AMERICAN UNIVERSITY OF BEIRUT

Holistic Approach to Energy Efficiency in Context-Aware Mobile Sensing

by

Raslan Hussein Kain

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
December 20 2019

AMERICAN UNIVERSITY OF BEIRUT

Holistic Approach to Energy Efficiency in Context-Aware Mobile Sensing

by

Raslan Hussein Kain

Approved by:

Prof. Hazem Hajj, Associate Professor — Advisor
Electrical and Computer Engineering

Prof. Zaher Dawy, Professor — Member of Committee
Electrical and Computer Engineering

Prof. Rabih Jabr, Professor — Member of Committee
Electrical and Computer Engineering

Dr. Sirine Taleb, External — Member of Committee
Electrical and Computer Engineering

Date of thesis defense: December 20, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Kain Raslan Hussein
Last First Middle

☒ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

□ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☒ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three X years from the date of submission of my thesis , dissertation or project.

07/02/2020
Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would like to extend my appreciation to Prof. Hazem Hajj for his unwavering support and priceless guidance throughout my time at the American University of Beirut. I would also like to thank Dr. Sirine Taleb for her contribution in the proceeding research work that was the foundation of this thesis, and for the assistance she graciously provided and for being part of the thesis committee. Moreover, I would like to thank Prof. Zaher Dawy and Prof. Rabih Jabr for taking the time to be part of my thesis committee and for their helpful feedback at the time of the thesis proposal. Last but not least, I want to thank my family and colleagues for their persistent and strong support and encouragement all throughout my academic pursuits.

An Abstract of the Thesis of

Raslan Hussein Kain for Master of Engineering
Major: Electrical and Computer Engineering

Title: Holistic Approach to Energy Efficiency in Context-Aware Mobile Sensing for Multi-Context Recognition

Mobile devices and sensors have limited battery lifespan thus limiting their feasibility in context recognition applications. As a result, there is a need to provide mechanisms for energy-efficient operation of sensors in settings where multiple contexts are monitored simultaneously. Past methods for efficient sensing operation have been hierarchical by first selecting the sensors with least energy consumption, then devising individual sensing schedules that trade off energy and delays. The main limitation of the hierarchical approach is that it does not consider the combined impact of sensor scheduling and sensor selection. This paper aims at addressing this limitation by considering the problem holistically and devising an optimization formulation that can simultaneously select the group of sensors while also considering the impact of their triggering schedule. The optimization solution is framed as a Viterbi algorithm and providing mathematical representations for multi-sensor reward function, the user's behavior. Experiment results showed an average improvement of 60% compared to the state of the art hierarchical approach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

advances in ubiquitous computing, such as smart wearables and sensor networks, have provided opportunities for health monitoring and human-centered context-aware systems. Mobile devices with sensory capabilities are commonly used to recognize the contexts of the user and provide appropriate assistance and services. Context relates to numerous areas of human-centric activities, such as health-care monitoring [2] [3], activity recognition [4], social networking [5], location [6] [7], and emotion recognition [8] [9]. A context state describes one of many possible depictions of an entity within a context. States from separate contexts are mutually disjoint, for example in the context activity, the user can be walking, sitting, or running. On the other hand, in the context of emotion, the states can be happy, sad, angry. Context states are typically recognized by processing data collected from smartphone sensors (accelerometer, gyroscopes, GPS, etc.), wearable sensors (electrocardiograms (ECG), heart rate sensors, body temperature sensors), or wearable devices (smartwatches and headsets).

A key issue with context aware applications is the large demand on battery energy attributed to sensors' power consumption [10], and algorithms that increase computational workload [11]. To minimize delays in context recognition , sensors would need to be triggered continuously. However, it would be more energy efficient to use the sensors intermittently, by turning them off when a state remains unchanged and then back on when a new state is expected to be encountered. Unfortunately, the time of state change cannot be perfectly predicted beforehand, since anticipation of state changes is not deterministic and is still a significant challenge in the field of anticipatory mobile computing [12]. To achieve energy efficiency, a sensing schedule, should be devised to minimize delays in contextual change but also minimize the energy consumption required by frequent sensor triggers.

In addition to efficiency in timely triggers, there is an opportunity for synergy across choices of sensors when multiple contexts are desired and there are multiple sensing options for the same context. In such cases, the ideal choice of sensors is one that minimize the energy consumption across all sensors being used for all contexts, not just for each context separately. There has been active research to devise efficient sensing mechanisms for context recognition. Some methods have focused on individual sensors or single context [13, 14]. While more recent research advances [1, 15] can trade-off energy and delays for multiple groups of sensors and multiple contexts. One promising approach proposed by Taleb et al. [15] showed a hierarchical approach by first selecting groups of sensors then determining individual sensing schedules. While their work has achieved significant improvements compared to prior work, the main limitation of the work is its inability to account for the combined impact of for schedule decisions and sensor selection.

The aim of this work is to overcome this limitation by proposing a holistic optimization approach that can simultaneously consider sensor selection and sensing schedules towards optimal trade-off between overall energy consumption and delays in detecting desired contexts. The solution is framed as a Vietrbi algorithm with personalized capture of user behavior that reflects most probable time instances for change in context. The contributions of this work are:

- A holistic optimization formulation for simultaneous decisions on sensor choices and sensor schedules. The optimization takes into account personalized user behavior that reflects times spent in different context states.

- Viterbi based solution for the optimization with mathematical formulation for 1) multi-sensor reward function accounting for energy in groups of sensors and delays in detecting context change 2) user behavioral model that includes the probability of context state change based on historical user behavior 3) state transition options with corresponding rewards.

Chapter 2

Literature Review

The amount of research done in energy efficient mobile sensing has increased in recent years, with the advent of Body Sensor Networks (BSN) that has facilitated the development of context aware applications for purposes of health monitoring, activity recognition, location tracking, etc. The in-depth surveys [16] [17] [18] detail some of the work done in the area of energy efficient mobile sensing, and their content is used in our proposed solution for information about sensor specifications and different possibilities of sensor groups for context recognition.

Perez-Torres et al. in [16] present a comprehensive study of power-aware smartphone-based strategies for context recognition, focusing on hardware and software based strategies, mostly for location based applications. Our work falls under the category of software based strategies for multiple contexts. Rault et al. in [17] present a survey existing energy-efficient approaches, from sensor selection, power on time reduction, communication reduction, and computation reduction for health care applications. Yurur et al. in [18] survey the literature of context-aware applications for mobile platforms over a period of 13 years. Previous work on efficient sensors' operations for context recognition can be categorized into work that has considered sensor selection only, sensor scheduling, or both.

2.1 Sensor Selection

A context may be recognized by different combinations of sensor with each combination having a different impact on energy consumption and accuracy in context recognition. Some works of research aimed to select sensors to reduce the amount of energy consumption while considering the impact on the accuracy in context recognition for the sensor choices. Such as in the paper by Taleb et al. [19], which presents and algorithm that uses a heuristic for the selection mechanism. The heuristic consists of choosing the sensors that maximize a ratio of the level of context recognition accuracy divided by the energy consumption of the sensors with constrains based on sensor availability, and battery level. Gao et al. in [20] propose a framework that selects a set of sensors to reduce energy consumption attributed to wireless data transmission in a two step process, First a Naive Bayes classifier classifies the state using a designated initial sensor set and attains a priori probability. Then expert knowledge is used to select a subset of the initial sensors based on the state. Convex optimization is used to minimize a trade-off of transmission energy and the probability context miss-recognition. A framework presented by Kang et al [21], called SeeMon, selects a set of sensors named the Essential Sensor Set (ESS) by solving a variation of Minimum Set Cover problem solved by a heuristic algorithm which greedily selects the most cost-effective sensors iteratively capable of recognizing the context while trading-off computational complexity and energy savings in terms of data transmission rate. The ESS gets updated either continuously or periodically based the available battery levels.

Dynamic Sensor Selection Activity Recognition is a method presented by Zappi et al. [22] which trades-off power consumption and activity recognition accuracy by adapting the set of sensors once the energy of sensor nodes (i.e. smart-phone, smart-watch, etc.) get depleted. The approach either selects the sensor cluster that gives the highest accuracy after enumerating all the possibilities, or by selecting the sensor cluster that first meets an accuracy threshold during enumeration. Another approach by Gordon et al. [23], selects sensors based on the predicted future activity state of the user using a first-order Markov Chain. The approach evaluates a weighted mapping of each activity to the sensors in terms of the loss in accuracy compared to other sensors. The weighted map is generated by applying nearest neighbor classifier against all training vectors for each state and simulating different feature combinations (each feature combination is linked to a sensor group), and seeing the varying effect on accuracy of different combinations.

2.2 Sensor Scheduling

The continual operation of sensors drains the battery supply of mobile devices quickly. So the challenge is to derive a schedule that minimizes energy consumption while avoiding delays in detecting changes in context state. Taleb et al. [1] presented a dynamic means of generating a sensing schedule using a Viterbi based algorithm, called Viterbi based Context Aware Mobile Sensing or VCAMS. VCAMS finds the sequence of sensor triggering decisions that maximizes a cumulative reward function, called the Viterbi path, in the context of hidden Markov models. The reward function accounts for energy consumption of the sensor used for context recognition, with a constant value, and a delay component composed of a weighted sum of delay penalty and a recognition reward. Rachuri et al. [14], proposed an adaptive sensor triggering method based on the use of a feedback mechanism that reduces or increases sensor inactivity time by a multiplicative function based on the current classification of the context state. The current state was classified using a Gaussian Mixture Models classifier (GMM) as either missable or unmissable. A missable event corresponds to no change in the state or state change that was not of interest leading to a increase in the period sensor inactivity, while an unmissable event corresponded to a state change leading to decrease in the period. In a follow-up paper by Rachuri et al. [24], a framework called sociable sense was created that captures user behavior in office environments and uses the previous method to schedule sensor operation. However, a linear reward-inaction algorithm was used to predict missable and unmissable events in place of the Gaussian Mixture Models classifier (GMM) and adjust the sensor inactivity time accordingly. Yurur et al. [25] present a Hidden Markov Model (HMM) based framework to recognize activity with a generalized expectation maximization algorithm to achieve a trade-off between energy consumption and accuracy. The state transition probabilities of the HMM are updated, i.e. sensors are triggered to fill a sliding window frame with sensor observations to recognize user states using a Viterbi algorithm, when an entropy rate of the user state transition matrix converges into a stable value.

There are several works of research that utilize sensors that consume little amounts energy for activity recognition, such as an accelerometer and gyroscope, to schedule the operation of sensors that consume a lot of energy for location recognition, such as GPS and GSM. SenseLess by Abdesslem et al. [26] uses binary simple inference performed on accelerometer data accessed every 10 seconds. Similarly in the Jigsaw system by Lu et al. [27], recognizes activity using accelerometer and microphone data, in addition to the expected running time and energy budget, in a Markov Decision Process to schedule GPS sensor operation for location recognition. SmartDC by Chon et al. [28] uses and accelerometer to recognize historical pattern of movement to infer the location, If the location could not be recognized, a sensing schedule was generated, based on maximizing a reward function that minimize an energy cost partitioned into levels based on an accuracy of location recognition threshold.

2.3 Sensor Selection and Scheduling

A few works of research have aimed to combine both techniques in one framework, however they all follow a hierarchical structure. Wang et al. [13] present a hierarchical sensor management system called Energy Efficient Mobile Sensing System (EEMSS) that first selects the sensors and then schedules sensing while minimizing a trade-off between accuracy, delay, and energy consumption. The framework consists of a sensor management scheme that manually links the users states with specific sensors by deciding which sensors to activate based on the current recognized state. When a state transition gets detected, the next set of sensors in the sequence is activated. Also, a sensing schedule, called sensor duty cycle in the paper, is manually generated through empirical tests, having energy accuracy and latency trade-off in mind. This is cumbersome to do and does not allow for adaptability and finding the Pareto optimal trade-off similar to other methods have done by automating the process. Orchestrator presented by Lee et al. [29], is a resource coordination system that satisfies the resource demands of the multiple applications, system-wide policies, and meeting resource availability of devices. A processing planner generates multiple plans which specify the combinations of sensors with the associated accuracy in context recognition. These plans are pre-defined by developers specifying requirements on sensors, and then only some of them are selected based on whether they support the maximal context recognition requests, minimize energy consumption, and maximize recognition accuracy with available resources. Moreover, sensor triggering and data transmission are done periodically at fixed time intervals, according to the estimated energy level availability and accuracy requirement.

More recently, Taleb et al [15] considered the decision of both sensor selection and scheduling. The method, dubbed EGO, relies on an ontology that contains specifications such as sensors and machine learning parameters. The approach filtered out combinations of sensors groups that don't meet manually set accuracy and energy budget constraints. The group with the minimum energy consumption per trigger was selected. The framework uses a Viterbi based algorithm presented in VCAMS [1] to generate the sensing schedules of the selected sensors as described in section 2.2. The sensing schedule based on a user state behavioral model predicts when the user may change state. The last step in the algorithm is to synchronize the schedules of multiple sensors common across different context recognition requests.

EGO relies on a hierarchical structure, it first selects the groups of sensors, then schedules their operation, and finally synchronizes the schedules of the common sensors. Such an approach doesn't take into account how the operation of the chosen sensor groups according to a sensing schedule will impact energy consumption and delay in state change detection. Our work proposes a holistic approach that takes into account the energy and delay costs of operating sensors according to optimized sensing schedules, accounting for the impact of accuracy in context recognition on delay, in selecting the sensor groups to recognize multiple contexts simultaneously.

Chapter 3

Problem Description and Optimization Formulation

Given the current state of the user, the goal is to decide on the sensors and the timing of their triggers, that form a sensing schedule, to recognize multiple target contexts simultaneously while minimizing the trade-off between total energy consumption by all sensors selected and the delays incurred in context recognition. The problem is illustrated in figure 3.1.

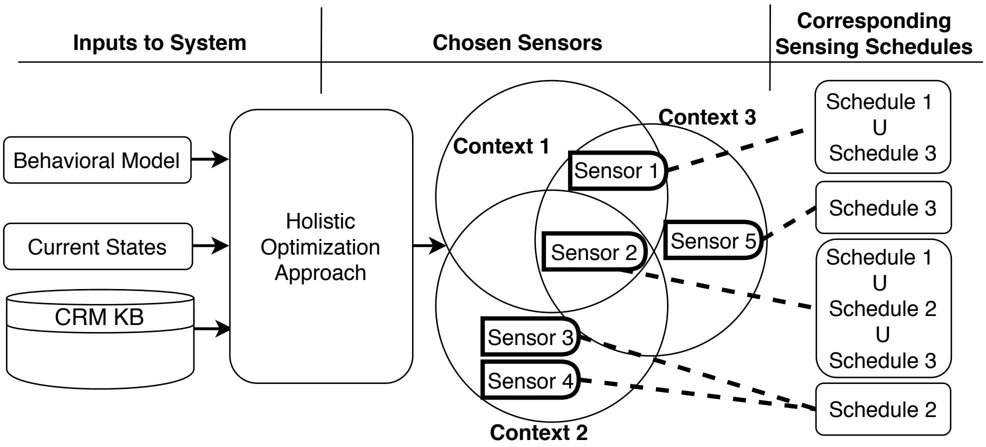


Figure 3.1: Illustration of the system inputs and outputs. The inputs are the user's current context states, behavior model, and the CRM KB. The outputs are the selected sensors and their respective sensing schedules. The selected sensors are either used in recognizing a single context with a sensing schedule corresponding to the group, or else used in recognizing multiple context, and have a synchronized sensing schedule obtained by the union of the schedules corresponding to the different sensor groups.

