Get the  $C(s_i^{highest})$ 
5. For each  $r_j$ 
    Get the  $C(r_j)$ 
    If  $C(r_j)$  is low
        Do situation strategy
    Else
        If  $C(s_i^{highest})$  is low
            Do situation strategy
        Else
            Do hybrid strategy
        End If
    End If
End For

```

Figure 2 The pseudo code of Controller for selecting the adaptation strategies.

If one of the resources is running low, the AE checks the inferred situations reported by FSI Engine (FSIE) and considers the criticality value of the situation with the highest membership degree. The highest grade of membership implies the highest level of confidence in occurrence of a situation. The AE considers this situation as the current situation.

If the situation with the highest membership degree has a low criticality value, it means the application requires low accuracy, and the Controller executes the situation-aware adaptation again. However, if the occurring situation's criticality value is high, the Controller triggers the hybrid adaptation strategy that combines situation and resource-aware methods and uses the results of both to determine the adjusted value of the mining parameter.

3. An Overview of the OMM Toolkit

The Open Mobile Miner (OMM) toolkit is a generic toolkit for mobile data mining. OMM is easy to use and extensible, and can be deployed on a range of mobile devices and customized for application specific needs. OMM leverages a holistic adaptation approach for mobile data mining that we have developed. Figure 3 shows the OMM architecture.

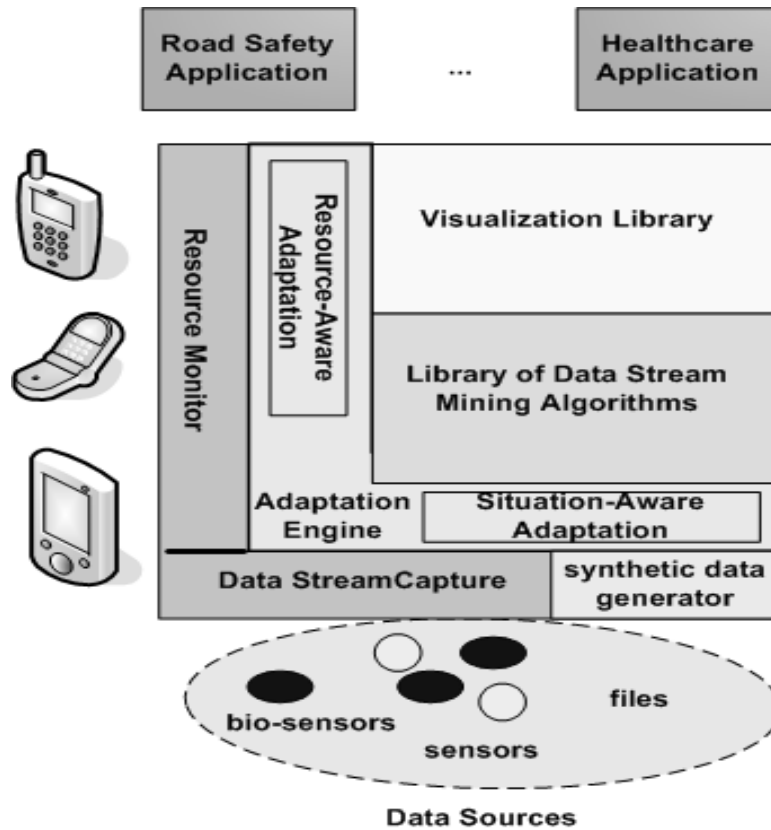


Figure 3 An overview of the Open Mobile Miner (OMM) toolkit.

OMM presents an important step forward in taking mobile data mining from theory to real-world application development and deployment. The key components of the architecture are as follows:

Data Sources - The streams of data that need to be analyzed are generated at the data sources. OMM can receive and analyze data from four different sources: i) sensors or biosensors that transmit either through Bluetooth, WiFi, or other protocols; ii) a data generator that can generate a specified number of streams each with a specified distribution (e.g. Binomial, Gaussian, Poisson, Uniform etc.), for the specified parameters; iii) reading recorded data in a local CSV file and re-play it as a stream; iv) replay the contents of a CSV file as a stream from a web source.

Data Stream Capture - This component receives data streams from the various sources and passes it either to the data stream mining algorithms or the adaptation engine depending on whether the analysis process has been initialized to operate in an adaptive manner or not. This component may perform some buffering of data so as to enable determining the data rate and preventing loss of data.

Resource Monitor - This component is responsible for assessing the levels of memory, processor and battery available on the device. This information in conjunction with the data stream rates constitutes the principal basis for performing adaptation. Resource monitor primarily communicates the resource level information to the Adaptation Engine. This component is – unlike the others – operating system specific. Given the range of mobile devices that are being developed and their diverse operating systems (e.g. Nokia phones run the Symbian OS, Google GPhone runs the Android OS and the iPhone runs iPhone OS) – this component has to implement the OS specific functions to access low-level computational characteristics.

Library of Data Stream Mining Algorithms - The analyzer library provides a range of data stream mining analysis algorithms for mobile data mining. Table 4 shows the implemented algorithms in OMM (discussed in Section 3). All these algorithms are able to operate on real-time data streams such as data from sensors or biosensors.

Table 4 A list of OMM Algorithms.

Method	Algorithm
Classification	LightWeight Classification (LWClass) (Gaber et al. 2004) integrates the AOG concept into K-Nearest-Neighbours classification.
Clustering	LightWeight Cluster uses an AOG-based clustering algorithm that considers a threshold distance measure for clustering of data (Gaber et

	al. 2004)
	RA-VFKM integrates AOG with VFKM (Very Fast K-means) (Shah et al. 2005)
	RA-Cluster and ERA-Cluster (Gaber and Yu 2006; Phung et al. 2007) is an adaptive micro-clustering algorithm using concepts of AOG, AIG and APG.
Time series analysis	RA-SAX (Resource- Aware version of Symbolic Approximation (SAX)) is a resource-aware time series analysis technique (Lin et al. 2003).
Frequent Items	LightWeight frequent items (Gaber and Yu 2006) is based on AOG first calculates the number of frequent items according to the available memory and adjusts this number to deal with the high data rates.

