Pers Ubiquit Comput (2014) 18:205–221
DOI 10.1007/s00779-013-0638-2

ORIGINAL ARTICLE

Activity recognition for creatures of habit

Energy-efficient embedded classification using prediction

Dawud Gordon · Jürgen Czerny · Michael Beigl

Received: 2 October 2012 / Accepted: 21 November 2012 / Published online: 3 March 2013
© Springer-Verlag London 2013

Abstract Energy storage is quickly becoming the limiting factor in mobile pervasive technology. We introduce a novel method for activity recognition which leverages the predictability of human behavior to conserve energy by dynamically selecting sensors. We further present a taxonomy of existing approaches to dynamically reducing consumption while maintaining recognition rates. The novel algorithm conserves energy by quantifying activity-sensor dependencies and using prediction methods to identify likely future activities. The approach is implemented and simulated using two activity recognition data sets, and the effects of the novel method are evaluated in terms of recognition rates, energy consumption, and prediction rates. The results indicate that switching off sensors only significantly affects prediction under extreme conditions and that these effects can be counteracted by adjusting system parameters. Large savings in energy can be achieved at very low cost, for example, recognition losses of 1.5 pp with 84.8 % energy savings for the first data set, and 2.8 pp and 89.9 % for the second.

1 Introduction

As concepts from pervasive and mobile computing become more mainstream, the community seeks practical approaches for realizing pervasive technology. Situation, context, or activity recognition techniques provide a method for machines to recognize human and social situations, allowing them to act proactively without contradicting or offending their users. Modern devices such as smart phones or wireless sensor nodes are now able to support these algorithms [28] as processing power and memory improve over time according to Moore's Law.

Energy storage in such devices is not subject to the same doubling effects and is quickly becoming the limiting factor in pervasive technology. This can be seen clearly when reviewing the battery lifetimes for mobile phones over the past 10 years. The cost of communication in terms of energy consumption is another factor which does not scale according to Moore's Law, indicating that for intelligent wearable applications to be practical, methods for low-power situation recognition must be embedded in mobile devices.

Embedded classification for mobile devices is not a new concept and goes as far back as 1997 [8], where Bouten et al. used simple signal processing techniques to measure activity levels of users wearing a mobile device. Several methods for low-power embedded context classification have been introduced in the activity and context recognition community. Cacmakci et al. [9] and Stäger et al. [27] introduce straight-forward approaches to low-power recognition of contexts and activities in embedded systems using inertial or audio sensors, respectively. Krause et al. [17] propose to dynamically reduce sensor sampling rates to conserve energy, thereby greatly increasing battery lifetime. A similar approach was presented by Sun et al. [29] where coarse-grained activity levels were locally recognized to adjust the sensor sample rate. Benbasat et al. [2] introduce a method for conserving energy in a system with redundant sensors which are switched on and off dynamically based on the level of activity currently measured. Roy et al. [25] use sensor configurations which are selected for specific activities based on the minimum requirements of an application. One cumulative result of

D. Gordon (✉) · J. Czerny · M. Beigl
Karlsruhe Institute of Technology (KIT),
Vincenz-Prießnitz-Strasse 1, Karlsruhe, Germany
e-mail: Dawud.Gordon@kit.edu

the research mentioned is that there is always a trade-off between how well activities can be recognized and how much energy it costs to do it [6, 28].

We conducted a survey of work done in this field (see Sect. 2) revealing that focus is mostly on motion-based sensors, with the accelerometer being the most popular sensing modality. The survey also shows that initially, research was focused on custom hardware and sensor network platforms, but recently it has shifted toward mobile phones. Early methods for low-power recognition began with systems engineering, but more recently, dynamic sensor selection and sample rate reduction have been the tools of choice. These dynamic approaches often use the current activity or activity level as an indicator of the optimal current configuration, or wake to more expensive system states hierarchically to conserve energy.

We present a different, novel method for using context (or activity) prediction to further conserve energy. Many things we do have a certain repetitiveness or periodicity about them [31] and are therefore predictable to a certain extent. This information can also be used to improve recognition abilities [22]. Context prediction is the process of using a context history to predict contexts, situations, or activities which will occur in the future [20]. This can be done at several different abstraction levels [26], ranging from extrapolating raw sensor data into the future, to predicting abstract concepts such as activities. We propose that it can be used to reduce the power consumption of the recognizing device as well.