The holistic optimization takes as input a Context Recognition Models Knowledge Base (CRM KB), the history of past user behavior (behavior model), and the current context state of the user of each desired context. When a selected sensor is common between multiple groups recognizing different contexts, its sensing schedule must serve the recognition operation of each context. Thus, the sensing schedules of the different sensor groups, that the common sensor is part of, require synchronization to be assigned to the sensor, by taking the union of the schedules. The sensors that are unique to a sensor group are designated the sensing schedule of that particular sensor group. The sensors common to the multiple sensor groups are designated the synchronized sensing schedule of the sensor groups. We assume that, for a particular context state, sensing switches to continuous once the time interval exceeds the length of the sensing schedule. During continuous sensing, there would be no incurred delays as the sensors are always on.

Each context is denoted c_l with $l = 1, \ldots, L$, where L is the total number of context that can be recognized by the system. For each context c_l the states are denoted by x_l^j, where $j = 1, \ldots, J_l$, and J_l is the number of states for context c_l. For example, the activity context can be in of the following states: walking, sitting, working; The location context can be at home, work, or cafe; The emotion context can have the following states: happy, sad, and neutral. Each context c_l requires sensory data accessible by the mobile and wearable devices handling the computations required for the recognition. Hereafter, when l appears as an index for a term it means that that term is related to context c_l.

3.1 Knowledge Base for Context Recognition Models

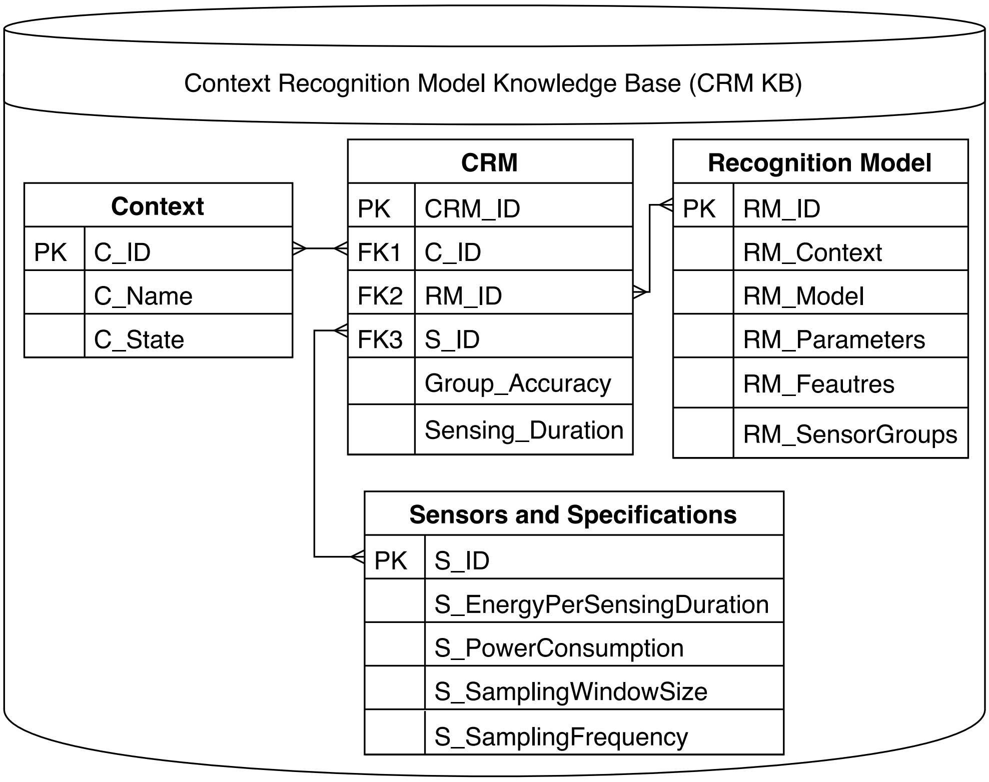
Our solution makes use of the wealth of past knowledge in the field of context recognition. In particular, we assume the availability of a knowledge base that contains information about context recognition models, sensors they can use, and accuracy that can be achieved along with sensors' specifications. The knowledge base contains specifications about ML models for context recognition including specifications for required sensors. Such information can be easily extracted from credible research publications, like the one used in EGO [15].

The Context Recognition Model Knowledge Base (CRM KB) is illustrated in figure 3.2 and contains:

- Context table containing the primary key, name of contexts that may be recognized (C_Name), and the different possible context states (C_State).
- Recognition model table containing the primary key, the context recognized my the model (RM_Context), the Machine learning model or algorithm used to recognize the context (RM_Model), the parameters of the model (RM_Parameters), the features required by the model (RM_Features), and the needed group of sensors (RM_SensorGroups).

- Sensors and Specifications table containing the primary key and for each sensor the energy consumed per sensing duration (S_EnergyPerSensingDuration), the power consumption value (S_PowerConsumption), the recommended sampling frequency (S_SamplingFrequency), and the sampling window size (S_SamplingWindowSize).

- Context recognition model relational table, containing the primary key (CRM_ID), and the foreign key of the three entity tables found in the CRM KB. For the context table (C_ID), the recognition model table (RM_ID), and the sensors and specifications table (S_ID). In addition to the senor group context recognition accuracy and the sensing duration, which depend on the combination of information from all the entity tables.



Context Recognition Model Knowledge Base (CRM KB)

Context	
PK	C_ID
	C_Name
	C_State

CRM	
PK	CRM_ID
FK1	C_ID
FK2	RM_ID
FK3	S_ID
	Group_Accuracy
	Sensing_Duration

Recognition Model	
PK	RM_ID
	RM_Context
	RM_Model
	RM_Parameters
	RM_Feautres
	RM_SensorGroups

Sensors and Specifications	
PK	S_ID
	S_EnergyPerSensingDuration
	S_PowerConsumption
	S_SamplingWindowSize
	S_SamplingFrequency

Figure 3.2: The Context Recognition Model Knowledge Base (CRM KB), containing the information relevant to context recognition: 1) Context 2) Sensors and Specifications 3) Recognition Model and 4) the associations of the three together to recognize a context..

The information, needed for sensor decisions, can extracted by querying the CRM KB. The following is a list of information extracted for our solution along with the notations:

- The set of possible sensor groups for each context is denoted by π_l.

- Each sensor group within π_l is represented by $\mathcal{G}_n^l$ where n represents one of the possible groups of sensors to recognize c_l. N_l is the total number of groups in π_l, i.e. that $n = 1, \ldots, N_l$.

- Each sensor is denoted by S^m, where m indicates the specific sensor. Assuming the availability of M embedded and wearable sensors, $m = 1, \ldots, M$. A sensor can belong to different sensor groups. The sensors that make up a group are typically available in mobile and wearable devices, such as a smartphone and smartwatch. Example sensors include: GPS, accelerometer, gyroscope ...etc.

- The energy consumption is denoted by $E_{\mathcal{G}_n^l}$ for each sensor group $\mathcal{G}_n^l$. This energy is composed of the energy spent turning the sensor on and off, CPU, data transmission, as in the case of a wearable sensor.

- The accuracy achieved by each sensor group G_n^l in detecting contextual state x_l^j is denoted $A_{\mathcal{G}_n^l}^j$.

- The time duration needed to recognize a state x_l^j is denoted $\delta_{\mathcal{G}_n^l}$. $\delta_{\mathcal{G}_n^l}$ includes the time to turn the sensors on and collect enough sensory data to recognize a context state. For example, the time to recognize a location state with a GPS, according to the process described in [30], is 20 seconds. While the time required to recognize an activity state with an accelerometer, according to the process described in [31], is 5 seconds.

- The specifications for the ML model needed to recognize a particular context including:
  - The possible states x_l^j in a given context c_l.
  - Machine learning models that would be used for a given choice of sensors.
  - Data features collected for each machine learning model.

3.2 User Time Spent in each Context State

Our solution also makes use of a behavioral model that provides opportunity for personalizing the sensing operation to particular users and particular states. Previous approach [1], to derive user behavior, assumes one pattern of behavior for each state and derives a single time limit for the user in those state. We propose an alternative method that takes a more realistic assumption of user's behavior by accounting for variation of user's behavioral pattern within each

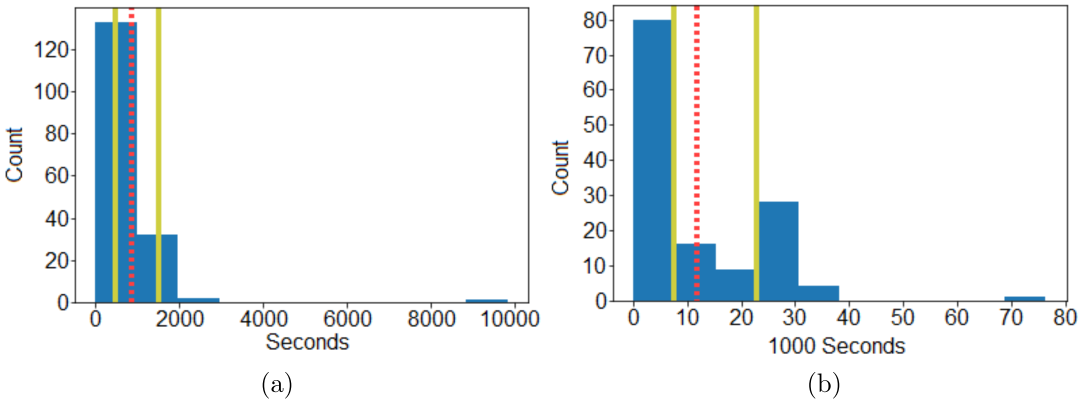


Figure 3.3: Distribution of time durations spent in Walk (a) and Sleep (b) states in an activity context. The red dashed line is the result of previous approach [1] with one time limit, while the yellow lines are obtained using our method of modeling the behavior using multiple time limits.

state. As a result, a user can have, instead of one time limit, multiple time limits within each state. Our proposed method uses a frequent pattern mining approach, where we capture the most frequent times spent in each state. The method requires two parameters to be specified: 1) The size of the histogram bins reflecting the desired granularity for the patterns of time limits, and 2) The threshold of counts within a bin to consider the pattern frequent enough. These two parameters can be set based on the desired time resolution and what the system designer deems as frequent. For the purpose of our experimentation, A histogram of the different values with bin sizes equal to 10% of the longest time duration recorded for the state. The threshold of counts was selected to be 5% of the time durations recorded for the state. The time limits are computed by taking the middle time duration of the bins that exceed the threshold of counts. Additionally, if consecutive bins are not individually highly populated, multiple bins are combined to form a frequent pattern. The middle time duration of the consecutive bins is taken as a time limit.

Figure 3.3 shows time durations spent in the two activity states, ”Walk” and ”Sleep”. The yellow lines indicate the time limits generated by our method. In figure 3.3(b) there are four highly populated bins (the first four), that get captured by the two time limits represented by the yellow lines. The red dashed linein both figures indicates the time limit obtained by averaging all the time durations. The behavioral model can be updated over time by tracking the user’s time spent in each state. When the count of a bin surpasses the 5% threshold, the list of time patterns, or time limits, for each state is updated. The resulting time limits for each state x_l^j, are denoted by $T_{l,h}^j$, where the h index represents the time limit in ascending order, where $h = 1, \dots, H$, H is the total number of time limits for a state and $T^j_{l,H}$ is the time limit with the longest duration.

The time limits, $T^j_{l,h}$, are used to generate a sensing schedule, denoted by $a^i_{\mathcal{G}^l_n}$, for a sensor group $\mathcal{G}^l_n$, where a sensing decision is taken every $\delta_{\mathcal{G}^l_n}$ seconds, and the each particular time instance is referred to as $t^i_{\mathcal{G}^l_n}$, where $i = 1, \ldots, I_{\mathcal{G}^l_n}$, and $I_{\mathcal{G}^l_n}$ is the last decision instance before sensing becomes continuous and is computed as shown in 3.1.

$$I_{\mathcal{G}^l_n} = \frac{T^j_{l,H}}{\delta_{\mathcal{G}^l_n}} \tag{3.1}$$

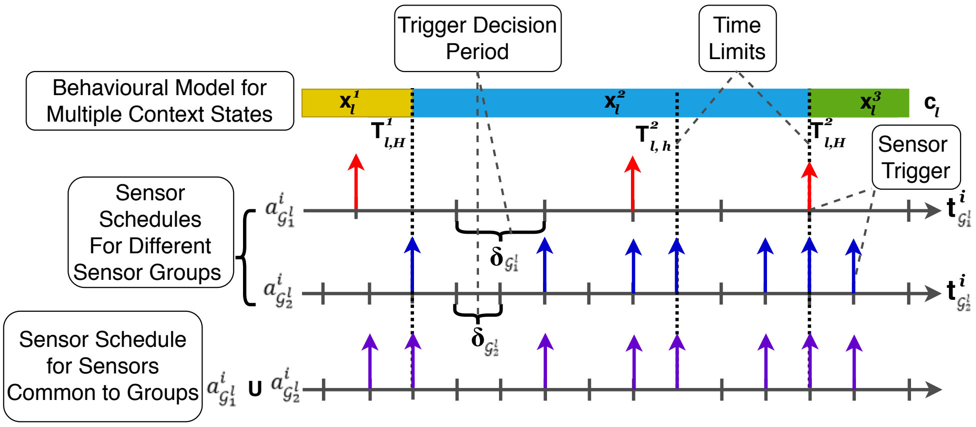

Trigger Decision Period
Time Limits
Behavioural Model for Multiple Context States
x^1_l
x^2_l
x^3_l
c_l
$T^1_{l,H}$
$T^2_{l,h}$
$T^2_{l,H}$
Sensor Trigger
Sensor Schedules For Different Sensor Groups
$a^i_{\mathcal{G}^l_1}$
$t^i_{\mathcal{G}^l_1}$
$\delta_{\mathcal{G}^l_1}$
$a^i_{\mathcal{G}^l_2}$
$t^i_{\mathcal{G}^l_2}$
$\delta_{\mathcal{G}^l_2}$
Sensor Schedule for Sensors Common to Groups
$a^i_{\mathcal{G}^l_1} \cup a^i_{\mathcal{G}^l_2}$

Figure 3.4: The segmentation of time schedule for 2 sensor groups capable of detecting the same state, with each group having a different triggering decision interval. In addition to a synchronized schedule for common sensors between the 2 sensor groups.