Visualization Library - The visualization library allows the results of the analysis process to be shown using custom visualization techniques. Given that many applications will typically require custom visualizations, the toolkit needs to facilitate integration of application specific visualization. The visualization middleware performs the task of continuously obtaining the output of the algorithms (e.g. cluster details) as they are available and also maintains limited generic information regarding visualization preferences (e.g. colors and shapes used to represent clusters). It is noteworthy that visualization of data stream mining on mobile devices is very much an emerging area of study. As such there are only early results on generic visualization algorithms/techniques that are available in the literature (Gillick et al. 2010). There are many challenges such as coping with incremental results; dynamic changes in the analysis results, coping with the limited screen real-estate that needs to manage screen-clutter as it evolves and having an effective battery-consumption strategy. Clearly, there is also a need for user-evaluation

in terms of the HCI issues, as well as tailoring of visualizations suited for different kinds of analysis. As such, our approach in this context has been to design OMM such that there are mechanisms to make the output accessible from the analysis process via a visualization middleware and enable the application developers to integrate application-specific visualizations.

Adaptation Engine - This component manages the adaptation process in terms of obtaining information regarding the data stream characteristics (e.g. data rates) from the data sources, resource-levels (i.e. status of computational resources including battery levels) of the device (i.e. resource criticality) and situation criticality, and then instrumenting the performance of the data stream mining algorithms according to this information.

The Adaptation Engine has two main strategies for adjusting dynamically the functioning of the data mining algorithms according to the various parameters by varying accuracy levels. These strategies include resource and situation-aware techniques that can be used individually or combined as a hybrid approach according to the principles outlined previously in Section 2.

The next section discusses how this has been implemented in the OMM toolkit.

4. Implementation of OMM

The motivation for the development of the Open Mobile Miner (OMM) was to provide a generic tool to facilitate research on mobile data mining. The OMM toolkit is split into two parts: A Core that provides all the functionality needed to do adaptive mobile mining and a graphical user interface (GUI) that facilitates ease of use for the Core's functionality through graphical controls. Figure 4 illustrates the implementation structure of OMM.

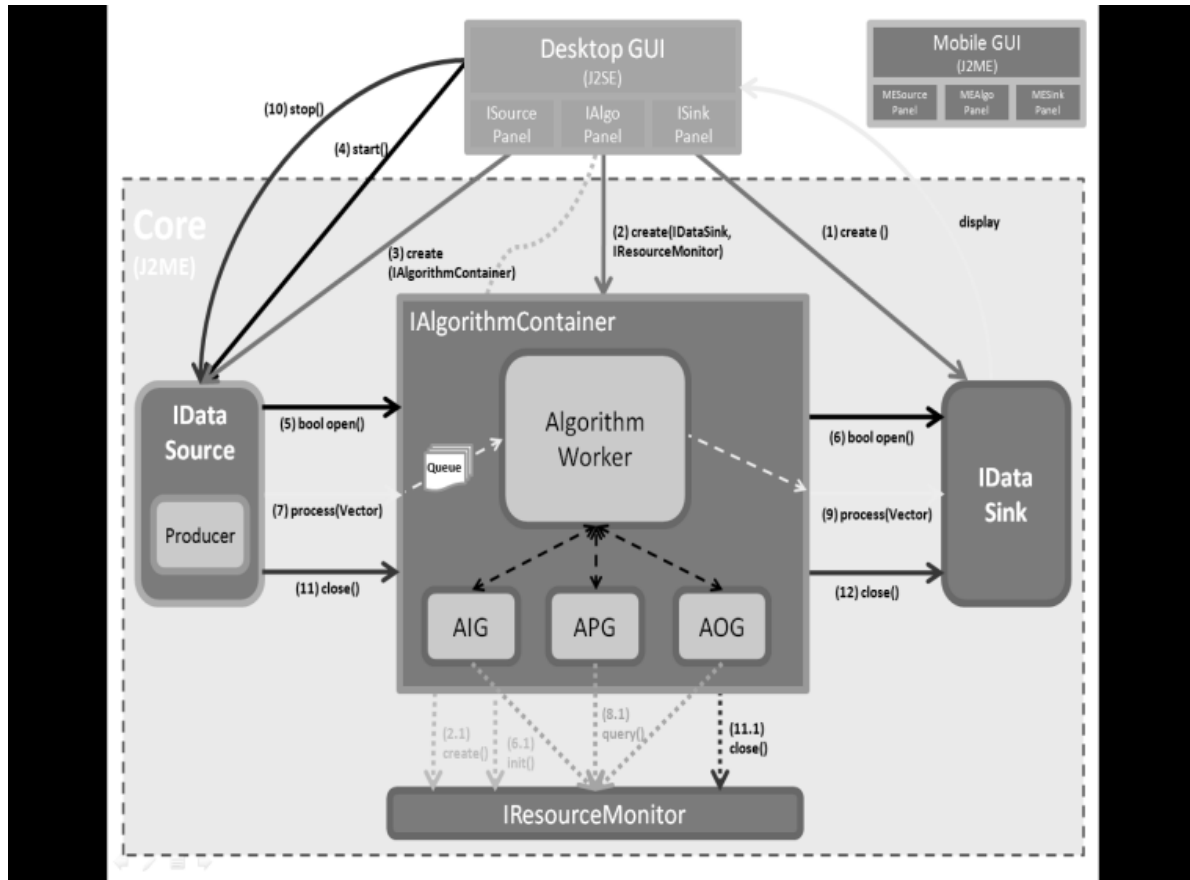


Figure 4 The OMM Toolkit's implementation structure and its main components.

4.1 The Core

The Core consists of three major interfaces: IDataSource, IDataSink and IAlgorithmContainer.

The Core utilizes three utility interfaces. These include IResourceMonitor, ISituationMonitor and IStatsConsumer to provide support for resource awareness, situation awareness and runtime statistics respectively. Within OMM's core, the data just keeps flowing upstream through an algorithm. The data source acts as an adaptor for the system to the incoming data stream converting items into the necessary format. In turn, the data sink can be used to transform results into any desired format for visualization. Generally, the data path is set up as follows:

1) A sink is created and then the sink is passed as an argument to the algorithm container.

The algorithm container supports an arbitrary number of sinks in order to output data in several ways concurrently. Due to simplicity, Figure 4 only shows one sink.

2) Additionally, the algorithm container will generally need an `IResourceMonitor` and/or an `ISituationMonitor` to implement the adaptation strategies presented earlier in the paper.

As such OMM can perform analysis with no adaptation, or any other type such as resource-aware adaptation, situation-aware adaptation or hybrid adaptation.

3) As a last step, the `IAlgorithmContainer` reference is passed into the `IDataSource` to establish the link between them. OMM supports a wide variety of data sources as explained in Section 3.