The idea is simple. Traditionally, all sensors are used constantly even though certain sensors may only be necessary to detect specific activities. As a result, energy may be wasted when sensors are enabled which are not necessary to detect the current activities. Given a scenario where activities are performed in a manner which is predictable, probable future activities can be forecast. Sensors which are not needed to decipher probable activities from each other can be turned off (or the sample rate reduced), conserving energy without greatly impacting recognition rates. The risk is that incorrect predictions cause sub-optimal sensor configurations, further leading to incorrect recognition and prediction.

In this work, we propose that by leveraging the predictability of human actions, it is possible to tip the balance of the energy/recognition trade-off to conserve energy resources. We proposed this concept in a poster [15] and evaluated it initially on a single data set [11]. Since the initial publication, other related research in the field has emerged [33], but an exhaustive evaluation of system behavior as presented here has not yet been conducted.

The performance is simulated using two preexisting activity recognition data sets [14, 24], where artificial data sets are generated from these sets in order to evaluate different scenario parameters. We evaluated the algorithms in terms of activity recognition rates, energy savings achieved, and the prediction accuracy with respect to system parameters. The results indicate that the novel approach allows for application and scenario-specific selection of the recognition/energy trade-off, producing large energy savings, even for small recognition losses (e.g., recognition losses of 1.5 pp with 84.8 % energy savings for first [14], and 2.8 pp and 89.9 % for the second data set [24]).

This article is structured as follows. A survey of research conducted toward reducing the energy costs of embedded recognition is presented in Sect. 2. In Sect. 3, the proposed method and algorithm is presented, including the context recognition, prediction, and sensor selection processes. The experiment implementation and simulation environment is presented in Sect. 4, along with a description of the data sets used and their preparation. Section 5 contains the results of the simulation with respect to energy consumption, classification, and prediction values, the implications of which are discussed in Sect. 6. Finally, the article is concluded in Sect. 7.

2 A survey of energy-efficient recognition

For embedded and mobile systems, power consumption is one of the most critical attributes [17]. We surveyed system approaches for reducing the power consumption footprint of online, embedded activity recognition in order to generate on overview of this field. To our knowledge, no such survey of applications and attributes has yet been conducted making this a novel contribution of this work.

The survey was conducted based on following parameters which where deemed to be of importance for understanding the breadth of the research. Motivation for these parameters is taken from related work, although the parameters do not cover the entire design space for such systems.

- *Platform* This parameter describes the hardware platform used for recognition. This information is of importance as it gives the reader an indication of the amount of resources which are available for embedded recognition. For example, embedded recognition on a mobile smart phone [3] probably has an order of magnitude more processing power and memory than an embedded wrist watch platform [17].
- *Sensing modality* The sensors used for an application give an indicator of the order of the problem. Sensors have different properties, for example, an embedded accelerometer has a far shorter startup time and power consumption [34] compared to a GPS receiver [21].

- *Conservation approach* There are several different approaches to the problem of energy conservation with different affects on other components of the system. In some of the approaches taken to reducing power consumption, design choices are empirically explored to find systems which consume less energy. Other approaches involve designing dynamic systems which adjust themselves based on the current situation to minimize energy consumption without violating some quality criteria.
- *Control method* of the aforementioned conservation approaches, several of them dynamically optimize certain parameters, for example, sensor sample rate, sensor selection, or execution mode. In order to perform these operations, the decision process requires some type of input in order to close the control loop. Here, the type of input used to control the conservation approach is surveyed.
- *Recognition algorithm* Different algorithms are equipped to specific degrees for certain problems, therefore posing advantages in certain situations. Each algorithm is in itself a trade-off between accuracy and processing power. Certain types can handle missing features and sensors intrinsically such as nearest neighbors approaches and Bayesian inference [11] while others such as neural network approaches and decision trees must be specifically adapted for such issues [3, 29]. The selection of a classifier algorithm therefore provides insight into the energy/recognition trade-off conducted in the work.
- *Application* which contexts or activities are recognized greatly changes the applicability and general usefulness of the approach. Algorithmic evaluations are also quite specific to the application domain. It is therefore important to note in which domain the research was conducted in order to be able to estimate usefulness in other areas.
- *Reproducibility* one of the major issues which we see in this field is the reproducibility of results. While methodologies and algorithms may be well defined and formalized, re-implementation is time consuming and effort intensive. A system is considered reproducible if either (1) the hardware platform is available for purchase and the code basis is published, or (2) the data set on which the evaluation was conducted is publicly available.