Figure 3.4 illustrates an example of two sensing schedules ($a^i_{\mathcal{G}^l_1}$ and $a^i_{\mathcal{G}^l_2}$) for sensor groups ($\mathcal{G}^l_1$ and $\mathcal{G}^l_2$) in recognizing context c_l. $a^i_{\mathcal{G}^l_1}$ has a triggering decision period of $\delta_{\mathcal{G}^l_1}$ and $a^i_{\mathcal{G}^l_2}$ has a period of $\delta_{\mathcal{G}^l_2}$. The user behavior model captures the historical patterns of user's behavior in different context states. These patterns are represented by time limits, denoted by $T^j_{l,h}$, that indicate when there is a likelihood of change in a context state. On the top of figure 3.4 the behavior model is shown for each context state x^j_l with the corresponding time limit $T^j_{l,h}$. The time limits influence the triggering decisions, as can be seen in figure 3.4, showing that sensing decisions for $a^i_{\mathcal{G}^l_1}$ and $a^i_{\mathcal{G}^l_2}$. Moreover, if sensor groups $\mathcal{G}^l_1$ and $\mathcal{G}^l_2$ have common sensors, then their schedule is $a^i_{\mathcal{G}^l_1} \cup a^i_{\mathcal{G}^l_2}$ which is the synchronization of the two sensing schedules $a^i_{\mathcal{G}^l_1}$ and $a^i_{\mathcal{G}^l_2}$, as illustrated as the bottom sensing schedule in figure 3.4.

3.3 Mathematical Formulation

Mathematically, the optimization problem can be formulated as a weighted sum of two objectives, minimizing energy consumption and delay based on scalarization [32]. The potential sets of sensors are formed by the union of the selected groups of sensors to recognize each context c_l, represented as $\mathbf{U}_l\mathcal{G}^l_n$. The specific choices of sensor groups and their schedules are obtained by aiming to achieve an optimal trade-off between energy and delay. The formulation minimizes the combination of energy consumption of the union of sensor groups $E^j_{\mathbf{U}_l\mathcal{G}^l_n}$ while recognizing context state x^j_l and cumulative delays for all contexts in contextual state detection $D^j_{U\mathcal{G}^l_n}$ according to the assigned sensing schedule $a^i_{\mathcal{G}^l_n}$, multiplied by the boolean decision variable $y_{\mathcal{G}^l_n}$ for the sensor group considered in the computation, normalized by the maximum possible values of energy and delay.

$$\min_{y_{\mathcal{G}^l_n},\, a^i_{\mathcal{S}m}} \sum_{\mathcal{G}^l_n} \left(\omega_t \frac{E^j_{U\mathcal{G}^l_n}}{E_{U\mathcal{G}^l_n,max}} + (1-\omega_l)\, \frac{D^j_{U\mathcal{G}^l_n}}{D^j_{l,max}} \right) \left(\prod_l y_{\mathcal{G}^l_n} \right) \tag{3.2}$$

with the following constraint:

$$\sum_n y_{\mathcal{G}^l_n} = 1, \quad \forall\, c_l \tag{3.3}$$

- $E^j_{\mathbf{U}_l\mathcal{G}^l_n}$ is the total energy consumption resulting from triggering the union of sensors $\mathbf{U}_l\mathcal{G}^l_n$. $E^j_{\mathbf{U}_l\mathcal{G}^l_n}$ is computed as the sum of sensor triggers multiplied by the energy consumption per trigger $E_{\mathbf{U}_l\mathcal{G}^l_n}$ of the sensor groups recognizing each context state according to the respective sensing schedules. To account for uncharacteristic changes in user behavior, i.e. changing state before or after a time limit, the average of the energy expenditure for multiple scenarios of multi-context recognition is used to obtain the final result. Each scenario ends when the user changes their state, and thus the number of triggers prior to the context state change is counted.

- $D^j_{U\mathcal{G}^l_n}$ represents the delays incurred in using the selected sensors for context recognition of the different states x^j_l. $D^j_{U\mathcal{G}^l_n}$ is computed as the difference in time between when the state has changed and the sensor trigger following the change, and averaged over multiple scenarios of multi-context recognition.

- $y_{\mathcal{G}^l_n}$ is a boolean decision variable denoting the selection of sensor group $\mathcal{G}^l_n$. $y_{\mathcal{G}^l_n}$ is 1 when the groups of sensors is being considered in the computations, otherwise $y_{\mathcal{G}^l_n}$ is 0 when the sensor groups energy is not included in the minimization term.

- $a^i_{S^m}$ is a vector of boolean decision variables denoting the trigger schedule for sensor $S^m \in U\mathcal{G}^l_n$ at different time instances $t^i_{\mathcal{G}^l_n}$ as follows:

$$a^i_{S^m} = \begin{cases} 1, & \text{when sensor } S^m \text{ is triggered at } t^i_{\mathcal{G}_n} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

- ω_l is a weighting factor that provides a balance between the two objectives, and is user-specified depending on the application.. For example in health context delay in state change might be more important than the energy consumption, thus a smaller value of ω_l.

- $E^j_{U_l\mathcal{G}^l_n,max}$ in the maximum energy consumption value, which is equal to the last time limit $T^j_{l,H}$ multiplied by $\Omega_{\mathbf{U}_l\mathcal{G}^l_n}$, where $\Omega_{\mathbf{U}_l\mathcal{G}^l_n}$ is the power consumption value of the union of sensor groups $\mathbf{U}_l\mathcal{G}^l_n$ when the sensors are operating continuously.

$$E^j_{U_l\mathcal{G}^l_n,max} = \Omega_{\mathbf{U}_l\mathcal{G}^l_n} \times T^j_{l,H} \tag{3.5}$$

- $D^j_{l,max}$ is the the maximum delay value, which is equivalent to $T^j_{l,H}$ since it represents the case when the generated sensing schedule does not decide to sense until $t^I_{\mathcal{G}^l_n}$, or when $t^i_{\mathcal{G}^l_n} = T^j_{l,H}$,

$$D^j_{l,\max} = T^j_{l,H} \tag{3.6}$$

Equation 3.2 exploits the synergy between the selected groups of sensors by considering the total energy cost consumed by the union of sensors $\mathbf{U}_l\mathcal{G}^l_n$ of the selected groups to recognize the different contexts. Constraint 3.3 states that exactly one group of sensors for each context is to be chosen. The objective function 3.2 takes the form of a mixed-integer linear programming (MILP) problem, which is NP-Hard [33], where the one variable is a boolean variable for the choice of sensor groups and the other is a vector of boolean variables of varying size representing the sensing schedule of the selected sensor. The solution is guaranteed to converge to a global minima because of the convexity of linear problem [34]. The optimization problem is solved by choosing sensors to recognize the requested context with the respective sequence of sensing decisions for each sensor that minimize the objective function 3.2.

The table 3.1 on the next page summarizes the notations used in the formulation.

Table 3.1: List of notations

Groups	Notations	Descriptions
Sets	L	Set of contexts
	M	Set of available sensors S^m
	N_l	Set of sensor groups capable of recognizing context l
	J_l	Set of total states in context c_l
	H	Set of time limits in state x_l^j
	$I_{\mathcal{G}_n^l}$	Number of sensing decision instances in the sensing schedule of sensor group $\mathcal{G}_n^l$ before continuous sensing is applied
	π_l	Set of sensor groups
Parameters	c_l	Particular context out of L total requested contexts
	x_l^j	Particular state in context c_l
	$\mathcal{G}_n^l$	Sensor groups that can detect context c_l, where n corresponds to each sensor group, such that, $n = 1, \ldots, N_l$
	$\Omega_{\mathbf{U}_l \mathcal{G}_n^l}$	Power consumption value for continuous sensing operation of sensor group $\mathcal{G}_n^l$
	S^m	Sensor belonging to sensor group $\mathcal{G}_n^l$
Optimization values	ω_l	Where $0 < \omega_l < 1$is the weighting factor
	E_m	Energy consumption by each sensor S^m
	$E_{\mathcal{G}_n^l}^j$	Estimated energy consumed by group $\mathcal{G}_n^l$ with sensor schedule $a_{\mathcal{G}_n^l}^i$
	A_m^j	Accuracy of sensor $S_{\mathcal{G}_n^l}^m$ in recognizing state x_l^j
	$A_{\mathcal{G}_n^l}^j$	Accuracy that group $\mathcal{G}_n^l$ can guarantee for state x_l^j
	$D_{\mathcal{G}_n^l}^j$	Delay in recognizing state x_l^j using sensor group $\mathcal{G}_n^l$
Variables	$t_{\mathcal{G}_n^l}^i$	Time instants related to the sensor group $\mathcal{G}_n^l$
	$T_{l,h}^j$	Most frequent time durations the user spends in state x_l^j, where h is the number of time limits such that $h = 1, \ldots, H$ and $T_{l,H}^j$ is the last time limit
	$p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)$	State Survival probability representing the likelihood of remaining in state x_l^j at any time $t_{\mathcal{G}_n^l}^i$ as a function of $T_{l,h}^j$
	ε	Near zero value that is reached by the survival probability $p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)$ at $t_{\mathcal{G}_n^l}^i = T_{l,h}^j$
	η	Factor of the time limits $T_{l,h}^j$ where the survival probability $p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)$ represented by the Uniform function drops to ε
Decision Variables	$y_{\mathcal{G}_n^l}$	Binary variable representing selection of group $\mathcal{G}_n^l$
	$a_{\mathcal{G}_n^l}^i$	Vector representing sensor triggering for group $\mathcal{G}_n^l$ actions in context: 1 for sense and 0 for not sense

Chapter 4

Overview of Proposed System

The solution to the optimization problem needs to be used in real-time by context recognition systems to decide which groups of sensors and when to trigger them to consume the least energy and achieve minimal delay in multi-context recognition. To achieve real-time performance, we propose to pre-compute optimized sensing schedules and recognition delays for different possible combinations of sensor groups and context states, and make decisions among these choices online. As a result, the approach is split into two stages, an online stage and an offline stage as illustrated in figure 4.1.

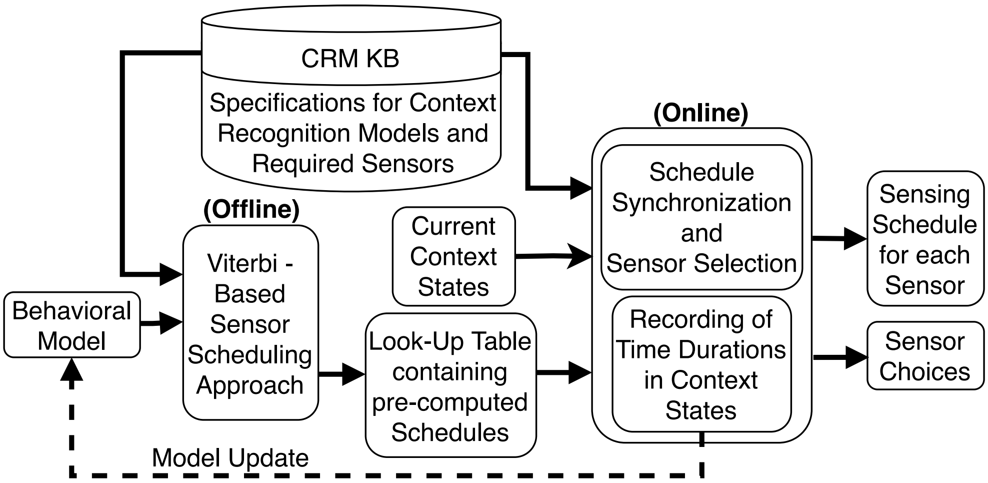

CRM KB
Specifications for Context Recognition Models and Required Sensors
(Offline)
Viterbi - Based Sensor Scheduling Approach
Behavioral Model
Current Context States
Look-Up Table containing pre-computed Schedules
(Online)
Schedule Synchronization and Sensor Selection
Recording of Time Durations in Context States
Sensing Schedule for each Sensor
Sensor Choices
Model Update

Figure 4.1: The holistic optimization approach split into the two stages, the online stage and the offline stage. The offline stage provides the best sensing schedule for a particular context and a group of sensors. The online stage finds the best synergy between sensor options for the simultaneous recognition of multiple context, and determines the best combinations of sensors and their sensing schedules.

The offline stage addresses the problem of determining the optimized sensing schedule that provides the best trade-off between energy consumption and delay for each particular context. These combinations are then stored in a look-up-table (LUT) that is used online. The online stage addresses the problem of finding the groups of sensors and their schedule to achieve best synergy between sensor options for the simultaneous recognition of multiple contexts. The online system provides multi-context trade off between energy and delay.

4.1 Online System: Sensor Selection and Schedule Syncronization

The proposed solution is based on the Viterbi algorithm, where the goal is to make the sensing and sensor selection decisions that maximize a holistic reward function. The details are further described below.

Algorithm 1 details the procedure for selecting the sensors. The online system runs every time a context state changes. The approach starts by computing combinations of sensor groups $\mathbf{U}\mathcal{G}_n^l$ from the set of sensor groups π_l capable of recognizing all the desired contexts. For each possible sensor group combination combination $\mathbf{U}_l\mathcal{G}_n^l$ we obtain the sensing schedules from the LUT generated using the Viterbi based algorithm for each sensor group $\mathcal{G}_n^l$ and synchronize the sensing schedules for sensors common to multiple sensor groups. The energy consumption and delay attributed to the union of the sensor groups being considered are evaluated to account of the synchronized schedules of the common sensor groups. The energy value attributed to the sensor groups is re-computed without the energy consumption per trigger of the common sensors E^m. While for the common sensors, the energy consumed according to the synchronized schedule is computed and added to the total energy value of the combination of sensor groups.

For each unique sensor S^m in $\mathcal{G}_n^l$ we assign it $a^i_{\mathcal{G}_n^l}$ as $a^i_{S^m}$, and for the common sensors, the corresponding synchronized schedule, which is the union of the schedules, and we assign it as $a^i_{S^m}$. The energy and delay values are used to calculate objective function values according to equation 3.2. Finally the combination of sensor groups $\mathbf{U}_l\mathcal{G}_n^l$ based on $y_{\mathcal{G}_n^l}$ and corresponding sensing schedule $a^i_{S^m}$ for each sensor S^m giving the minimum objective function value are selected. Each time a new context state is detected, the time spent in previous context state is recorded and added to the historical record, which is used to update the behavior model, to account for any changes in used behavior.