4.2 The GUI

OMM's GUI provides an interactive GUI with graphical controls for easy use and performing experiments. The core functionality is accessible from the GUI by selecting the components to connect. The user is required to enter the necessary parameters for the respective source, sink or algorithm and can eventually run the system. Furthermore, a tight integration with any software can be achieved by accessing the OMM Core functions directly via the API. This is done in a straightforward manner by instantiating component classes directly. To setup the system, one selects source, algorithm and sink (see Figure 5). After pressing the select button, a tree of available components is shown. After making a selection, a box containing the available parameters is displayed allowing adjustment of the component's behavior as required. If the output should be displayed in the GUI's output tab, the `SEGUISinkWidget` (from the list) has to be selected. OMM also allows saving the current selection and configurations from the widgets

into an XML file. This file can be loaded back into the GUI at another point of time or deployed on a mobile device and used to run OMM without having to configure it manually beforehand.

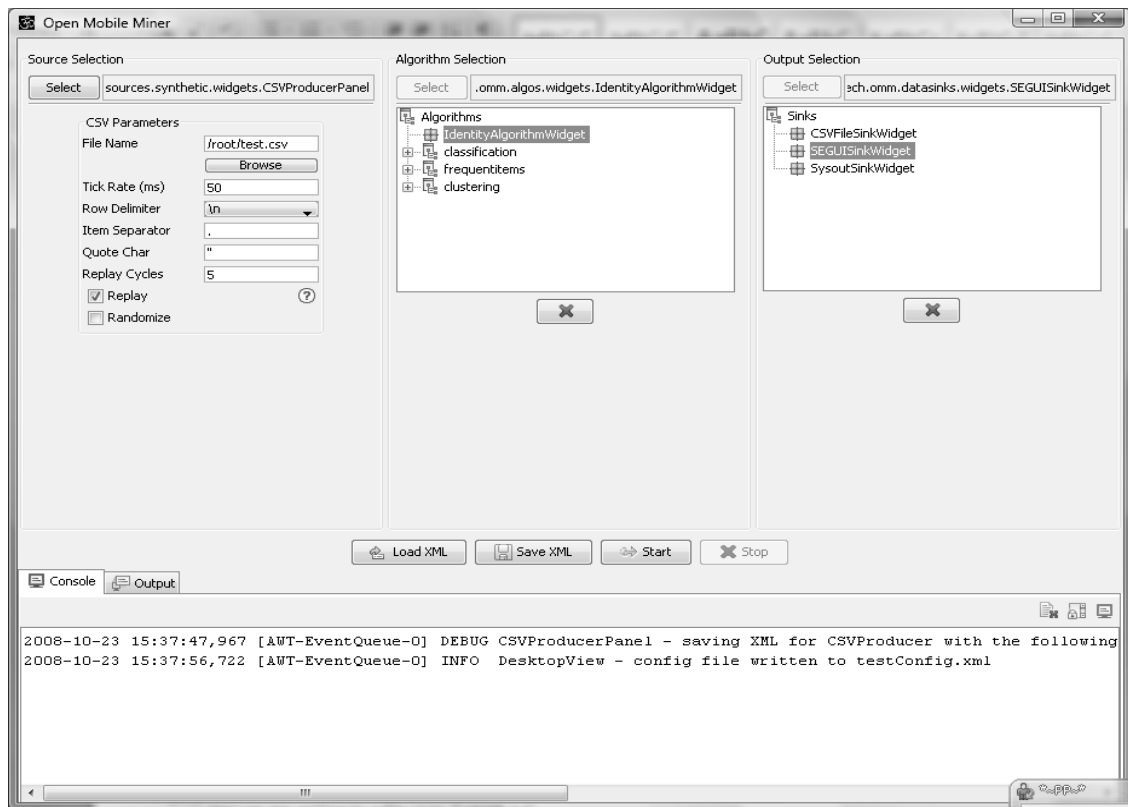


Figure 5 An overview of the OMM Desktop GUI.

The mobile GUI is similarly structured to the Desktop GUI. It can be configured to load configurations from an XML file previously generated by the Desktop GUI using the “Load” option on the welcome screen. The OMM GUI is easily extensible. For instance, a new custom source can be instantiated by including an `ISourcePanel` on the classpath. The OMM GUI will support the new source and display it as an option in the respective component’s tree listing.

4.3 Visualization

As discussed earlier, OMM’s visualization middleware enables to visualize the analysis results. Figure 6 illustrates a custom visualizer that displays the results of the RA-Cluster algorithm.

These results represent clustering real-time locations of emergency and police personnel during an emergency. Such a real-time analysis and visualization could enable emergency authorities to quickly understand the areas where the impact is greatest and allow re-deployment of personnel in real-time. The visualizer uses color and size to visualize the evolving cluster strengths, and is adaptive to screen clutter, cluster overlap, and varying energy levels on the phone. It also allows the visualization to be personalized using various visualization thresholds (Gillick et al. 2010).

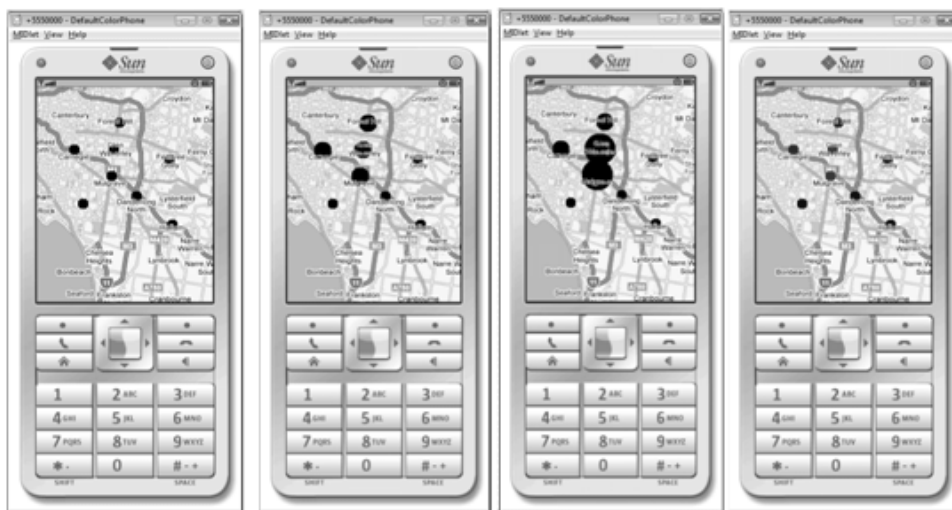


Figure 6 The results of RA-Clustering captured by custom-built cluster visualizer.