2.1 Physical attributes

In Table 1, the surveyed works are listed with respect to their technical details. As indicated there, earlier platforms were often custom-built proprietary sensor nodes specially designed for the recognition operation [4, 5, 8, 9, 28], probably due to the unavailability of standardized sensing devices. Although some more recent research projects also incorporated some custom hardware [19, 23, 30, 33], a trend can be seen toward the use of mobile phones [3, 10, 21, 29, 32, 34]. Combinations of devices have also been used, where mobile phones are selected along with customized hardware for the recognition task [10, 15, 19, 23, 30].

When observing the different sensing modalities surveyed, it quickly becomes apparent that the accelerometer is the most popular sensor used. This is not surprising as embedded activity recognition has a large overlap with the wearable sensing community, where sensing motion provides great insight [3]. Often, the accelerometer was used alone [1, 3, 8, 9, 10, 15, 32, 34] to recognize physical activities (see Table 2). Other times, it was combined with other modalities to better capture physical signals [2, 4, 5, 11, 17, 23, 25, 28, 29, 30]. Accelerometers are also used to incorporate physical sensing modalities into other types of recognition systems, for example, video [33] or location systems [21]. The sensors used are dependent on the application, that is, the activities or contexts which were recognized, where the effectiveness of the sensing modality is given by the influence of the activity on that sensor.

2.2 Recognition attributes

Early systems were exploratory in nature and investigated the performance of recognition under constrained resources. Here, the focus was on system design, where the hardware/software architecture was constructed in such a way as to maintain low consumption using standard algorithms with little or no adaptation at run time [4, 5, 8, 9, 28].

The effects of reducing sensor sample rate to a reduced but constant value throughout operation with respect to energy consumption and recognition loss indicate that for certain applications it can be advantageous [17]. A further improvement can be achieved by adapting the sample rate of the sensors which reduces their consumption. Adaptive sampling, however, requires a control parameter to set the sample rate correctly so as not to cause deterioration in recognition rates. In the surveyed work, this has been done using some indicator of the current system state, such as an activity level indicator [2, 29], or an actual recognized activity [17, 21, 30, 32, 34] or location [18], or a combination of those [21]. This method is dependent on the sensor modalities (see Table 1) with respect to the warm-up times and power consumptions for effectiveness.

A second method for reducing energy is to dynamically select sensors to be activated or deactivated during a certain period of time [1]. Here again, a method for selecting sensors for each classification is needed, where Au et al.

Table 1 Technical details of work surveyed in embedded context and activity recognition

References	Year	Sensor modality	Platform
Bouten et al. [8]	1997	Accelerometer	Sensor node (proprietary)
Cakmakci et al. [9]	2002	Accelerometer	SoundButton sensor node (proprietary)
Bharatula et al. [5, 4]	2005	Accelerometer, light, microphone	Sensor node (proprietary)
Krause et al. [17]	2005	Accelerometer, microphone, light, temperature	eWatch wearable platform
Benbasat and Paradiso [2]	2007	Accelerometer, gyro, tilt switch	Gait shoe (proprietary)
Stäger et al. [28]	2007	Microphone, accelerometer, light	Sensor node (proprietary)
Thatte et al. [30]	2010	ECG, accelerometer (wireless)	Smart phone, wireless sensors
Berchtold et al. [3]	2010	Accelerometer	Smart phone
Raffa et al. [23]	2010	Accelerometer, gyroscope	Smart phone, ContextWatch (proprietary)
Lin [18]	2010	GPS, WLAN, bluetooth, cell	Smart phone
Paek et al. [21]	2010	GPS, WLAN, bluetooth, cell, accelerometer	Smart phone
Roy et al. [25]	2011	Accelerometer, gyroscope, light, temperature	SunSPOT sensor node
Sun et al. [29]	2011	ECG, accelerometer	Smart phone
Lu et al. [19]	2011	Microphones (internal/external)	Smart phone, sensor node (proprietary)
Gordon et al. [13]	2012	Accelerometer (wireless)	Smart phone, JenniSense sensor node
Wood et al. [33]	2012	Camera, accelerometer	Wearable camera (proprietary)
Au et al. [1]	2012	Accelerometer (wireless)	MicroLeap wearable platform
Yan et al. [34]	2012	Accelerometer	Smart phone
Gao et al. [10]	2012	Accelerometer (wireless)	Smart phone, Shimmer sensor node
Wang et al. [32]	2012	Accelerometer	Smart phone
Gordon et al. [11]	2012	Accelerometer, MVS [14], light, temp	Sensor node (proprietary)

propose using the last recognized activity to conduct this. Theoretically, once the sensors have been activated, there is no reason why methods of adaptive sampling cannot be applied, although this has not been evaluated in the work surveyed.