To illustrate how we calculate the objective function for a given group of sensors with their sensing schedule using the data, shown in table 4.1, for three hypothetical scenarios of recognizing an activity state "Walk". The context state is recognized using an accelerometer sensor $\mathcal{G}_2^1$, which consumes $E_{\mathcal{G}_2^1} = 0.3$ mJ of energy per trigger for $\delta = 10$ seconds. Three scenarios are represented capturing

Algorithm 1 Overview of Online Algorithm for Sensor Selection and Sensing Schedules

Input: x_l^j, π_l, S^m, E_m, $E^j_{\mathcal{G}^l_n}$, $D^j_{\mathcal{G}^l_n}$ $a^i_{\mathcal{G}^l_n}$

x_l^j : the current context state of the user contexts $\forall\, l = 1, 2, \ldots, L$

From CRM KB:

π_l: the set of possible sensor groups $\mathcal{G}^l_n$ for each of the requested L contexts

S^m: the available sensors

E_m: the energy consumption by each sensor S^m

From LUT:

$E^j_{\mathcal{G}^l_n}$: the pre-computed energy consumption for each group of sensors $\mathcal{G}^l_n$ in recognizing context state x_l^j

$D^j_{\mathcal{G}^l_n}$: the pre-computed delay of each sensor group $\mathcal{G}^l_n$ in recognizing context state x_l^j

$a^i_{\mathcal{G}^l_n}$: the pre-computed optimized sensing schedules for the sensor groups in π_l in recognizing each context state.

Output: $y_{\mathcal{G}^l_n}$, $a^i_{\mathcal{G}^l_n}$

$y_{\mathcal{G}^l_n}$: The boolean variable representing the selected sensor groups to recognize multiple contexts states x_l^j

$a^i_{S^m}$: Sensing schedules corresponding to the sensors of the selected group including the synchronized schedules

```
Derive all the possible union of sensor groups $\mathbf{U}_l\mathcal{G}^l_n$ available in $\pi_l$
for each $\mathbf{U}_l\mathcal{G}^l_n$ do
    for each $S^m \in \mathbf{U}_l\mathcal{G}^l_n$ do
        if $S^m \in \mathcal{G}^l_n$ then
            Assign $a^i_{\mathcal{G}^l_n}$ obtained from the LUT to $S^m$ of group $\mathcal{G}^l_n$ as $a^i_{S^m}$
        else if $S^m \in \mathcal{G}^l_w$ & $\mathcal{G}^l_q$ for $w \neq q$ then
            Synchronize sensing schedules $a^i_{\mathcal{G}^l_w}$ & $a^i_{\mathcal{G}^l_q}$ and assign to $S^m$ as $a^i_{S^m}$
        end if
    end for
    Calculate objective function values according to equation 3.2
    Track sensors $S^m \in \mathbf{U}_l\mathcal{G}^l_n$ and sensing schedules $a^i_{S^m}$ with minimum objective
        function value according to equation 3.2
end for
Return sensors $S^m \in \mathbf{U}_l\mathcal{G}^l_n$ with $y_{\mathcal{G}^l_n} = 1$ and the corresponding sensing
    schedules $a^i_{S^m}$ having the minimum objective value
if state $x_l^j$ changes to $\overline{x_l^j}$ then
    Record time spent in changed context state $x_l^j$
end if
```

Table 4.1: Three Scenarios in Walk State Detection with Time Limits $T^1_{1,1} = 3431$ and $T^1_{1,2} = 10274$, with unit of time in seconds.

Walk Activity Scenarios	Time of Actual State Change	Time of Nearest Sensor Trigger to State Change	Triggers During Scenarios
Scenario (1)	2000	1990, 2020	40
Scenario (2)	5000	4980, 5050	90
Scenario (3)	12000	-	180

the energy, delay, and objective function values are computed by taking the average values over the three scenarios and then normalize by the maximum possible values. We note that past the time instance equivalent to the time limit $T^1_{1,2} = 10274$ seconds, continuous sensing is applied. So, at time instance $t^i_{\mathcal{G}^1_2} = 12000$ seconds for the third scenario, which is past $T^1_{1,2}$ unlike the other scenarios where the state change happens before the last time limit, there is no delay from having the sensors turned off according to the sensing schedule.

$$\text{Normalized Energy} = \frac{E^j_{\mathcal{G}^l_n}}{E^j_{\mathcal{G}^l_n,\max}} = \frac{1}{3} \cdot \sum_{Scenario=1}^{3} \frac{E^1_{\mathcal{G}^1_2}}{T^1_{1,2} \cdot E_{\mathcal{G}^1_2}/\delta}$$

$$= \frac{1}{3} \cdot \sum_{1}^{3} \frac{\#\text{ of Triggers} \times E_{\mathcal{G}^1_2}}{T^1_{1,2} \cdot E_{\mathcal{G}^1_2}/\delta} = \frac{1}{3} \cdot \frac{40 \times 0.3 + 90 \times 0.3 + 180 \times 0.3}{10274 \times 0.3/10} \cong 0.1$$

$$\text{Normalized Delay} = \frac{D^j_{\mathcal{G}^l_n}}{D^j_{l,\max}} = \frac{1}{3} \cdot \sum_{Scenario=1}^{3} \frac{D^1_{\mathcal{G}^l_2}}{T^1_{1,2}}$$

$$= \frac{1}{3} \cdot \sum_{1}^{3} \frac{\text{Trigger After Change} - \text{Trigger Before Change}}{T^1_{1,2}}$$

$$= \frac{1}{3} \cdot \frac{2020 - 1990 + 5050 - 4980}{10274} \cong 0.01$$

$$\text{Objective Value} = \omega_l \frac{E^j_{\mathcal{G}^l_n}}{E^j_{\mathcal{G}^l_n,\max}} + (1 - \omega_l) \frac{D^j_{\mathcal{G}^l_n}}{D^j_{l,\max}} = 0.1 + 0.01 = 0.11$$

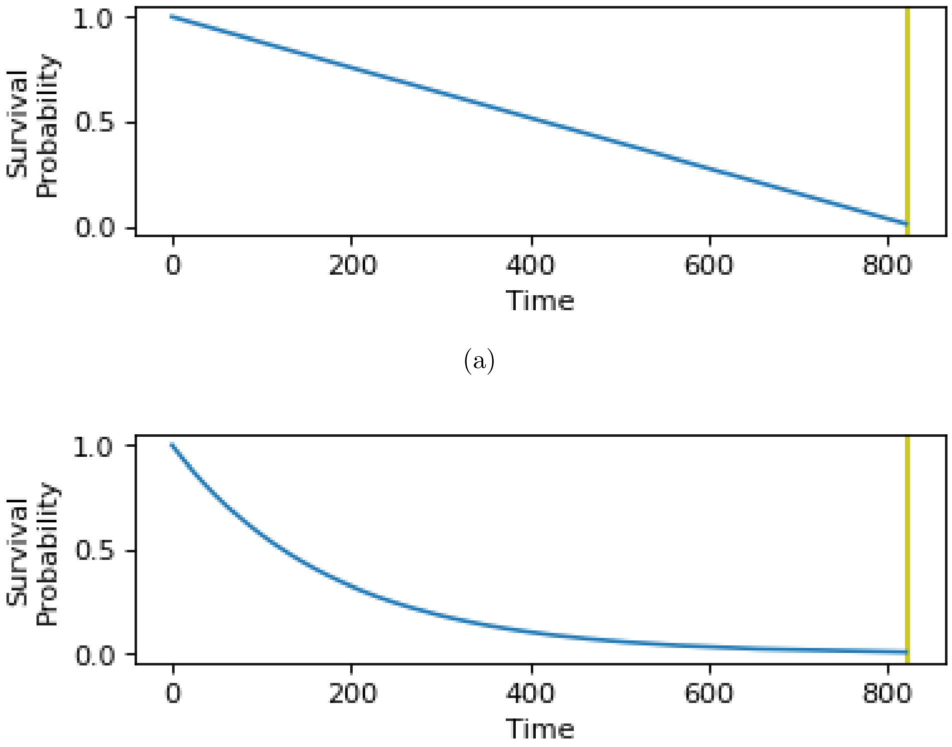
For the experiment we don't consider value of the weighting parameter ω_l, described in equation 3.2, since energy and delay are given equal weight to maintain the generality of the approach. However, for an actual implementation, ω_l may be leveraged to generate sensing schedules to optimize performance by prioritizing either energy reduction or delay reduction.

4.2 Offline System: Viterbi based Sensor Scheduling

The Viterbi algorithm is used in the offline stage to generate the sensing schedules for each sensor group in recognizing each context state according to the established behavior model by accounting for the likelihood of changing to a new context state at any time. The schedules and the associated energy consumption and delay values are stored in a Look-up Table for later use during the online stage. The sensing schedule is composed of a sequence of sensing decisions that directs the sensor operation during the recognition of a particular context state. An optimized schedule balances energy and delay since triggering the sensors too often leads to excessive energy consumption, and too little leads to increased delays. There are three aspects that increase the complexity of the sensor scheduling task. 1) The user may change their context state at any time, so there remains an uncertainty at all times in the sensing decision, which we represent with a survival probability [35]. 2) When making the sensing decision, an important factor is the time elapsed since the last time the sensor were turned on, creating a dependency between different the different elements of the vector representing the sensing schedule. 3) The number of sensing decision sequences grows exponentially with the number of triggering decisions required for each schedule. These aspects make enumeration methods unfeasible, and necessitates the use of algorithmic methods to find the solution to the sensor scheduling problem. We propose the use of a dynamic programming algorithm, called the Viterbi algorithm [36], to solve the sensor scheduling problem, with the aim of maximizing the utility of the possible combinations of sensors in recognizing multiple contexts concurrently. There are two specific aspects that are useful for determining an efficient sensing schedule. The first aspect are the time limits for a user in each state. For example, if a user is likely to spend a lot of time in a particular state, there would be no need to trigger the sensors often until the state is expected to change. The other useful aspect of the behavior model is knowing at anytime time the likelihood of state change, called state survival probability. The survival probability would be high when there it is unlikely that the user will have a state change. On the other hand, the probability would be low when it is imminent that the user will change state.

4.2.1 User Behavior State Survival Probability

For each state, the time limits are used to compute the survival probability, denoted $p^j(t, T^j_{l,h})$, of the user remaining in the state. The likelihood of staying in the state correlates with the length of time t the user spends in a state, and when the time durations reaches the time limits $T^j_{l,h}$, the survival probability reaches a close to zero value, denoted by ε. During the time preceding the time limits, the variation of the survival probability is subject to the distribution of the time durations found in the statistical record for each state. The survival probability may be modeled as a linear or an exponential decay. Figure 4.2(a) illustrates an example of the survival probability with a linear decay rate, while figure 4.2(b) illustrates an example of the survival probability having an exponential decay rate.



(a)

(b)

Figure 4.2: Illustration of the survival probability using the (a) Linear function (b) Exponential function.

A linear decay would be used when random changes in state are not exhibited by the distribution of the time durations, while an exponential decay would be used when changes in state occur more randomly. We note that an exponential function cannot decay to zero, thus, ε is used in place of zero as an asymptotic representation. On the other hand, a distribution of time duration for a state may exhibit a a uniform patter, where the time durations recorded are values close to the time limits only. So, the survival probability may be represented by a uniform function, having a high confidence in survival probability throughout the context recognition period, with a sudden drop within a range of the time limits, denoted by η. Figure 4.3 illustrates an example of the survival probability with a uniform representation.

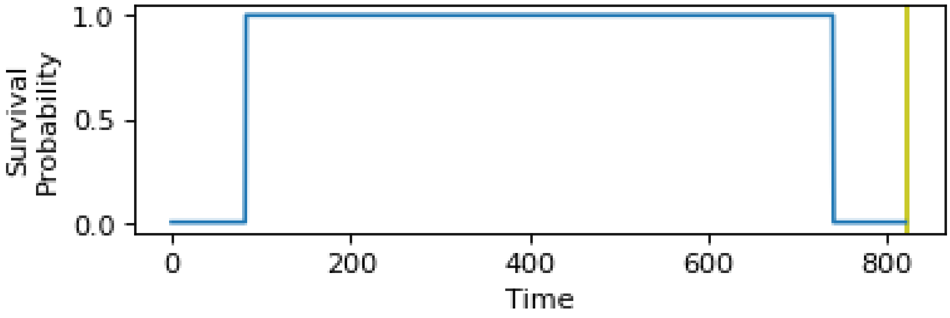
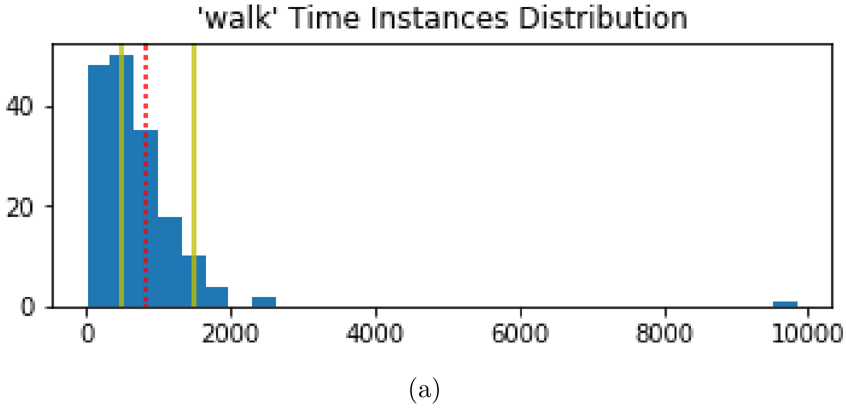
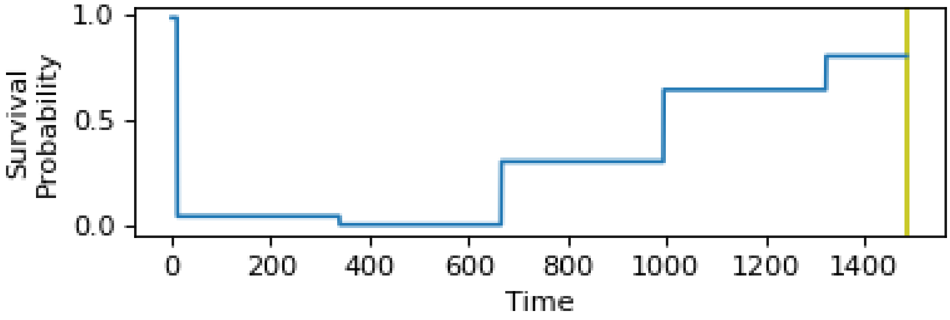


Figure 4.3: Illustration of the survival probability using the Uniform function.



(a)



(b)

Figure 4.4: (a) Histogram of time duration spent in a state found in the user historical record, (b) Survival probability using the Distribution function for the survival probability.

The probability may be formulated as a direct mapping of the distribution of the time durations recorded for a state, by taking the compliment of the density of each bin in the histogram as the value of the survival probability at the time corresponding to each bin. Figure 4.4(a) shows the distribution of time durations for the "walk' activity with the corresponding time limits, and the resulting survival probability is illustrated in figure 4.4(b). We discuss the mathematical representation of the different possible survival probabilities in the following section and the experiments we conducted to evaluate them in section 5.3.3.