The preceding sections presented the conceptual framework and the implementation of the OMM toolkit along with the theoretical underpinnings of its adaptation strategy. We now present the evaluation of the platform for developing and deploying efficient mobile data mining applications. Our evaluation strategy is twofold. Firstly, we aim to show how mobile data applications can be easily configured and deployed in a completely flexible way using the OMM toolkit. Our second aim is to present the effectiveness and efficiency of the situation-aware adaptation strategy and demonstrate the improvements it brings to mobile data mining applications when compared with the previous state-of-the-art resource aware adaptation strategies.

We now present a case study which shows the use of OMM to develop a mobile healthcare application which applies situation-aware and hybrid techniques.

5. Mobile Data Mining For Healthcare: A Case Study And Experiments

Mobile healthcare and patient monitoring technology are becoming increasingly prevalent.

Recently, innovations in mobile communications and low-cost of wireless biosensors have paved the way for development of mobile healthcare (Leijdekkers and Gay 2012; Rodriguez, Goni and Illarramendi 2015) that provide a convenient and constant way of monitoring of vital signs of patients. A significant challenge for healthcare monitoring applications is to process and analyze continuous data streams with resource constrained devices such as a smart phone in real-time. Our proposed adaptation strategies and light-weight mining algorithms provided by OMM can significantly benefit the mobile healthcare applications to address this challenge.

In the following section, we present the case study of a mobile patient monitoring application using OMM.

5.1 A Mobile Health Monitoring Application

We have implemented a mobile health monitoring prototype that applies the situation-aware and hybrid adaptation approaches to the ERA-Cluster algorithm. The prototype is built for patients who suffer from blood pressure fluctuations and reasons about the health-related situations including ‘normal/healthy’ and ‘hypertension’ (caused by high blood pressure). The context attributes used for this application include systolic and diastolic blood pressure, room temperature and heart rate, which are obtained from a Bluetooth-enabled ECG biosensor from Alive Technology (Alive Technology) attached to the patient’s chest. The data mining algorithm that we used in our prototype is the ERA-Cluster algorithm (Phung et al. 2007). ERA-Cluster is a

resource-aware clustering algorithm extended from RA-Cluster (Gaber et al. 2006) that targets wireless sensor networks. Similar to RA-Cluster, the settings of the ERA-Cluster algorithm can be adapted to changes in battery level and remaining memory using the concepts of the granularity-based adaptation. The prototype is implemented in J2ME and tested on the Nokia N95 mobile. The architecture and its implementation are depicted in Figure 7.

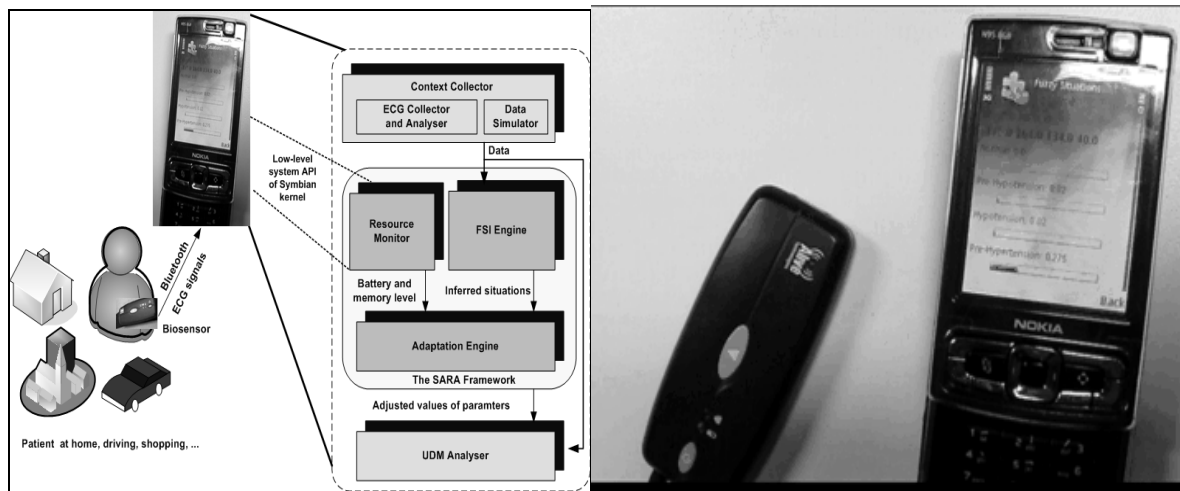


Figure 7 The architecture of the health monitoring prototype and its implementation.

5.2. Accuracy evaluation of ERA-Cluster

The ERA-Cluster algorithm is an example of the OMM's resource-aware mining algorithms. It performs resource-aware adaptation by adjusting the input and output rates (and accuracy) according to the resource availability. During the adaptation, it is important that the input and output rates are changed/adjusted within the certain lower and upper bound thresholds in order to maintain an acceptable level of accuracy.

To determine the lower and upper bounds for ERA-Cluster, Phung et al. (2007) performed a comparative evaluation of the ERA-Cluster with the well-known and widely-used kmeans algorithm of Weka (Witten and Frank 2001). In the evaluation, ERA-Cluster was run

over the dataset with 660 records to create a number of microclusters. Over the same synthetic data, kmeans was run 3 times with $k = n$ to create the same number of clusters. Figure 8 shows the results which indicate that ERA-Cluster is able to maintain a similar level of accuracy compared to kmeans while performing resource-aware adaptation. According to this experiment, the lower and upper bounds of 100 and 400 for the sampling intervals, and the radius with minimum and maximum values of 4 and 45 could produce an acceptable level of accuracy.