Another method for conserving energy is to break the recognition process down into a hierarchical pipeline, where each level activates higher-order processes under certain conditions, thus avoiding superfluous operations if lower levels deem them unnecessary. These methods can range from activating the system based on activity indicators, that is, (e.g. activity level) for waking up activity recognition [2, 23], modular classification systems [3], movement detection for activation of more expensive location sensors [21], or sensing the presence of voices for speaker identification [19].

In Table 1, some instances employ wireless sensors which seem to indicate a distributed approach as apposed to an embedded one [1, 15, 30]. In these works, the authors observed the systems as being closed, that is, the energy consumption of both classification and sensor usage was investigated, and are therefore relevant to this survey. The methods then used for energy consumption optimization must account for the same problems as embedded systems and therefore employ the same techniques, for example, adaptive sample rates [30] or sensor selection [1]. One difference is that the volume of sensor data has a great affect on system consumption, which can be addressed using preprocessing [10, 12, 13].

2.3 Summary of the survey

To summarize the survey, when designing low-power embedded recognition applications, the system should be designed to reduce overall consumption by using low-power components and algorithms. Once this has been accomplished, further savings can be achieved by making the system dynamic in nature to adapt to changing requirements. Here, there are 3 methods for energy conservation which have been used. First, the sample rates of the sensors can be dynamically adapted to current requirements, where reducing the sample rate also reduces energy consumption. Second, the designer can opt to turn off sensors entirely for a sample period, further reducing energy consumption but risking deterioration of recognition values if not done correctly.

Third, a method for structuring recognition components can be implemented in a hierarchical way, such that low-power components wake those with higher consumption only when they are needed. Often times, the third method benefits from dedicated hardware components which conduct low-power listening for activity cues. These three

Table 2 Recognition approaches of work surveyed in embedded context and activity recognition

References	Recognition algorithm	Conservation approach	Control method	Application	Reproducible
Bouten et al. [8]	Correlation analysis	Low-power design	None, static	Energy expenditure	No
Cakmakci et al. [9]	Bayesian inference	Low-power design	None, static	Basic physical activities	Yes (code)
Bharatula et al. [5, 4]	Decision tree	Low-power design	None, static	Office activities	No
Krause et al. [17]	SVM	Adaptive sample rate	Current activity	Basic physical activities	No
Benbasat and Paradiso [2]	CART decision tree	Adaptive sample rate, hierarchical wake up	Activity level	Wearable gait monitoring, animal monitoring	No
Stäger et al. [28]	C4.5 decision tree	Adaptive sample rate	None, static	Kitchen activities	No
Thatte et al. [30]	SVM, Bayesian inference	Adaptive sample rate	Current activity	Basic physical activities	No
Berchtold et al. [3]	Fuzzy inference	Modular classifier pipeline	Current activity	Basic physical activities	No
Raffa et al. [23]	HMM	Hierarchical pipeline	Activity level	Gesture recognition	No
Lin [18]	Bayesian estimation	Adaptive sample rate	Current location, accuracy requirement	Location	No
Paek et al. [21]	Onset detection	Adaptive sample rate	Location	Location	No
Roy et al. [25]	Classifier independent	Adaptive sample rate, sensor selection	None, static	Basic physical activities	No
Sun et al. [29]	Decision tree	Adaptive sample rate	Activity level	Basic physical activities	No
Lu et al. [19]	Bayesian inference	Hierarchical wake up	Speech (activity) level	Speaker identification	No
Gordon et al. [13]	DT, nB, kNN	Data preprocessing	None, static	Group activities	Yes (data)
Wood et al. [33]	K-Means	Adaptive sample rate	Predicted future activities	Sleep, office, cycling	No
Au et al. [1]	HMM	Adaptive sensor selection	Current activity	Basic physical activities	No
Yan et al. [34]	Bayesian inference	Adaptive sample rate	Current activity	Basic physical activities	No
Gao et al. [10]	nB, DT	Sensor selection	Current activity	Basic physical activities	No
Wang et al. [32]	Semi-Markov process estimation	Adaptive sample rate	Current activity	Movement detection	Partly (data)
Current Work [11]	HMM, kNN	Adaptive sensor selection	Predicted future activities	Basic physical activities	Yes (data)

methods can also be combined with each other to further improve consumption rates. Each of these methods has different advantages and disadvantages under different conditions and scenarios. Unfortunately, based on previous publications, it is not possible to make comparative statements about the energy consumption and recognition rates for these methods due to reproducibility factors.