4.2.2 Mathematical Representation of Survival Probability

The state survival probability decays until a time limit is reached, then the survival probability is reset to 1. When the maximum time limit is surpassed, sensing switches to continuous, i.e. the survival probability remains near zero (ε), and we do not have information about the behavior of the user. In the work [1] an Exponential function was used for the state survival probability, described mathematically as following:

$$p^j\left(t^i_{\mathcal{G}^l_n}, T^j_{l,h}\right) = e^{\ln(\varepsilon).\frac{t^i_{\mathcal{G}^l_n}}{T^j_{l,h}}} \tag{4.1}$$

The Uniform function can be described as following: The value is set to 1 anytime $t^i_{\mathcal{G}^l_n}$ is not near a time limit, otherwise if $t^i_{\mathcal{G}^l_n}$ is within a period of time duration that is a fraction η of a time limit, i.e. $\eta \times T^j$. or the beginning of the schedule, then the survival probability is ε. The Uniform function is described mathematically as following:

$$p^j\left(t^i_{\mathcal{G}_n}, T^j_{l,h}\right) = \begin{cases} 1, & \text{if } \eta \times T^j_{l,h} < t^i_{\mathcal{G}_n} < (1-\eta) \times T^j_{l,h} \\ \varepsilon, & \text{otherwise} \end{cases} \tag{4.2}$$

The Linear function, is simply a linearly decaying survival probability, starting at 1 and reducing to ε when a time limit is reached, and resetting to 1 if the time limit reached is not the final time limit $T^j_{l,H}$ following the below equation:

$$p^j\left(t^i_{\mathcal{G}^l_n}, T^j_{l,h}\right) = \left(\frac{-1+\varepsilon}{T^j_{l,h} - T^j_{l,h-1}}\right) * t^i_{\mathcal{G}_n} + 1 \tag{4.3}$$

In the time period before reaching the first time limit, where $h = 1$, then $T^j_{l,h-1} = 0$. The survival probability can be derived from the distribution of time duration, to produce what we called the distribution function. The function doesn't require all the time limits obtained, it only requires the last time limit $T^j_{l,H}$ as follows:

- The time durations found in the historical record for a given state are sorted in increasing order.
- Group the time durations into 30 bins, with each bin representing 3.3% of the longest time duration in the historical record.
- Count the number of time durations per bin.
- Calculate the survival probability assigned to each bin range, using the following formulation:

$$p^j\left(t^i_{\mathcal{G}^l_n}, T^j_{l,H}\right) = 1 - \frac{\text{\# of occurrences}}{\text{Largest \# of occurrences } \forall \text{ bins}} \tag{4.4}$$

Where $t^i_{\mathcal{G}^l_n} \in$ bin range. Moreover, we tested multiple bin sizes, 1%,2%, 3.3%, 5%, 6.6%, and 10% of the longest time duration in the historical record, with 3.3% giving the best result.

4.2.3 Reward Function

The Viterbi based algorithm generates a sensing schedule by making a decision at each time instant $t^i_{\mathcal{G}^l_n}$ whether to trigger the sensors or not to trigger. The possible sequential decisions form a path of decision nodes linked by edges, of sensing or non-sensing nodes, arriving at the final node at $t^i_{\mathcal{G}^l_n} = T^j_{l,H}$, where the user is most likely to transition into another context state. The utility of transition between nodes, or the sensor triggering decision, is measured by a metric called reward function $R()$, with each edge having its own defined metric. The Viterbi algorithm aims at finding the sensing schedule decisions that maximize the accumulated reward function over the entire schedule. Each sensing decision has an associated value, and maximizing the accumulated reward results in consuming less energy and incurring less delay. Mathematically, the Viterbi objective function aims to find the sequence of triggering decisions $a^i_{\mathcal{G}^l_n}$ to maximize the accumulated reward function as follows:

$$\underset{a^i_{\mathcal{G}_n}}{\operatorname{argmax}} \sum_{i=1}^{I_{\mathcal{G}_n}} R\left(a^i_{\mathcal{G}^l_n}, \Delta t^{i+1}_{\mathcal{G}^l_n}, p^j(t^i_{\mathcal{G}^l_n}, T^j_{l,h})\right) \tag{4.5}$$

- $a^i_{\mathcal{G}_n}$ represents the sensing schedule, where for each i, a sensor triggering decision is made at $t^i_{\mathcal{G}^l_n}$.
- $I_{\mathcal{G}_n}$ is the last decision instance before sensing becomes continuous, such that $I_{\mathcal{G}_n} = \frac{T^j_{l,H}}{\delta_{\mathcal{G}^l_n}}$.

- $\Delta t_{\mathcal{G}_n^l}^{i+1}$ is the elapsed time since the preceding sensor "Sense" decision.
- $p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)$ is the state survival probability.

The reward function takes into account the energy consumption of the sensors relative to each other and for the accuracy of the sensor group in recognizing context state. Accounting for accuracy is necessary since an incorrectly recognized context state would lead to additional delays, due to the elapsed time until the state is correctly recognized. The function also takes into account the possible delay, based on the time period since the preceding sensor activation decision, i.e. when the sensing schedule turned the sensor on. The state of the user is not known before triggering the sensors, so the reward function is probabilistic and depends on the likelihood of having transitioned into a new state which is captured by a survival probability $p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)$. For $t_{\mathcal{G}_n^l}^i = T_{l,h}^j$, the survival probability reduces to near zero, i.e. ε, and sensing switches to continuous, The value of the reward function depends on the combination of the decision taken and the survival probability value. The reward function is divided into two instantaneous rewards $r()$, the first instantaneous reward represents the case where a state does not change at the new time instance, while the second instantaneous reward is when a state change occurs. Mathematically, the reward function can be derived from the expected value of the instantaneous rewards:

$$\begin{aligned} R\left(a_{\mathcal{G}_n^l}^i, \Delta t_{\mathcal{G}_n^l}^{i+1}, t_{\mathcal{G}_n^l}^i\right) &= E_s\left[r\left(x_l^j, a_{\mathcal{G}_n^l}^{i+1}, \Delta t_{\mathcal{G}_n^l}^{i+1}\right)\right] \\ &= p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j) \cdot r\left(x_l^j, a_{\mathcal{G}_n^l}^{i+1}, \Delta t_{\mathcal{G}_n^l}^{i+1}\right) + \left(1 - p^j(t_{\mathcal{G}_n^l}^i, T_{l,h}^j)\right) \cdot r\left(\overline{x_l^j}, a_{\mathcal{G}_n^l}^{i+1}, \Delta t_{\mathcal{G}_n^l}^{i+1}\right) \end{aligned} \tag{4.6}$$

E_s is the expected reward that depends on whether the current state being recognized has changed or not, i.e. x_l^j or $\overline{x_l^j}$ respectively. At each time instant $t_{\mathcal{G}_n^l}^i$ there is a probability $p^j(t_{\mathcal{G}_n^l}^i, T_{l,H}^j)$. To make decisions at each step whether to trigger the sensors or not, a value is attributed to the instantaneous reward $r()$, to calculate the best triggering decision for the next step, i.e. $a_{\mathcal{G}_n^l}^{i+1}$. To penalize long periods of inactivity of the sensors, that might lead to delays, and to reward quick recognition of a state change, we measure the difference in time between the next decision and the last triggering instance as $\Delta t_{\mathcal{G}_n^l}^{i+1}$.

The instantaneous reward reflects the impact of sensing decision on consumed energy and delay in detecting a state change. The energy reward component is computed as:

$$-\frac{E_{\mathcal{G}_n^l}^j}{E_{\mathcal{G}_n^l,\max}} \tag{4.7}$$

Where $E_{\mathcal{G}_n^l}$ is the energy cost of triggering sensor group $\mathcal{G}_n^l$, and it is normalized by the maximum instantaneous energy value $E_{\mathcal{G}_n^l,max}$ of the available sensor groups to recognize the requested context. The delay component is reflected in the combination of two terms; the probability of recognition captured by accuracy of the model and the amount of delay captured by time elapsed $\Delta t_{\mathcal{G}_n^l}^{i+1}$ since last sensing decision:

$$-\alpha.(\Delta t_{\mathcal{G}_n^l}^{i+1}) + \frac{\beta.A_{\mathcal{G}_n^l}}{(\Delta t_{\mathcal{G}_n^l}^{i+1})} \tag{4.8}$$

Depending on the combination of triggering decision taken (Sense or Don't Sense) and user state x_l^j or $\overline{x_l^j}$ (”No State Change” or ”State Change”), the instantaneous reward is a weighted sum of the two components: 1) Energy and 2) Time Elapsed.

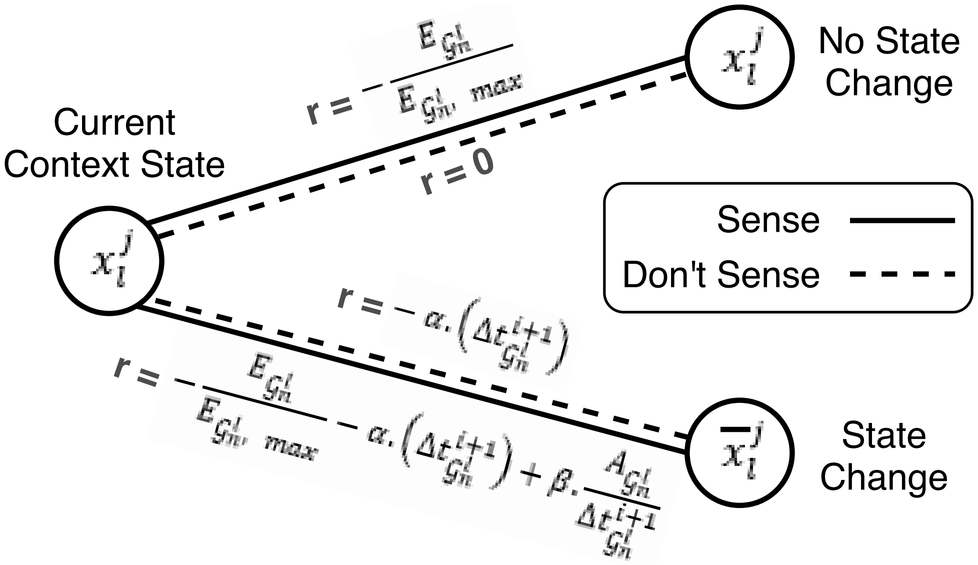

Current Context State
x_l^j
No State Change
x_l^j
State Change
$\overline{x}_l^j$
Sense
Don't Sense
$r = -\frac{E_{\mathcal{G}_n^l}}{E_{\mathcal{G}_n^l,\ max}}$
$r = 0$
$r = -\alpha.\left(\Delta t_{\mathcal{G}_n^l}^{i+1}\right)$
$r = -\frac{E_{\mathcal{G}_n^l}}{E_{\mathcal{G}_n^l,\ max}} - \alpha.\left(\Delta t_{\mathcal{G}_n^l}^{i+1}\right) + \beta.\frac{A_{\mathcal{G}_n^l}}{\Delta t_{\mathcal{G}_n^l}^{i+1}}$

Figure 4.5: State transition diagram showing the instantaneous reward values for the different conditions, of ”Sense” and ”Don't Sense” sensor triggering decision, depending on whether the state has changed or not.

Figure 4.5 shows the instantaneous reward values with the state change to triggering action conditions. The triggering decision are represnted in the figure by a line for the”Sense” decision, and dashed line for the ”Don't Sense” decision. When the decision is to sense and the state does not change (x_l^j), then the instantaneous reward is represented by the energy cost penalty. However, if the state has indeed changed ($\overline{x_l^j}$), then in addition to the energy penalty, the time elapsed component is added. The time elapsed component is composed of two terms, a delay penalty weighted by α, and a state recognition reward weighted by β. The delay term penalizes the instantaneous reward in proportion to the time elapsed $\Delta t^{i+1}_{\mathcal{G}^l_n}$. The state recognition term is inversely proportional to $\Delta t^{i+1}_{\mathcal{G}^l_n}$ to reward recognizing the change in state as soon as possible and is proportional to the accuracy of the sensor group $A^j_{\mathcal{G}^l_n}$ in recognizing state $x^j_l$to account for the chance of incorrect recognition of the context state. Incorrect recognition of the context state can take the form of false positives or false negatives, which lead to additional delays and energy consumption because additional sensor triggers are required before the correct state is recognized. Thus, accuracy is directly related to delay, and needs to be accounted for in the reward function to optimize the sensing schedule based on the specifications of the sensor groups used for context recognition.

The α and β are weighting factors, and their combination is specified by finding the optimal Pareto solution to an optimization formulation 4.9, as applied to a single sensor group takes the form:

$$\underset{(\alpha,\beta)}{\operatorname{argmin}}\, \omega_l \frac{E^j_{\mathcal{G}^l_n}}{E^j_{\mathcal{G}^l_n,\max}} + (1-\omega_l)\,\frac{D^j_{\mathcal{G}^l_n}}{D^j_{l,\max}} \tag{4.9}$$

The terms in the formulation are the same as those found in equation 3.2, only they are related to single sensor group ($E^j_{\mathcal{G}^l_n}$), rather than the union of multiple groups ($\mathbf{U}E^j_{\mathcal{G}^l_n}$). Once the optimal combination of α and β are obtained, they are applied to the reward function to find the optimal sensing schedule for each sensor group, following the steps described in Algorithm 2 to fill the LUT with the generated sensing schedules.

Algorithm 2 Sensor Scheduling Algorithm

Input: π_l, S^m, $A_{\mathcal{G}_n^l}$, E_m, $T_{l,h}^j$

From CRM KB:
π_l: the choices of groups of sensors
$\mathcal{G}_n^l$ for each of the requested L contexts S^m: the available sensors
$A_{\mathcal{G}_n^l}$: Accuracy of group $\mathcal{G}_n^l$
E_m: the energy consumption by each sensor S^m

From User Behavior:
$T_{l,h}^j$: the time limits of the context states x_l^j

Output: $E_{\mathcal{G}_n^l}^j$, $D_{\mathcal{G}_n^l}^j$, (α, β) $a_{\mathcal{G}_n^l}^i$
$E_{\mathcal{G}_n^l}^j$: the energy consumption for each group of sensors $\mathcal{G}_n^l$ in recognizing context state x_l^j
$D_{\mathcal{G}_n^l}^j$: the delay of each sensor group $\mathcal{G}_n^l$ in recognizing context state x_l^j
(α, β): the Pareto optimal weighting factors for the Viterbi reward function in recognizing context x_l^j with sensor group
$a_{\mathcal{G}_n^l}^i$: the sensing schedules for the sensor groups in π_l in recognizing each context state.

```
for each sensor group $\mathcal{G}_n^l \in \pi_l$ do
    Calculate $E_{\mathcal{G}_n^l}$ as the sum of $E_m$ of $S^m \in \mathcal{G}_n^l$
    for each context state $x_l^j$ do
        for each $(\alpha, \beta)$ pair do
            Run Viterbi algorithm to determine sensing schedule for $\mathcal{G}_n^l$ in
              recognizing state $x_l^j$
            Compute the objective function value according to 4.9
            Track the resulting energy consumption, delay, and the $(\alpha, \beta)$ pair
              for the derived sensing schedule.
        end for
        Store the sensing schedule $a_{\mathcal{G}_n^l}^i$ with the minimum objective function
          value and the associated energy consumption $E_{\mathcal{G}_n^l}^j$, delay $D_{\mathcal{G}_n^l}^j$,
          and $(\alpha, \beta)$ pair in the Look-Up Table (LUT)
    end for
end for
```

Chapter 5

Experiments and Results

We conducted a set of experiments to evaluate the performance of the proposed method. The experiments include the evaluation of three aspects: the holistic approach that simultaneously determines the optimized group of sensors and their schedules, the behavior model, and the survival probability. The holistic approach is compared to state of the art [15], which used a single pattern of time spent in a given state and was based on hierarchical selection of groups then determining sensing schedules.