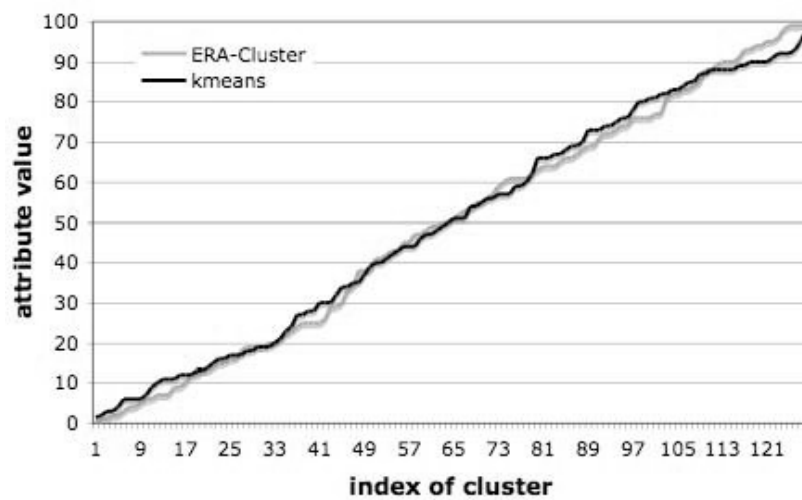


Figure 8 Evaluation of ERA-Cluster and kmeans (adapted from Phung et al. (2007)).

To demonstrate that our adaptation methods can improve lifetime without reducing accuracy levels, we maintain the exact same lower and upper bounds for the ERA-Cluster algorithm as done in (Phung et al. 2007) but control the algorithm accuracy using situation-aware and hybrid strategies.

5.3 Comparative experimental evaluation

Previous studies in mobile data mining (Gaber et al. 2005; 2006; Gaber 2009) experimentally evaluated resource consumption of mobile devices with and without the resource-aware approach and their results showed that the resource-aware adaptation can preserve resources and

improve the cost-efficiency of data mining algorithms. Therefore, in our evaluation, we compared the situation-aware (SA) and hybrid techniques only with the resource-aware (RA) method to show the benefits of our approach over the resource-aware technique.

5.3.1 Settings

In our experiments, we used a resource-aware data mining algorithm named ERA-Cluster (Phung et al. 2007). The cost-efficiency of mobile data mining algorithms is measured with respect to the longevity of mining operations (i.e. running time) and the level of availability of resources (i.e. memory and battery charge). ERA-Cluster provides these adjustable parameters: (i) sampling interval for controlling the algorithm input and thereby battery consumption; and (ii) the cluster's radius distance measure for adaptation of the output rate that impacts the memory usage. The sampling interval has an application-specific lower and upper bounds of 100 and 400, and the radius is assigned with minimum and maximum values of 4 and 45. These values are based on the results of experiments discussed in Section 5.2 and are specific to ERA-Cluster.

We consider the four variations (see Table 2) based on the two levels of low and high for resources and two levels of critical and non-critical for situations. Considering our health monitoring application, the critical situation applies to 'hypertension' and the non-critical situation is associated with 'normal/healthy'. The criticality threshold values that we use are application-specific. For situation criticality, we assign two thresholds of 0.3 and 0.7, and for resource criticality, we define the lower and upper bound thresholds of 0.15 and 0.45 based on our observations of resource consumption patterns in the Nokia N95 phone.

For the first three variations (see Table 2), we compare the situation and resource-aware methods (i.e. total of 6 different runs). This is due to the fact that the hybrid method is proposed

for those scenarios where both resources and situations are critical and this does not apply to the first three cases. For the last variation, we have compared the hybrid, SA and RA methods (i.e. total of 3 different runs). We have repeated each application run five times and used the average result in our evaluation.

During the application run the mobile phone SIM card was not removed because we were interested to conduct our experiments in real-world settings. Since smart phones' functionalities such as voice calls, text messaging, web browsing, playing video or audio and running applications can significantly affect the power consumption; during the experiments we did not use any of these functionalities and kept the phone in an idle state. The mobile phone's operating system can also improve power management by using Battery Saver or Power Saver modes that controls the functions such as screen brightness. During experiments, the phone was not used for any other purposes other than testing, and there was not any factor controlled by the operating system which could have impacted our results.

5.3.2 Test Data

The data has been generated in a range such that simulates the occurrence of each health-related situation (according to fuzzy sets of FSI rules). However, to consider the energy consumption by the Bluetooth communication between the sensor and the mobile phone, it was important to include the ECG sensor in the experiments. Hence, we used the ECG sensor and the mobile phone was continuously receiving the sensory data. However, we overwrote this data by the simulated heart rate to simulate the critical situations. The Alive biosensor's ECG data has the following structure: packet header (6 bytes), ECG header (5 bytes), ECG data (n bytes), accelerometer header (5 bytes), accelerometer data (n bytes) and a checksum 1 byte. To process

and convert the ECG signals into heart rate data we used the MobiHealth¹ open source framework which enables collecting ECG signals and computing the heart rate.

The complete dataset for each situation consists of approximately 60,000 records. Each record consists of four data elements that represent the values of systolic and diastolic blood pressure, heart rate and room temperature. To perform a fair evaluation and to use the same data in each repeating run (for each situation), the generated datasets have been saved to three files (corresponding to each situation). At the start of each repeating run, the stored data is read and fed into a data generator program that publishes the data with at a rate of 1 record/100 msec. The complete dataset for each situation consists of approximately 60,000 records.

5.3.3 Experiments 1 and 2 for Non-Critical Situations

The first experiment is performed for scenarios in which resources are available (i.e. resource availability level between 100-85%) and the occurring situation is non-critical (i.e. both situation and resource at the low criticality level) and applications do not need high accuracy. We compare the results of our experiments for each strategy (RA and SA) based on the application running time, memory and battery consumption and parameter values of the algorithm adjusted according to each strategy. During Experiment 1, the situation-aware method increases the sampling interval of mining algorithm to reduce the input rate and therefore it improves the conservation of battery and longevity of operations. The RA technique uses a lower sampling rate to maintain a higher level of accuracy because of availability of battery. This leads to more computation and consumption of battery, and decreases the running time of application.

¹ <http://sourceforge.net/projects/mobhealth/>

Experiment 2 considers scenarios in which the occurring situation is non-critical but resources are running low which refers to the resource availability level between 84 and 50%. In such scenarios, the situation-aware method regardless of resource levels decreases the accuracy due to the application's needs, leading to less resource consumption. On the other hand, the RA method adapts the settings of the mining algorithm and moderately decreases the accuracy to deal with low level of resources. In these cases, although each strategy considers different factors, the results are similar with respect to accuracy and running time.

The summary of the results of Experiment 1 and 2 considering battery and memory usage are depicted in Figure 9. The bar chart is created based on the memory and battery level values and running time of the application in seconds. The taller columns indicate more efficiency and shorter columns represent faster consumption of resources and lower efficiency.