Once these design choices have been made, a method for controlling the dynamic conservation approach can be selected. The first method is to use a general indicator of the activity level to control the conservation approach. An activity indicator is especially effective when combined with a hierarchical wake-up pipeline due to the low computational complexity of these indicators [19]. Another method is to use the current activity which has been recognized to adapt the system to the requirements of recognizing that activity. Here, the risk is that since an activity must first be detected before adaptation, the system has issues with detecting activity transitions. This design decision structure can be represented as a taxonomy of approaches to embedded recognition which is shown in Fig. 1. The static system control mode has been deprecated to indicate that it is not the focus of this survey and is therefore not exhaustive.

In this work, we examine a new method for controlling the energy-conserving mechanism. The method is not specific to the energy conservation approach and can be combined with adaptive sampling, sensor selection, or a pipelined activity recognition chain [11, 15, 33], but is evaluated here with a sensor selection approach. Using context prediction, a future probability distribution can be generated which allows the system to be proactive in nature [26], instead of only reacting to the current system state. Using future activities eliminates the lag incurred by

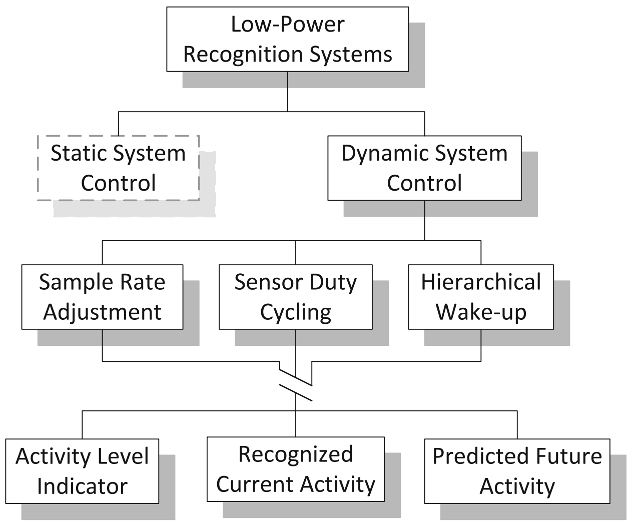

Fig. 1 A taxonomy for low-power, embedded activity recognition

having to recognize the current activity state or level [26] before being able to react to it. This can improve the power consumption footprint and the correctness of the recognition during the lag, or activity transition period. However, incorrect predictions may lead to mis-configurations, a research question which is evaluated in the rest of this work.

3 Proposed algorithmic approach

The standard process for activity recognition using machine learning algorithms is straightforward. Sensors are sampled in parallel at an arbitrary but constant rate for a period of time. The data are then saved as a discrete multidimensional vector, referred to as a sample window. This window is processed using different algorithms to generate signal features, for example, standard deviation, average, FFT, or cepstral coefficients. Which features are used depends on the application (i.e., which activities we want to recognize), and the type of sensors being used, and are referred to all together as a feature vector. A machine learning algorithm is given the task of learning to recognize which activity was occurring during the sample window, based on its feature vector.

We propose integrating prediction into the process to improve energy consumption as demonstrated in Fig. 2. First, *activated* sensors are sampled to generate a sample window. The sample window is then processed into a feature vector and *classified* as to which activity is being performed. Based on the classification history, future activities which are likely to occur are *predicted*. An appropriate sensor configuration is then activated to distinguish only the *likely activities*, and the process repeats itself.

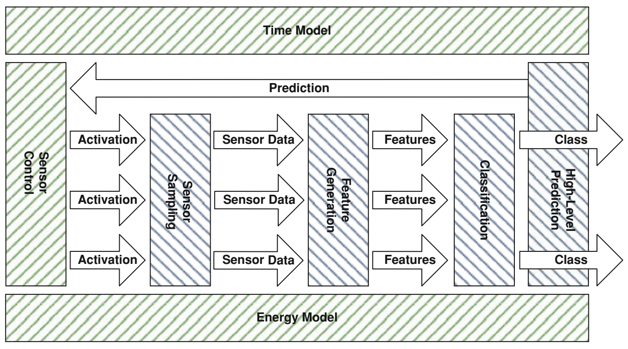

Fig. 2 The novel algorithm (*backslash*) and simulation environment (*slash*)