Table 5.1: CRM KB Sensor Groups Data

Context	Ref.	Group ($\mathcal{G}_n^l$)	Sensors (S^m)	Energy (mJ) ($E_{\mathcal{G}_n^l}$)	Accuracy ($A_{\mathcal{G}_n^l}$)
Activity	[37]	$\mathcal{G}_1^1$	Acc.[1], Compass, PPG[2], ECG[3], Resp. [4], SaO2[5], Skin Temp.[6]	1.22	86%
	[38]	$\mathcal{G}_2^1$	Acc.	0.3	80%
Location	[30]	$\mathcal{G}_1^2$	GPS	2.5	85%
	[30]	$\mathcal{G}_2^2$	Acc., GPS, GSM	1.1	71%
Health	[39]	$\mathcal{G}_1^3$	Acc., ECG, Skin Temp.	0.62	91%
	[40]	$\mathcal{G}_2^3$	ECG	0.12	76%

[1] Acc.: accelerometer

[2] PPG: photoplethysmogram

[3] ECG: electrocardiogram

[4] Resp. refers to respiration sensor

[5] SaO2: oxygen saturation sensor

[6] Skin Temp.:skin temperature sensor

5.1 Experimental Setup

The sensors specifications used in the experiments are listed in table 5.1 along with the related references. The accuracy values obtained from the references are for the best case outcome. To reflect a practical scenario of using relatively accurate recognition models, we selected the sensor groups that achieve recognition accuracy above 70%. The information found in the CRM KB are obtained from published research papers of energy efficient context recognition systems. In the experiments we conducted we used the information found in [37, 38, 30, 39, 40].

To simulate context scenarios, we generated a dataset using the Smarter Time Android application to track activities of daily life, location, and time spent in each recognised activity and location over a period of two months. The application captures user contexts state time durations, which can be used to represent user behavior [41]. Three categories of context were recorded: Activity, Location, and Health. The tracked states for the activity context included ’Walk’, ’Sit’, and ’Jog’. Location states included ’Home’ and ’Work’. Health states included ’Healthy’ and ’Unhealthy’. At any time a user can be in any combination of states for the monitored contexts. For example, a multi-context scenario may consist of the case where the user is sitting at home and is healthy. The user would be in the ’Sit’ state for activity context, the ’Home’ state for location context, and the ’Healthy’ for state health context. The frequent time limits, representing the user behavior in the different states are summarized in table 5.2.

Table 5.2: User Behavior Time Limits for Different Context States

Context	State (x_l^j)	Single Time Limit (T_l^j)	Multiple Time Limits ($T_{l,h}^j$)
Activity	$x_1^1 =$ ”Walk”	$T_1^1 = 2675$	$T_{1,1}^1 = 3431$, $T_{1,2}^1 = 10274$
	$x_1^2 =$ ”Sit”	$T_1^2 = 612$	$T_{1,1}^2 = 696$, $T_{1,2}^2 = 2065$, $T_{1,3}^2 = 3433$
	$x_1^3 =$”Jog”	$T_1^3 = 825$	$T_{1,1}^3 = 880$, $T_{1,2}^3 = 2619$, $T_{1,3}^3 = 4357$
Location	$x_2^1 =$ ”Home”	$T_2^1 = 1929$	$T_{2,1}^1 = 1720$, $T_{2,2}^1 = 1481$, $T_{2,3}^1 = 5142$
	$x_2^2 =$ ”Work”	$T_2^2 = 1548$	$T_{2,1}^2 = 503$, $T_{2,2}^2 = 1481$, $T_{2,3}^2 = 2460$
Health	$x_3^1 =$ ”Healthy”	$T_3^1 = 2334$	$T_{3,1}^1 = 2295$
	$x_3^2 =$”Unhealthy”	$T_3^2 = 403$	$T_{3,1}^2 = 521$

5.2 System Parameters

The Pareto optimal (α, β) combination is selected, by enumerating the different possible combination to find the pair that gives minimum point according to equation (8).

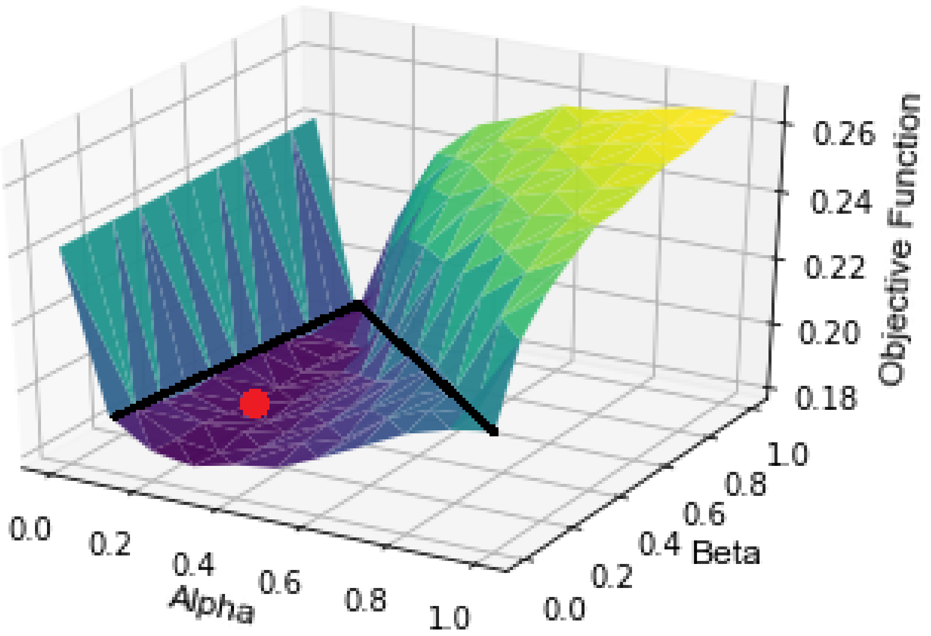
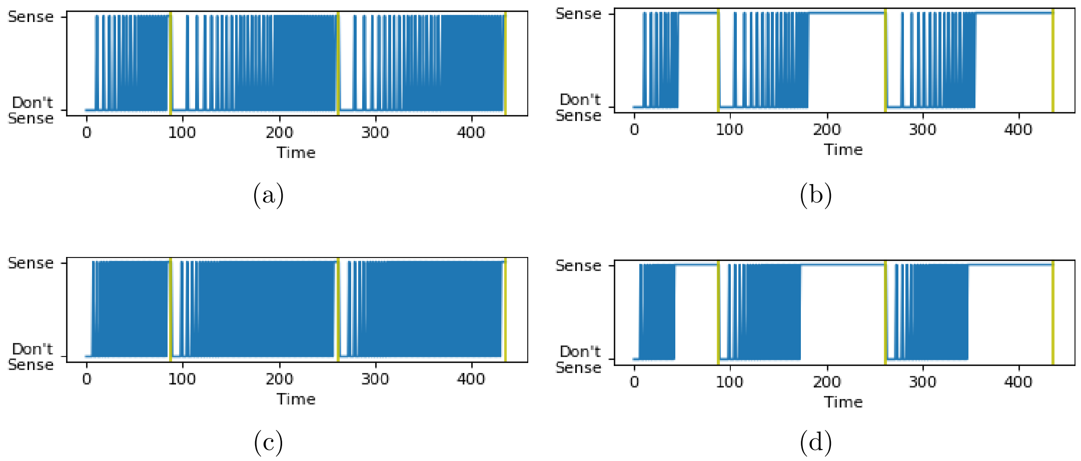


Figure 5.1: Sample plot of the parameters alpha and beta vs the resulting objective value, resulting from the sensing schedule for different combinations of (α, β). The red dot represents the chosen optimal combination (α, β) and the black lines in represents a boundary condition of (α, β) beyond which the objective function value increases exponentially.

As shown in figure 5.1, it was observed that values of $\alpha + \beta > 1$ lead to sudden increase in objective function and divergence from global optimal. Also, the choice of $\alpha = 0$ leads to no sensing, which is to be avoided. As a result, the values of (α, β) where limited to satisfy the condition:

$$\alpha + \beta \leq 1 \quad \& \quad \alpha > 0 \tag{5.1}$$

In figure 5.1, the black lines indicate the boundaries of the search space and the brown dot is the (α, β) combination that corresponds to minimum objective function value. The sensing schedules in figure 5.2(a) and 5.2(c) correspond to values of (α, β) that are within the bounds of search space, and the sensing schedule alternates between "Sense" and "Don't Sense" up to each time limit. On the other hand, the sensing schedules in figures 5.2(b) and 5.2(d) correspond to values of (α, β) outside the bounds of the search space, and they show a consistent "Sense" decision without alternating back to "Don't Sense" long before reaching the time limits.



(a)

(b)

(c)

(d)

Figure 5.2: Sensing schedules for (α, β) combinations showing the continuous sensor triggering phenomenon for values of (a) $\alpha = 0.3$, $\beta = 0.7$, (b) $\alpha = 0.3$, $\beta = 0.8$, (c) $\alpha = 0.8$, $\beta = 0.2$, and (d) $\alpha = 0.8$, $\beta = 0.3$

For the experiment, we set $\delta = 10s$ for all contexts to simplify the experiments, Moreover, the Exponential function for the survival probability is used for both the holistic and hierarchical approaches.

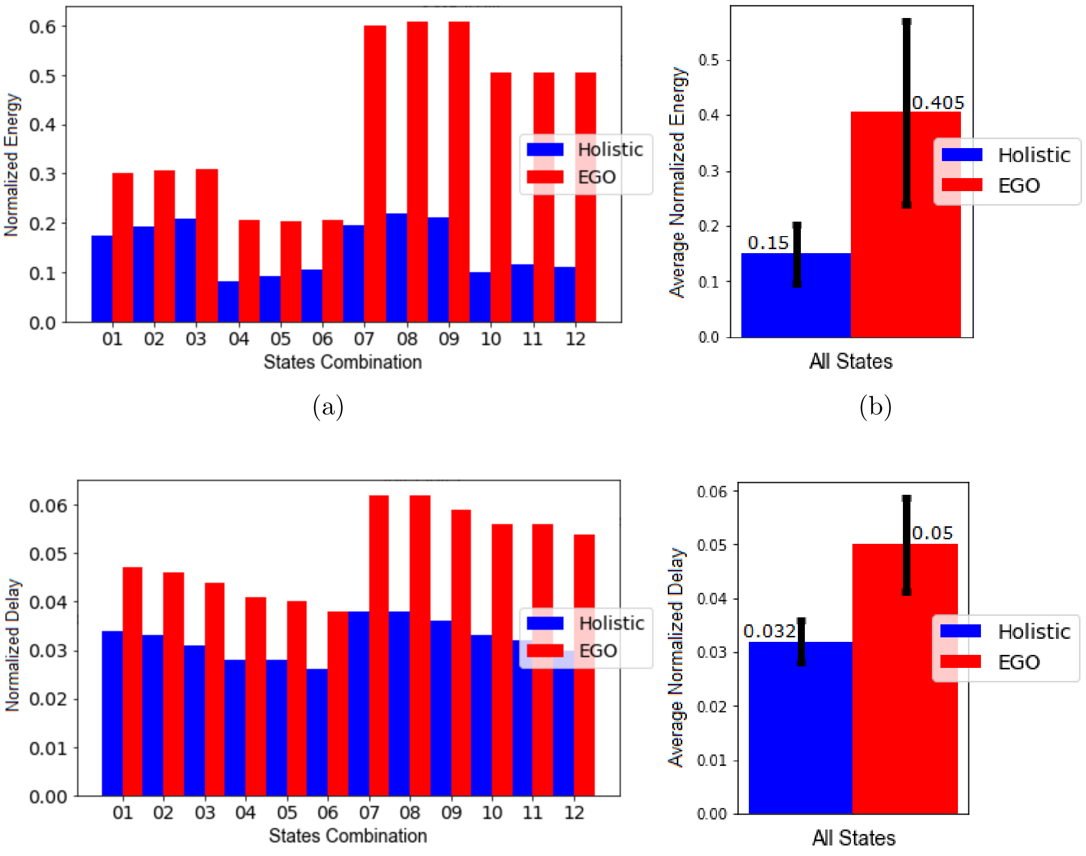
5.3 Comparison to State of the Art

In this section, we compare our proposed holistic approach to sensor selection and scheduling with state of the art approach (EGO [15]) that is based on hierarchical decision making by first selecting sensors then deciding on their schedule of operation. We consider all possible combinations of states for the three monitored contexts detailed in Table 5.2, making up 12 context scenarios.

5.3.1 Effect of Holistic Approach Versus Hierarchical

To test the merit of holistic versus hierarchical approach without other improvements, we assume all other conditions are the same for both approaches. This includes using the same sensor information and the same behavioral model for both. In particular, we use the proposed behavior model of multiple time limits for both methods. The results of the experiments are shown in figure 5.3.

For each of the 12 context scenarios, figure 5.3(a), shows the normalized energy values obtained for both the holistic and the hierarchical approach (EGO), while figure 5.3(c) shows the normalized delay values. For all the cases, the holistic approach performs better than the hierarchical in both delay and energy consumption. The actual energy consumption values in Joules and delay in sec-



(a) (b)

(c) (d)

Figure 5.3: Comparison between EGO and the holistic approach both using the new behavioral model for (a) normalized energy and (c) normalized delay for each states combination scenario, (b) Average normalized energy and (d) average normalized delay for all states combination scenarios

onds can be obtained by multiplying the normalized value with the maximum values, as previously illustrated in computations of energy and delay. However, since the purpose is to do a comparison, the actual values in terms of the corresponding units are not necessary. In figure 5.3(b), the holistic approach resulted in a normalized energy value of 0.15 (110 mJ) while the average normalized energy value for EGO was 0.405 (179 mJ), meaning the holistic approach showed an average improvement of 63% in normalized energy. In figure 5.3(d), the holistic approach resulted in a normalized delay value of 0.033 (11 seconds) while the average delay value for EGO was 0.05 (14 seconds), meaning the holistic approach showed an average improvement of 34% in normalized delay. When summing the delay and energy, the cumalative improvement is 60% in the objective function. Additionally, the standard deviation of the normalized energy value for the holistic approach is smaller than the standard deviation of EGO, as can be seen by the black line at the center of the bars in figure 5.3(b), indicating a more stable performance.