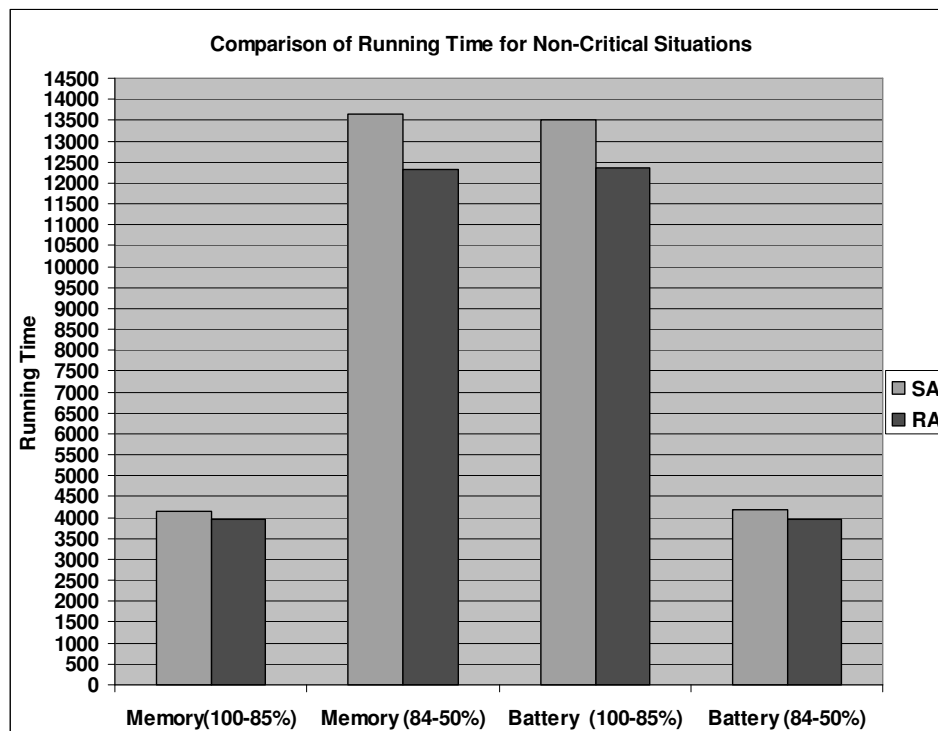


Figure 9 Comparison of SA and RA methods for non-critical situations when resources are available and when they are running low.

The overall results indicate that the SA strategy outperforms the RA method and is able to improve the cost-efficiency and continuity of mining operations. This improvement is more noticeable when the battery criticality level is at low level (i.e. 100-85%) and the memory criticality is at high level (i.e. 84-50%). During experiments we observed that when resources are at the low criticality level, it takes a longer time for the battery level to drop to 85% and when resources are running low, memory is consumed slower than battery.

5.3.4 Experiment 3 and 4 for critical situations

In Experiment 3, we consider the cases in which situations are critical and resources are available. In critical situations, the situation-aware method increases the algorithm accuracy due to the needs of application for a higher level of accuracy. This approach will lead to more consumption of resources. Alternatively, the resource-aware method that is performed regardless of situations considers the resource levels to determine the accuracy. Therefore, in this scenario for Experiment 3, both RA and SA attempt to increase the accuracy and there is not a trade-off between two approaches. Hence, we have not considered the hybrid strategy in Experiment 3. The results of Experiment 3 based on battery and memory usage are illustrated in Figure 10a.

Experiment 4 is performed for cases when both situations and resources are critical. In such cases, situation-aware method increases the accuracy and thereby consuming more resource. However, the resource-aware technique reduces the accuracy to deal with the low level of resources. To address this trade-off and to enable optimal use of resources while considering application's need for higher accuracy, the hybrid strategy considers both occurring situation and resource availability. Hence, in Experiment 4, we compare three approaches of SA, RA and hybrid considering both battery and memory are shown in Figure 10b.

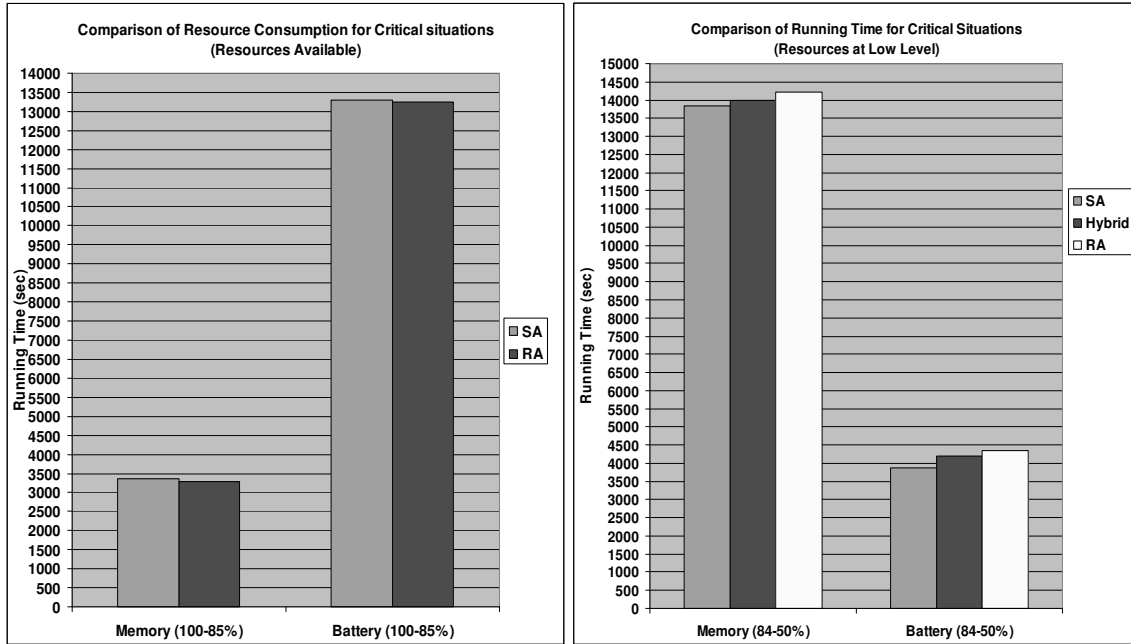


Figure 10 (a) Comparison of SA and RA methods for critical situations when resources are available, (b) Comparison of SA, RA, and hybrid methods for critical situations when resources are running low.