During the course of this research, we identified three parameters which affect the trade-off between energy and recognition. The first is the **predictability** κ of the sequence of activities, or the inherent predictability of the scenario itself. A low value for κ indicates that prediction results are little better than random, where a $\kappa = 1$ indicates a 100 % prediction accuracy. In real-world scenarios, κ simply equates to the prediction rate for a given predictor and scenario. This parameter cannot be influenced by the designer and can only be quantified by analyzing the scenario and predictor beforehand. The second parameter affecting performance is $\boldsymbol{\rho}$, **the number of classes which are predicted** at each time step. The more classes which are predicted, the better the chance that the next class is actually among the predicted classes (correct prediction), but the lower the savings will be as the system accounts for more possible activities. Therefore, ρ specifies the level of risk, which allows the designer to tip the odds toward recognition or energy as will be seen later. The third parameter is application specific and is referred to as the **loss parameter** $\boldsymbol{\lambda}$, which specifies the amount of recognition which can be sacrificed in order to conserve energy without breaking the application's requirements. A λ value of 0 indicates that optimizations causing any loss at all, however, minimal, are not acceptable, and $\lambda = 1$ means energy savings are of the utmost priority, and recognition rates are of no importance.

A useful analogy at each prediction/classification step is that of a wager. Here, κ, the predictability of the scenario, can be thought of as the probable outcome of the bet based on previous experience (prior distribution). The number of classes predicted, ρ, allows ρ different outcomes to be bet on at once: The higher ρ is, the better the chances of a correct bet, but the lower the payout in terms of energy saved. In this case, the wager λ is a specified amount of the total recognition rate, and the payout is in energy savings. Losing a bet (meaning a false prediction) is detrimental to classification lowering overall recognition rates.

3.1 Weighting sensors to activities

Here, we will present the method for selecting which sensors to activate based on predicted activities. When observing the chain of events in the activity classification process, each feature in the set of features used $f \in F$ is implicitly mapped onto a single sensor in the set of sensors $s \in S$. That sensor generates the data for this feature, producing the surjective mapping a of features onto sensors $a\colon F \rightarrow S$. Reversely, each sensor s_i is then "responsible" for a subset of features $\widetilde{F}_{s_i} \in F$, meaning the features in $\widetilde{F}_{s_i}$ are generated over data stream from sensor s_i.

Mapping activity classes onto the sensors over the features is not as simple. This mapping cannot be carried out independent of the classifying algorithm, as each algorithm has a different method of measuring the distance between two vectors. For example, nearest neighbors algorithms use a multi-dimensional distance measurement, often euclidean distance, between two vectors to separate them, probability-based models calculate where a vector lies in the probability distribution for a specific class, and decision trees often use entropy as an indicator of distance [7]. An overview of selecting features which best suite an embedded application is presented by Könönen et al. [16], providing a *sensor to application* mapping. While these algorithms potentially improve the quality of classification and reduce the computational load, they do not provide a mapping of features to classes by relevance or importance. A method for generating a *sensor to class (activity)* mapping by relevance, or importance was proposed by Roy et al. [25], which they referred to as quality of inference (QoINF)). As will be discussed later, this method is not effective for the approach and data set presented here.

Turning sensors on and off will result in a dynamic feature vector length, and for this reason, we will consider standard classifiers which can natively support this. Specifically, nearest neighbor classifiers are well suited to this task as omitting a feature represents a dimensional reduction of the labeled training vector space, and the missing features are simply excluded from the distance calculation. Probabilistic models are also well suited as the observational distributions for missing variables can be marginalized when calculating the probabilities of the hidden states. Both examples lose only the information that would have been gained from the missing features, but are not negatively affected further [7].

In order to generate the weighted mapping Q (the weight is the dependency of activities on sensors), training data are gathered for each class. Weight calculation was done by testing the trained classifier against all training vectors for each class and simulating different feature combinations. Selected features were turned off, and the dependency of each class on those features was evaluated. The degree of dependency is the drop in accuracy compared to the full feature vector: A large drop in recognition indicates a high dependency, a small drop, a low dependency.

Initially, the intent was to only evaluate the weight for each feature individually. The cost/dependency weights for a sensor could then be calculated by summation of the weights of its features, assuming that the cost of turning off two sensors is the cost of the one plus the cost of the other, or that $q_{\mathrm{cf}_i} + q_{\mathrm{cf}_j} \approx q_{\mathrm{cf}_{ij}}$ as indicated by Roy et al. [25]. This, however, proved to be too inaccurate to be useful due to the conditional dependence of features and sensors, making $q_{\mathrm{cf}_i} + q_{\mathrm{cf}_j} \le q_{\mathrm{cf}_{ij}}$ [16]. Therefore, Q values were calculated for each class against all possible sensor subsets directly, instead of by summing single feature or sensor values.