5.3.2 Performance Analysis

To study more closely the effect of using the holistic approach, we examine one of the context scenarios to find out why the holistic approach performs better. For the first state combination, where the states are, 'Jog', 'Work', and 'Unhealthy' , i.e. the scenario for the bar in figure 5.3(a) labeled '01'. As such, we look at all the possible sensor group combinations with their respective normalized energy and normalized delay values, as shown in figure 5.4, which represents all the sensor group combinations, with their respective normalized delay and energy.

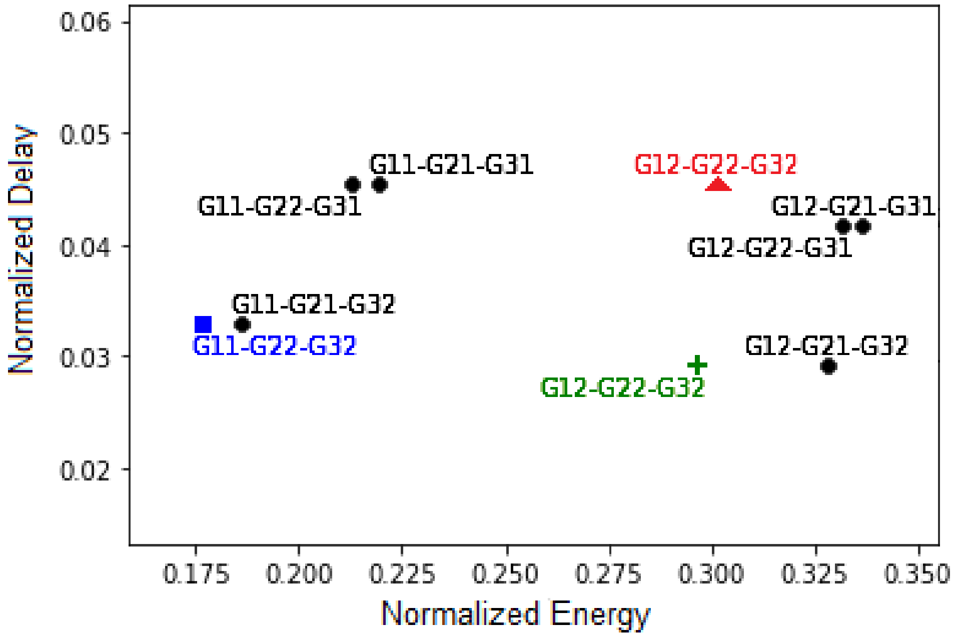


Figure 5.4: Representations of all sensor group combinations terms of their normalized delay and energy. The black dots are group combinations not selected by either method, the red triangle is the groups selected by EGO, the green cross is the same groups selected by EGO but with the values resulting from using the holistic approach, and the blue square is the group selected by the holistic approach

The black dots are for sensor combinations that have not been selected by either method, the blue square is for the group combination selected by the holistic approach and the red triangle is for the groups selected by EGO [15]. Additionally, the figure shows the sensor group combination chosen by EGO, but from the application of the Viterbi algorithm used in the holistic approach represented by the green cross.

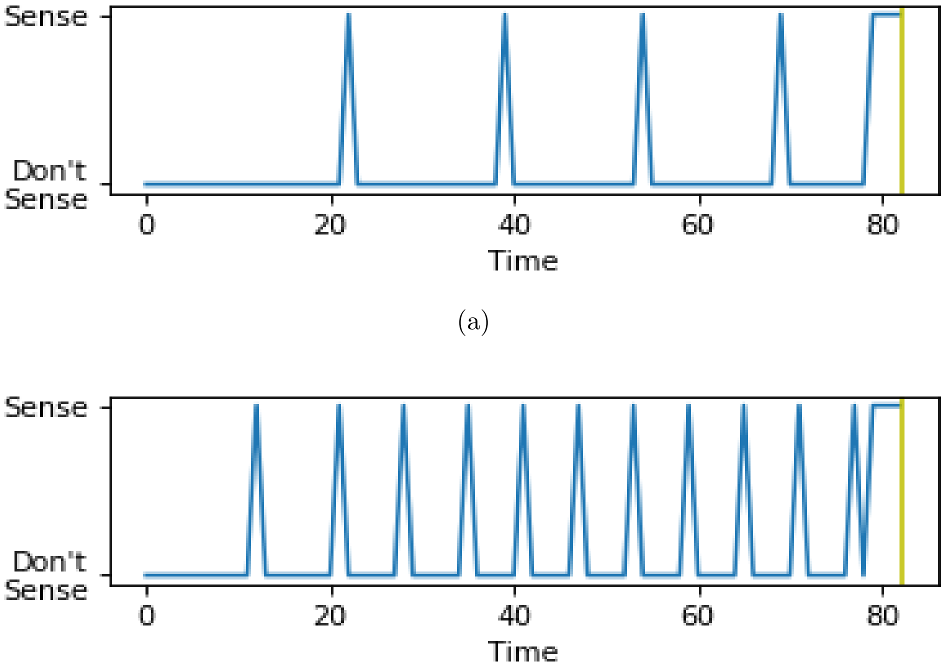
The groups selected by the holistic approach are $\mathcal{G}_1^1$ (G11), $\mathcal{G}_2^2$ (G22), and $\mathcal{G}_2^3$ (G32) while the groups selected by EGO are $\mathcal{G}_2^1$ (G12), $\mathcal{G}_2^2$, and $\mathcal{G}_2^3$. Both methods select the same groups, $\mathcal{G}_2^2$, and $\mathcal{G}_2^3$, for the location and health contexts but different groups, $\mathcal{G}_1^1$ and $\mathcal{G}_2^1$, for the activity context. The group selected by the holistic approach has a lower objective function value (0.21), while the combination selected by EGO has a greater value (0.35). As reflected in table 5.1, group $\mathcal{G}_1^1$ has the accelerometer sensor in common with group $\mathcal{G}_2^2$ and the ECG in common with $\mathcal{G}_2^3$, while $\mathcal{G}_2^1$ only has the accelerometer in common with group $\mathcal{G}_2^2$. Thus, by synchronizing the sensing schedules $a^i_{\mathcal{G}_1^1}$ and $a^i_{\mathcal{G}_2^2}$ for the accelerometer sensor, and $a^i_{\mathcal{G}_1^1}$ and $a^i_{\mathcal{G}_2^3}$ for the ECG sensor, the operation of group $\mathcal{G}_1^1$ reduced the energy costs attributed to both group $\mathcal{G}_2^2$, and $\mathcal{G}_2^3$. By taking advantage of the operation of the accelerometer and ECG as part of $\mathcal{G}_1^1$, the operational energy cost was reduced for $\mathcal{G}_2^2$ and $\mathcal{G}_2^3$. While for the groups selected by EGO, because $a^i_{\mathcal{G}_2^1}$ and $a^i_{\mathcal{G}_2^2}$ are synchronized for the accelerometer sensor, the operation of group $\mathcal{G}_2^1$ only reduced the energy costs of group $\mathcal{G}_2^2$, but not enough to surpass the energy reduction in the holistic approach.

In addition to energy improvements, Figure 5.4 shows an improvement in delay. The improvement is attributed to an improved representation of energy consumption of the sensor groups in the instantaneous reward of Viterbi algorithm. In EGO the energy is represented in terms of a constant and does not account for the differences in energy consumption between possible selections sensor groups to recognize a context. Which can represent the relative difference in energy consumption between the possible groups of sensors.

Figure 5.5 illustrates the difference in the sensing schedules between EGO, figure 5.5a, and the holistic approach, figure 5.5b, for the same selection sensor group $\mathcal{G}_2^3$. More frequent sensor triggering leads to more energy consumption, however, the increased energy consumption is compensated by the selection of additional common sensors leading to reduced energy consumption over all the expected operation time. More frequency sensing also leads to a lower delay value.

5.3.3 Impact of Multiple Time Limits

Here, we compare the proposed solution with both aspects of holistic optimization and improved behavioral model versus the original state of the art (EGO) using their own behavioral model which consists of averaging the historical time duration data spent in each state and adding a fraction of the standard deviation to compute a single time limit per state. The normalized energy, normalized delay, and the resulting objective function values for the 12 combinations of states and the average values are examined for the holistic approach and EGO in figure



(a)

(b)

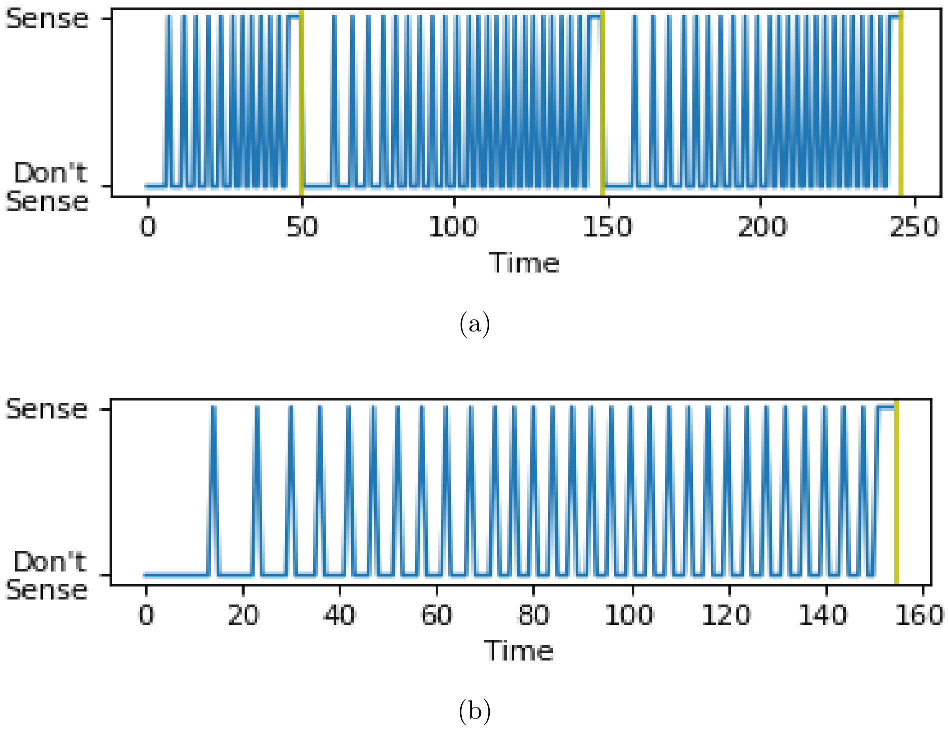
Figure 5.5: Sensing schedule generated using Viterbi algorithm for sensor group $\mathcal{G}_2^3$ with the reward function of (a) EGO and (b) Holistic approach

5.7. To keep the comparison fair, the maximum values in the denominators used for normalization is kept the same across both approaches, because the maximum values of energy and delay are different for the two approaches due to the difference in behavior modeling.

The use of the different modeling techniques, i.e. the binning technique and the averaging technique, results in different sensing schedules applied to a sensor group when detecting a state.

To illustrate the difference, Figure 5.6(a) has three time limits, obtained using the binning technique, each time a time limit is approached more sensing is encouraged by the Viterbi algorithm, but once it is passed, sensing triggers become scarcer. Figure 5.6(b) shows the result of the previous method in the literature [1] and that relies on only one time limit. Note that at the last time limit $T_{l,H}^{j}$, continuous sensing is applied until a change in state is detected.

For each of the 12 context scenarios, figure 5.7(a), shows the normalized energy values obtained for both the holistic and the hierarchical approach (EGO), while figure 5.7(c) shows the normalized delay values. As before, the holistic approach performs better than the hierarchical in both delay in state change detection and energy consumption for all the cases. In figure 5.7(b), the holistic approach resulted in an average reduction of 68% in normalized energy. In figure 5.7(d), the



(a)

(b)

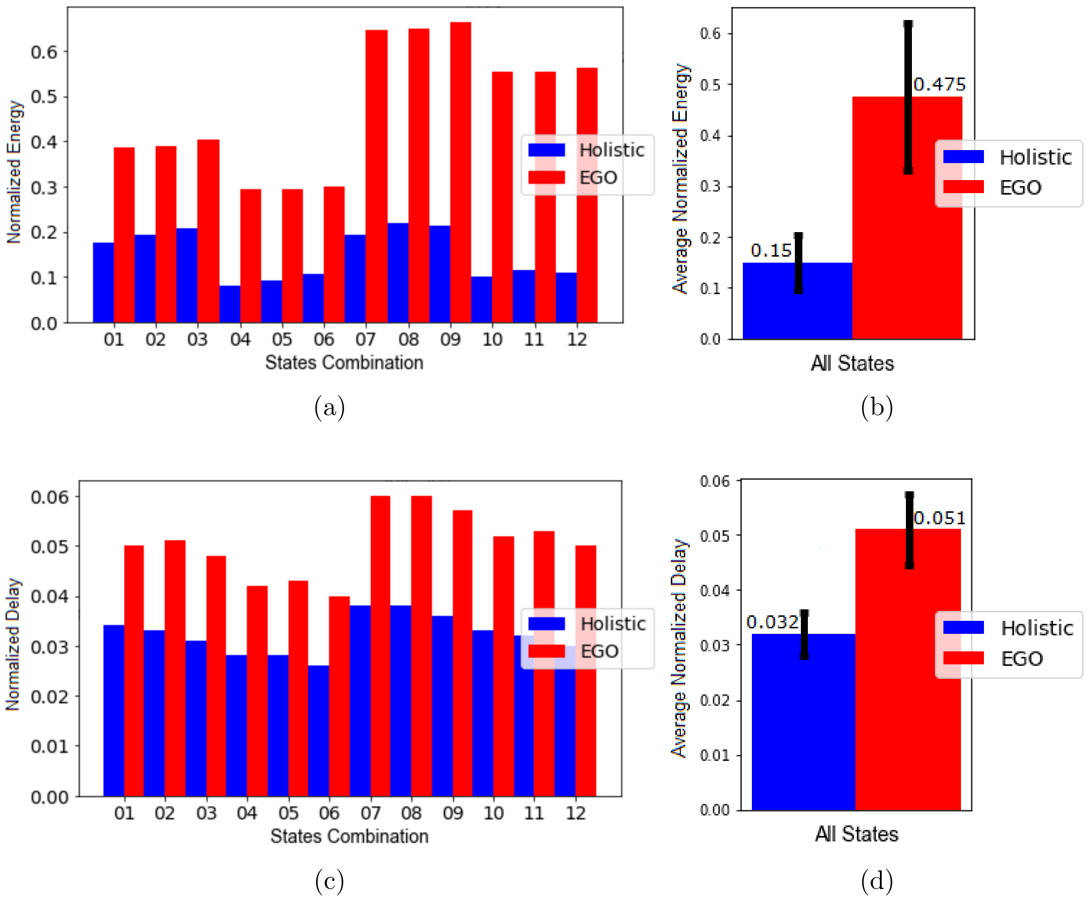
Figure 5.6: Sensing schedules using the same sensor group to detect the same state, with a) using the binning technique and b) using the VCAMS [1] method for modeling the behavior of the user.

holistic approach showed an average reduction of 34% in normalized delay, which is the same as before, meaning the modified behavioral model reduces energy consumption while maintaining the same levels of delay. Taking the sum of the energy and delay terms, the holistic approach showed an average reduction 65% for the objective function value using the holistic approach compared to EGO. In summary, the improvements attributed to the use of the binning technique are 5% for the normalized energy, 0% for the normalized delay.