In the Experiments 1, 2 and 3, the situation-aware technique is able to adapt the mining algorithms based on the accuracy needs of applications and improve the running time of application by preservation of resources. However, in Experiment 4 that both resources and situations are critical, the SA method is performed without considering the resources and can lead to the application failure. Alternatively, the RA method can achieve a longer running time but it does not consider the criticality of situations. In these cases, the hybrid strategy provides an elegant solution by resolving the trade-off between RA and SA methods and taking into account both resource levels and situations.

5.3.5 Estimations of overheads/costs

To measure the energy overhead of the situation and resource-aware framework with respect to the battery consumption, we run our health monitoring application when our framework is

enabled and when it is disabled (when only the mining algorithm runs), and then compare the results. We have performed 5 runs of each case to perform this evaluation and considered the average results in the comparison. Figure 11 shows the evaluation results. The average running time is 31260 seconds (i.e. 8 hrs and 41 min) when the situation and resource-aware adaptation framework is disabled and 30672.4 seconds (i.e. 8 hrs, 31 min and 12 sec) when it is enabled.

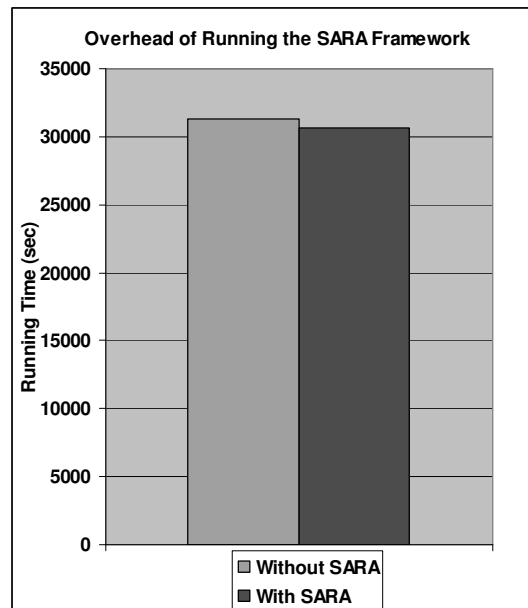


Figure 11 The overhead of running the application with adaptation in terms of battery usage.

This implies that operating the mobile data mining algorithm application with our framework tends to decrease the running time by approximately 10 minutes (i.e. 1.9% overhead). The situation and resource-aware adaptation framework is a light-weight software component that targets mobile devices and is able to maintain a minimal computational overhead. This is a marginal overhead of 1.9% in terms of energy usage. However, as shown in previous experiments, considering the energy savings that we obtain by having situation and resource-aware adaptation for the mining algorithm, we improve energy utilization up to 9.4%. Thus, we

can conclude that while there is a small processing usage overhead that our proposed framework has, this is offset by the benefits that it provides.

6. Conclusion

In this paper we have presented the architecture and implementation of the first integrated platform for mobile data stream mining. The innovation of OMM lies in not only the range of data stream mining techniques available for mobile data mining, but also its integrated and holistic adaptation strategies which have been established as essential factor for enabling real-time mobile data analysis. Furthermore, the toolkit has been shown to effectively enable a diverse range of information systems that incorporate mobile analysis applications. Finally, we have also demonstrated through our experimental evaluation, the efficacy and improved performance that situation-aware and hybrid adaptation strategies deliver over the state-of-the-art approaches which only factor in resource availability.

While the focus of the case study and performance evaluation presented in this paper is on mobile healthcare, we have also conducted experiments by using OMM for other applications such as real-time location analysis using clustering of GPS data as well as real-time analysis of stock market data. This demonstrates further the generic and flexible capability of the OMM toolkit to deploy and deliver a range of mobile data mining applications particularly for e-commerce, marketing, online shopping, etc.

As part of future work, we intend to use data stream mining algorithms for generating the rules that define situations as well as refining and maintaining the rule repository according to new patterns and changes in data. We are also working on extending the OMM toolkit for analyzing huge amounts of real-time data collected by mobile phone sensors using cloud

technologies, and aim to evaluate the feasibility and validity of OMM's mobile analytics as an effective mechanism for supporting large-scale mobile applications.

Acknowledgements

We express our deep thanks to Prof Frada Burstein for her advice and insightful suggestions with regard to this work and paper.

References

- Bifet, A., G. Holmes, B. Pfahringer, P. Kranen, H. Kremer, T. Jansen, and T. Seidl. 2010. "MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering." In *Proceedings of the 1st Workshop on Applications of Pattern Analysis, 1-3 Sep, 2010*, 11: 44-50.
- Bose I., and X. Chen. 2009. "Hybrid Models Using Unsupervised Clustering for Prediction of Customer Churn." *Journal of Organizational Computing and Electronic Commerce*, 19(2):133-151.
- Brezmes, T., J. Gorricho, and J. Cotrina. 2009. "Activity Recognition from Accelerometer Data on a Mobile Phone." *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, Lecture Notes in Computer Science, 5518: 796-799.
- Delir Haghighi, P., S. Krishnaswamy, A. Zaslavsky, and M. M. Gaber. 2008. "Reasoning about Context in Uncertain Pervasive Computing Environments." *EuroCSS 2008*, Zurich, Switzerland: 112-125.
- Delir Haghighi, P., A. Zaslavsky, S. Krishnaswamy, M. M. Gaber, and S. Loke. 2009. "Context-Aware Adaptive Data Stream Mining." *Journal of Intelligent Data Analysis* 13(3): 423-434.
- Delir Haghighi, P., M. M. Gaber, S. Krishnaswamy, and A. Zaslavsky. 2010. "Situation-Aware Adaptive Processing (SAAP) of Data Streams." In *Pervasive Computing : Innovations in Intelligent Multimedia and Applications*, edited by Hassanien A. et al., Springer, London: 313-338.
- Domingos, P., and G. Hulten. 2001. "A General Method for Scaling Up Machine Learning Algorithms and Its Applications to Clustering." In *Proceedings of Machine Learning Conference*: 106-113.
- Fua, T., F. Chunga, R. Luka, and C. Ngb. 2008. "Representing Financial Time Series Based on Data Point Importance." *Engineering Applications of Artificial Intelligence* 21(2): 277-300.
- Gaber, M.M., S. Krishnaswamy, and A., Zaslavsky. 2004. "Ubiquitous Data Stream Mining." In *Proceedings of Current Research and Future Directions Workshop held in conjunction with 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2004.