In order to correctly estimate the optimal sensor subset $\widetilde{S}$ for a sensing and classification step, the matrix Q must be calculated only once at training time. The resulting mappings can be seen in Fig. 3, showing one mapping of class $c_2 \in C$ onto sensors $\widetilde{S}_{1,3} \in S$ over features generated from $\widetilde{S}_{1,3}$. Each mapping in $b\colon C \xrightarrow{Q} F$ represents one element in the Q matrix, in this case $q_{c_2\widetilde{S}_{1,3}} \in Q$. The Q matrix is indexed by the power set of S without the empty set, or $\widetilde{S} \in \wp(S)\backslash\{\}$, and the classes $c \in C$, resulting in a $|C| \times (2^{|S|} - 1)$ matrix. The value at each point i, j indexed by c_i and $\widetilde{S}_j$ is the recognition loss when classifying c_i using sensor subset $\widetilde{S}_j$ compared to using all sensors S over a set of evaluation data samples. Now, for each class $c_i \in \widetilde{C}_{t+1}$ where $\widetilde{C}_{t+1}$ is the set of activities predicted to occur at the next time step, a set of sensors $\widetilde{S}_{t+1}$ can be identified which is optimal with respect to λ. This is accomplished by selecting the sensor subset $\widetilde{S}$ for the next period $t + 1$ such that it fulfills Eq. (1).

$$\widetilde{S}_{t+1} = \underset{\mathrm{En}(\widetilde{S})}{\arg\min}, \forall_{c \in \widetilde{C}_{t+1}} q_{\widetilde{S},c} \le \lambda \tag{1}$$

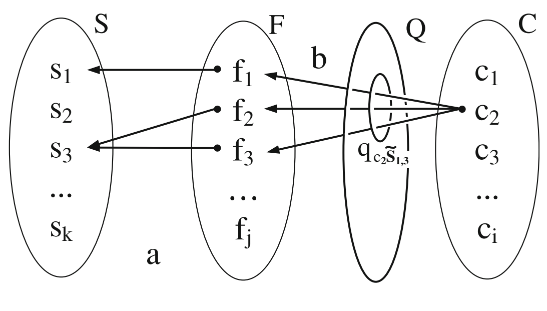

Fig. 3 Features (F) from sensors (S) s_1 and s_3 have value (Q) $\mathbf{q}_{c_2\widetilde{S}_{1,3}}$ for class (C) c_2

Where $\mathrm{En}(\widetilde{S})$ is the combined energy consumption of all sensors in $\widetilde{S}$. Simply put, in order to distinguish the classes predicted to occur $\widetilde{C}_{t+1}$ from each other, the sensor configuration $\widetilde{S}$ is selected which saves the most energy $\mathrm{En}(\widetilde{S})$ without violating the acceptable loss parameter $q_{\widetilde{S},c} \leq \lambda$ for *any* of the predicted activities $c \in \widetilde{C}_{t+1}$. This selects the sensor configuration with the lowest energy consumption that is still capable of recognizing the predicted classes with acceptable recognition rates. The next section will analyze the use of context prediction to generate a set of classes which are likely to appear in the next sample window ($\widetilde{C}_{t+1}$).

3.2 Context prediction

Context prediction is used to estimate a subset of all contexts or activities $\widetilde{C}_{t+1} \in C$ which are most likely to occur at the next time step $t+1$. The cardinality of $|\widetilde{C}_{t+1}| = \rho$ is a parameter which can be adjusted and allows the designer to select the recognition accuracy risk against the energy reward as will be shown in Sect. 5. This approach is independent of the algorithm or abstraction level used for prediction. Important is only the quantification of the predictability parameter κ which is simply an indicator of how well the predictor is able to forecast the given scenario (predictor accuracy). The results presented here should therefore still apply for all scenarios and prediction algorithms.

As indicated by Fig. 2, high-level context information at the activity or context abstraction level is used for prediction. Using low-level, sensory or feature data is also an option, but high-level prediction reduces complexity in terms of training and execution [26]. The algorithm used for prediction is a first-order Markov chain consisting of states $c \in C$. At each time step, the probability $P(c_{i,t+1}|c_t)$ for each $c_i \in C$ is calculated, and the ρ states with the highest probabilities are output as predictions.

4 Implementation and simulation

This section presents the algorithmic implementation and the simulation environment. Both were programmed using the Python programming language.

4.1 Simulation environment

The main concept is to leverage the predictability of human actions in order to conserve a large amount of energy while only sacrificing a small amount of recognition capabilities. The simulation environment was designed to evaluate the method for various degrees of predictability κ. Two published data sets which are publicly available were used in this evaluation.