5.3.4 Impact of State Survival Probability

Finally, we test the impact of the survival probability function on the sensing schedule generated by the Viterbi Algorithm as illustrated in figure 5.8.

The Uniform function in figure 5.8(a) shows sensing decisions at the start of the schedule and around the time limits. The Uniform function may lead to optimized results in cases of high confidence that the user only changes their state close to the obtained time limits, as that would reduce the obtained delays while reducing energy consumption drastically. For this experiment the Uniform function performed poorly because the user data is more stochastic and less de-



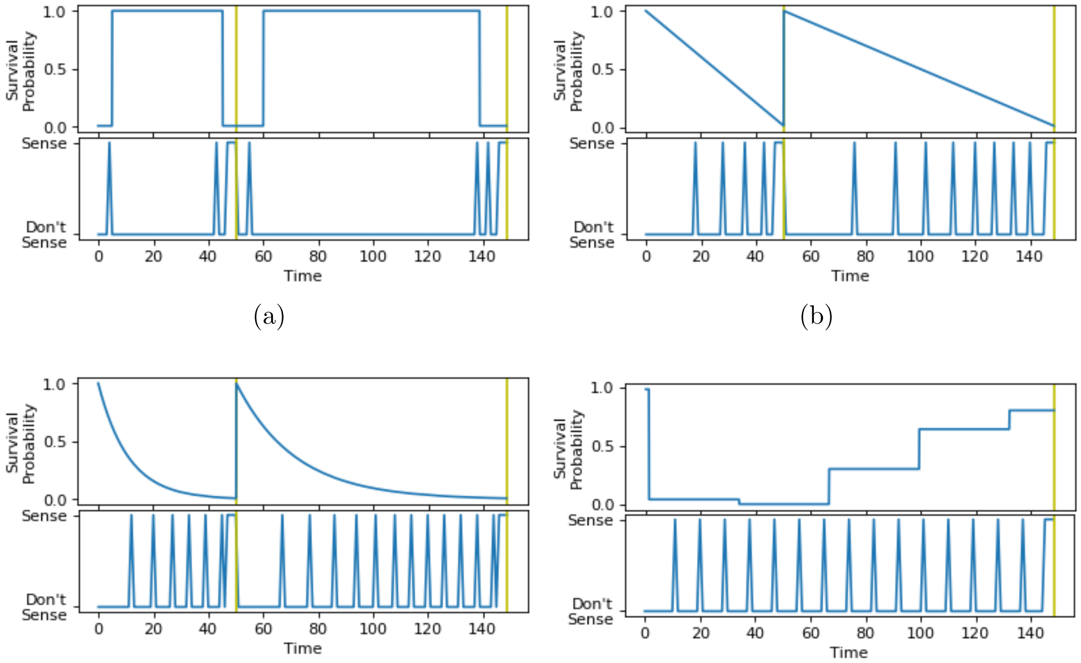
(a) (b)

(c) (d)

Figure 5.7: Comparison between the holistic approach using the new behavioral model vs EGO its own behavior model for (a) normalized energy and (c) normalized delay for each states combination scenario, (b) Average normalized energy and (d) average normalized delay for all states combination scenarios

terminable, i.e. we cannot know for sure that the state of the user will change exactly according to the behavioral model. Therefore, the distribution of the time duration data, needs to be taken into consideration when choosing the function modeling the survival probability of context states. The Linear function in figure 5.8(b) shows sensing decisions in the schedule that gradually become more frequent until reaching the time limits. The Exponential function in figure 5.8(c) shows sensing decisions in the schedule that quickly become more frequent, faster than that of the Linear function, until reaching the time limits. The Distribution function in figure 5.8(d) shows sensing decisions in the schedule of a constant frequency, starting at the lowest point in the survival probability.

The results for all context scenarios of the average normalized energy consumption and average normalized delay are shown in figure 5.9(a) and 5.9(b)



(a)

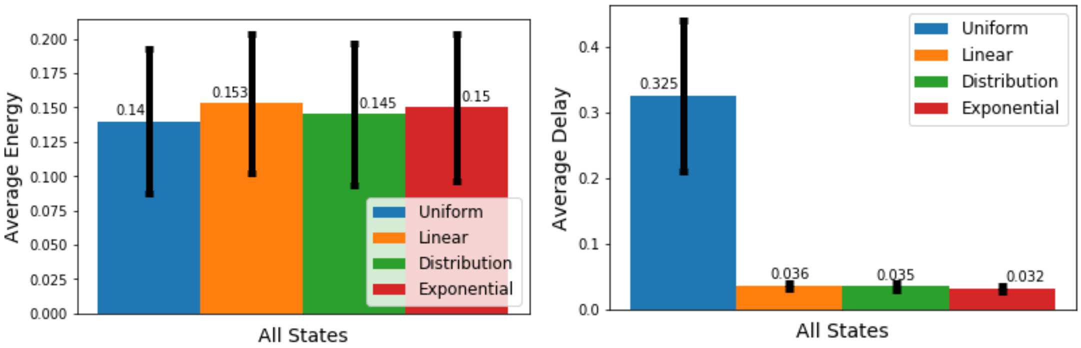
(b)

(c)

(d)

Figure 5.8: Illustration of impact on the sensing schedule generated by the Viterbi algorithm using the (a) Uniform function (b) Linear function (c) Exponential function (d) Distribution function

respectively.



(a)

(b)

Figure 5.9: (a) Average normalized energy consumption value for all context scenarios using the different survival probabilities (b) Average normalized delay value for all context scenarios using the different survival probabilities

In figure 5.9(a) the best result, in terms of energy consumption is the Uniform function followed by the Distribution function, while the worst is Linear function and followed by the Exponential function. In figure 5.9(a), the best result, in terms of delay is the Exponential function and followed by Distribution function, while the worst is Uniform function. Thus, the choice of survival probability function will impact the overall outcome, so it is beneficial to have an accurate representation of of state survival probability to ensure optimized sensing decisions.

5.4 Computational Complexity

In the offline stage, the holistic Viterbi algorithm is applied repeatedly to find the Pareto optimal weighting parameters, the number of times the algorithm is applied depends on how many combinations of (α, β) are to be examined, i.e. complexity is $O(N^2)$. We note that because of the (α, β) boundary conditions, as opposed to the previous work in VCAMS [1], the complexity becomes $O(\frac{1}{2}N^2)$. Additionally, since each iteration of (α, β) combination, is independent of the other, then the process of finding the optimal combination is parallelizable. The complexity of applying the Viterbi algorithm is $O(I.A^2)$, where I is the total number of time instances in which triggering decisions are made, and A is the number of distinct actions that can possibly be taken, which are to sense or not to sense. As for the complexity in the online stage, the computational complexity is dominated by the number of possible combinations $\Pi_{l=1}^{L} \mathcal{G}_n^{l'}$ for sensor groups to recognize the desired contexts. Thus the complexity is $O(L.N)$, where N is the number of groups per context, and L is the number of desired contexts for recognition.

Chapter 6

Conclusion

The paper described a holistic optimization approach to minimize energy consumption and delays in the simultaneous detection of multiple context states. The problem was formulated as an optimization problem, to simultaneously decide on the sensors and their sensing schedules. The contributions included a new user behavioral model based on capturing user's frequent patterns in every context state and the Viterbi instantaneous reward functions captured normalized energies allowing comparison across groups of sensors and the accuracy in context recognition as it impacts delays. Compared to previous state of the art, the proposed solution showed an improvement of 64% in energy reduction and 34% faster detection in state change. We were able to reduce the number of computations needed to find the optimal parameters for the Viterbi algorithm. Moreover, we showed the adaptation of the method to different state survival probabilities and the importance for accurately represent the user behavior. We acknowledge some limitations in our work. The need for a large dataset of historical time durations spent in contexts, which need to be collected over a long period of time, impact the immediate applicability of the method. That can be addressed by future work with Reinforcement Learning which can be explored for continual learning as user behavior changes. Moreover, for future work, the problem can be extended to find the group of sensors that can, not only trade off energy and delay, but also achieve a balance with best accuracy in multi-context recognition.

Bibliography

[1] S. Taleb, H. Hajj, and Z. Dawy, "Vcams: Viterbi-based context aware mobile sensing to trade-off energy and delay," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 225–242, 2017.

[2] D. Craig, "Cognitive prosthetics in alzheimer's disease: A trial of a novel cell phoned-based reminding system," *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, vol. 6, no. 4, p. S173, 2010.

[3] M. Skubic, "A ubiquitous sensing environment to detect functional changes in assisted living apartments: The tiger place experience," *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, vol. 6, no. 4, p. S173, 2010.

[4] O. D. Lara and M. A. Labrador, "A mobile platform for real-time human activity recognition," in *2012 IEEE consumer communications and networking conference (CCNC)*, pp. 667–671, IEEE, 2012.

[5] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme–injecting sensing presence into social networking applications," in *European Conference on Smart Sensing and Context*, pp. 1–28, Springer, 2007.

[6] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," 2006.

[7] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 119–134, 2007.

[8] E. Jovanov, A. Lords, D. Raskovic, P. G. Cox, R. Adhami, and F. Andrasik, "Stress monitoring using a distributed wireless intelligent sensor system," *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 3, pp. 49–55, 2003.

[9] R. W. Picard, E. Vyzas, and J. Healey, "Toward machine emotional intelligence: Analysis of affective physiological state," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 1175–1191, 2001.

[10] L. Ardito, G. Procaccianti, M. Torchiano, and G. Migliore, "Profiling power consumption on mobile devices," *ENERGY*, pp. 101–106, 2013.

[11] K. Nishihara, K. Ishizaka, and J. Sakai, "Power saving in mobile devices using context-aware resource control," in *2010 First International Conference on Networking and Computing*, pp. 220–226, IEEE, 2010.

[12] V. Pejovic and M. Musolesi, "Anticipatory mobile computing for behaviour change interventions," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pp. 1025–1034, ACM, 2014.

[13] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pp. 179–192, ACM, 2009.

[14] K. K. Rachuri, C. Mascolo, and M. Musolesi, "Energy-accuracy trade-offs of sensor sampling in smart phone based sensing systems," in *Mobile Context Awareness*, pp. 65–76, Springer, 2012.

[15] S. Taleb, H. Hajj, and Z. Dawy, "Ego: Optimized sensor selection for multi-context aware applications with an ontology for recognition models," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2518–2535, 2018.

[16] R. Pérez-Torres, C. Torres-Huitzil, and H. Galeana-Zapién, "Power management techniques in smartphone-based mobility sensing systems: A survey," *Pervasive and Mobile Computing*, vol. 31, pp. 1–21, 2016.

[17] T. Rault, A. Bouabdallah, Y. Challal, and F. Marin, "A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications," *Pervasive and Mobile Computing*, vol. 37, pp. 23–44, 2017.

[18] Ö. Yürür, C. H. Liu, Z. Sheng, V. C. Leung, W. Moreno, and K. K. Leung, "Context-awareness for mobile sensing: A survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 68–93, 2014.

[19] S. Taleb, N. Abbas, H. Hajj, and Z. Dawy, "On sensor selection in mobile devices based on energy, application accuracy, and context metrics," in *2013 Third International Conference on Communications and Information Technology (ICCIT)*, pp. 12–16, IEEE, 2013.

[20] L. Gao, A. K. Bourke, and J. Nelson, "Activity recognition using dynamic multiple sensor fusion in body sensor networks," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1077–1080, IEEE, 2012.

[21] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, "Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pp. 267–280, ACM, 2008.

[22] P. Zappi, D. Roggen, E. Farella, G. Tröster, and L. Benini, "Network-level power-performance trade-off in wearable activity recognition: a dynamic sensor selection approach," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 11, no. 3, p. 68, 2012.

[23] D. Gordon, J. Czerny, T. Miyaki, and M. Beigl, "Energy-efficient activity recognition using prediction," in *2012 16th International Symposium on Wearable Computers*, pp. 29–36, IEEE, 2012.

[24] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *Proceedings of the 17th annual international conference on Mobile computing and networking*, pp. 73–84, ACM, 2011.

[25] O. Yurur, M. Labrador, and W. Moreno, "Adaptive and energy efficient context representation framework in mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1681–1693, 2013.

[26] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with senseless," in *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pp. 61–62, ACM, 2009.

[27] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pp. 71–84, ACM, 2010.

[28] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Smartdc: Mobility prediction-based adaptive duty cycling for everyday location monitoring," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 512–525, 2013.

[29] Y. Lee, C. Min, Y. Ju, S. Kang, Y. Rhee, and J. Song, "An active resource orchestration framework for pan-scale, sensor-rich environments," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 596–610, 2013.

[30] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 299–314, ACM, 2010.

[31] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Aaai*, vol. 5, pp. 1541–1546, 2005.

[32] M. Ehrgott, “A discussion of scalarization techniques for multiple objective integer programming,” *Annals of Operations Research*, vol. 147, no. 1, pp. 343–360, 2006.

[33] M. R. Garey and D. S. Johnson, *Computers and intractability*, vol. 174. freeman San Francisco, 1979.

[34] C. Elsido, A. Bischi, P. Silva, and E. Martelli, “Two-stage minlp algorithm for the optimal synthesis and design of networks of chp units,” *Energy*, vol. 121, pp. 403–426, 2017.

[35] Z. Ma and A. W. Krings, “Survival analysis approach to reliability, survivability and prognostics and health management (phm),” in *2008 IEEE Aerospace Conference*, pp. 1–20, IEEE, 2008.

[36] A. J. Viterbi, *Viterbi Algorithm.* 04 2003.

[37] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, “Activity classification using realistic data from wearable sensors,” *IEEE Transactions on information technology in biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[38] D. Chu, N. D. Lane, T. T.-T. Lai, C. Pang, X. Meng, Q. Guo, F. Li, and F. Zhao, “Balancing energy, latency and accuracy for mobile sensor data classification,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 54–67, ACM, 2011.

[39] W.-J. Yi, O. Sarkar, S. Mathavan, and J. Saniie, “Wearable sensor data fusion for remote health assessment and fall detection,” in *IEEE International Conference on Electro/Information Technology*, pp. 303–307, IEEE, 2014.

[40] S. Khoor, K. Nieberl, K. Fugedi, and E. Kail, “Telemedicine ecg-telemetry with bluetooth technology,” in *Computers in Cardiology 2001. Vol. 28 (Cat. No. 01CH37287)*, pp. 585–588, IEEE, 2001.

[41] V. Servizi, F. C. Pereira, M. K. Anderson, and O. A. Nielsen, “Mining user behaviour from smartphone data, a literature review,” *arXiv preprint arXiv:1912.11259*, 2019.