- Gaber, M.M., S., Krishnaswamy, and A. Zaslavsky. 2005. "On-Board Mining of Data Streams in Sensor Networks." In *Advanced Methods of Knowledge Discovery from Complex Data*, edited by S. Badhyopadhyay, U. Maulik, L. Holder and D. Cook, Springer Verlag.
- Gaber, M.M., and P. S. Yu. 2006. "A Holistic Approach for Resource-Aware Adaptive Data Stream Mining," *Journal of New Generation Computing* 25(1): 95-115.
- Gaber M. M. 2009. "Data Stream Mining Using Granularity-based Approach", In *Foundations of Computational Intelligence* Vol. 6, edited by Abraham A., Hassanien A., and V. Snase, 47-66.
- Gaber, M.M., S. Krishnaswamy, B. Gillick, N. Nicoloudis, J. Liono, H. AlTaiair, and A. Zaslavsky. 2010. "Adaptive Clutter-Aware Visualization for Mobile Data Stream Mining." *The 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI'10)* 2:304-311.
- Gentry, J. A., M.J. Shaw, A.C. Tessmer, and D.T. Whitford. 2002. "Using Inductive Learning to Predict Bankruptcy." *Journal of Organizational Computing and Electronic Commerce* 12(1): 39-57.
- Gillick, B., S. Krishnaswamy, M. M. Gaber, and A. Zaslavsky. 2006. "Visualisation of Fuzzy Classification of Data Elements in Ubiquitous Data Stream Mining." In *Proceedings of the 3rd International Workshop on Ubiquitous Computing, ICEIS Press*: 29-38.
- Gillick, B., H. AlTaiair, J. Liono, S. Krishnaswamy, N. Nicoloudis, M. M. Gaber, A. Sinha, and A. Zaslavsky. 2010. "Clutter-Adaptive Visualisation for Mobile Data Mining." *Demo and Short Paper in the 10th IEEE Intel Conf on Data Mining (ICDM2010)*, Sydney, Australia.
- Hannay, P. and G. Baatard. 2011. "GeoIntelligence: Data Mining Locational Social Media Content for Profiling and Information Gathering." In *Proceedings of The 2nd International Cyber Resilience Conference*, Perth, Western Australia:29-37.
- Jin, Z., S. Yuwen, and A.C. Cheng. 2009. "Predicting Cardiovascular Disease from Real-Time Electrocardiographic Monitoring: An Adaptive Machine Learning Approach on a Cell Phone." *Conference of the IEEE on Engineering in Medicine and Biology Society*, Minneapolis, MIN.
- Kargupta, H., B. Park, S. Pittie, et al. 2002. "MobiMine: Monitoring the Stock Market from a PDA." *ACM SIGKDD Explorations* 3(2): 37-46.
- Kargupta, H., R. Bhargava, K., Liu, et al. 2004. "VEDAS: a Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring." In *Proceedings of the 4th SIAM DM Conference*. Lake Buena Vista, Florida, USA, April 22-24.
- Kargupta, H., M. Gilligan, V. Puttagunta, K. Sarkar, M. Klein, N. Lenzi and D. Johnson. 2010. "MineFleet: The Vehicle Data Stream Mining System for Ubiquitous Environments." *Ubiquitous Knowledge Discovery*, Lecture Notes in Computer Science 6202/2010: 235-254.
- Krishnaswamy, S., M. M. Gaber, M. Harbach, C. Hugues, A. Sinha, B. Gillick, P. Delir Haghighi, and A. Zaslavsky, "Open Mobile Miner: A Toolkit for Mobile Data Stream Mining", *Demo and short*

- paper, the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2009, Paris, Accessed July 5, 2012. <http://eprints.port.ac.uk/4140/1/D02-kdd09demo.pdf>.
- Leijdekkers, P., and V. Gay. "User Adoption of Mobile Apps for Chronic Disease Management: A Case Study Based on myFitnessCompanion®." *Impact Analysis of Solutions for Chronic Disease Prevention and Management*, Lecture Notes in Computer Science, 2012, Vol. 7251/2012, 42-49
- Lin, J., E. Keogh, S. Lonardi, and B. Chiu. 2003. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms." *the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, San Diego, California: 2–11.
- Padovitz, A., S. Loke, and A. Zaslavsky. 2004. "Towards a theory of context spaces." In *Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication Workshops*: 38-42.
- Perich, F., A. Joshi, T. Finin, and Y. Yesha. 2004. "On Data Management in Pervasive Computing Environments." *IEEE Transactions on Knowledge and Data Engineering Archive* 16(5):621-634.
- Phung, N. D., M. M. Gaber, and U. Roehm. 2007. "Resource-Aware Online Data Mining in Wireless Sensor Networks." *IEEE Symposium on Computational Intelligence and Data Mining*:139-146.
- Raghunathan, V., C. Schurgers, S. Park, and M.B. Srivastava. 2002. "Energy-Aware Wireless Microsensor Networks", *IEEE Signal Processing Magazine*: 40–50.
- Rodriguez, J., A. Goni, and A. Illarramendi. 2005. "Real-time classification of ECGs on a PDA", *IEEE Transactions on Information Technology in Biomedicine* 9(1): 23-34.
- Shah, R., S., Krishnaswamy, and M.M. Gaber. 2005. "Resource-Aware Very Fast K-Means for Ubiquitous Data Stream Mining", In *Proceeding of 2nd International Workshop on KD in Data Streams*, ECML/PKDD.
- Stahl F., M.M. Gaber, P. Aldridge, D. May, H. Liu, M. Bramer, and P. S. Yu. 2012. "Homogeneous and Heterogeneous Distributed Classification for Pocket Data Mining." *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, Lecture Notes in Computer Science 5(7100).
- Stager, M. 2007. "Power and Accuracy Trade-Offs in Sound-Based Context Recognition Systems." *Pervasive and Mobile Computing* 3(3): 300-327.
- Talia, D. and Trunfio, P. 2010. "Mobile Data Mining on Small Devices through Web Services." in *Mobile Intelligence*, edited by L. T. Yang, A. B. Waluyo, J. Ma, L. Tan. NJ, USA: John Wiley & Sons.
- Witten, I. H., and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition. Boston: Morgan Kaufman.
- Zappi, P., C. Lombriser, T. et al. 2008. "Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection." *Wireless Sensor Networks* 4913: 17-33.
- Zimmermann, H. 1996. *Fuzzy Set Theory - and Its Applications*. Norwell, Massachusetts: Kluwer Academic Publishers.