The data sets were selected because both of them are publicly available sets of numerical data gathered from wearable sensing modalities, making the results presented here easier to reproduce. The MVS data set contains a relatively large number of activities, but relatively few sensor modalities. The OPP data set contains relatively few locomotion activities, but with a large number of sensing modalities and locations. The goal was to select data sets which complement each other so as to demonstrate different performance aspects of the proposed approach under different conditions.

4.1.1 Micro-vibration sensor data set (MVS)

The first data set used for evaluation [14] contains 142 minutes of data from 4 sensors (see Table 3), sampled from 5 subjects performing 8 activities (taking a bus, riding a bike, walking, jogging, taking the elevator, typing at a desk, going up/down stairs, and standing).

4.1.2 The opportunity data set (OPP)

The second data set used for the evaluation [24] contains information from 72 sensors over 12 subjects, yielding 25

Table 3 Energy consumption rates for the simulated hardware components

Element		Dimensions	Energy cost (mW)		Data set	
Function	Name		Online	Offline	MVS [14]	OPP [24]
Light	APDS-9003	1	8.25	0.0	√	
Temperature	TC1047	1	0.1155	0.0	√	
Vibration	MVS 0608.02	1	0.0015	0.0	√	
Microprocessor	PIC18LF14K	1	0.0512	0.0	√	√
Acceleration	ADXL335	3	1.4	0.0	√	√
Magnetic field	HMC 5883L	3	0.33	0.0066		√
Gyroscope	ITG-3200	3	21.45	0.0165		√

hours of data. In order to reduce complexity of the simulation, a subset of the sensors was used. The sensors used were the following: acceleration on the hip, right knee, back, right hand, and left hand; gyroscope and magnetic field sensors on the back. The data set contains a myriad of labels, including locomotion modes, object interactions, interaction types, and which hand was used. For this evaluation, the locomotion mode labels were used, consisting of 4 activities (walk, stand, lie, and sit). The system was simulated as a single device connected to all sensors. While the capability of an embedded processor to handle this number of input streams or data is questionable, this does not affect the results of the evaluation as the processor is modeled as being always on with constant consumption. The energy consumption values were taken from a standard microprocessor and sensors, identical to that of the MVS data set (see Table 3). Actual device specifications and consumption values are not included in the OPP data set.

Both data sets were evaluated using the same preprocessing framework. The system simulates real time through a replay mechanism using the recorded data. The respective data set is cut up into one-second windows without overlap, over which features are generated. The resulting feature vectors are then fed to the novel algorithms as if they were being generated in real time. The sensor configurations are simulated, where for a specific sensor configuration $\widetilde{S}$, the features $\widetilde{F}_{\widetilde{S}}$ are present in the feature vector and all others are omitted. Once a sensor configuration $\widetilde{S}$ has been selected, the features $\widetilde{F}_{\widetilde{S}}$ are calculated. Per sensor, the following features are calculated [14]: average, standard deviation, area under the curve, min-max difference, Shannon entropy, and FFT peak.

The energy consumed by the device $\mathrm{En}(\widetilde{S})$ is recorded for the time step. The total consumption consists of the consumption of each sensor, as well as the energy consumption of the microprocessor during the course of the sample window, or one second. The energy model is simplistic, ramp-up and ramp-down times/consumptions of the sensors are not modeled, and the processor consumption is modeled as being constant regardless of load. This approximation does not account for the added load of prediction, but the method used here has a computational complexity of only $\mathcal{O}(|C|)$ [15]. The energy consumption rates for each device simulated can be found in Table 3. At each time step, a new $\widetilde{S}_t$ is provided by the algorithm, which results in a different feature vector consisting of features $\widetilde{F}_{\widetilde{S}}$, and a different energy cost. The amount of energy consumed can then be compared with the amount consumed for the reference case when $\widetilde{S} = S$, that is, when all sensors remain on, for comparison.

As with energy consumption, the simulation environment also records classification results, both for the novel algorithm and for the reference case. For each time step, the algorithm classifies $\widetilde{F}_{\widetilde{S}}$ and the result of the classification is recorded, along with the energy consumption. At the same time, the complete feature vector F_S, consisting of the entire feature set F, is also classified, and the result is stored for comparison with the reference system. In total, the simulator records the energy consumption and classification results for both the novel prediction-based activity recognition algorithm and the reference case when all sensors remain on for both data sets.

4.2 Artificial data set generation

In order to evaluate the behavior of the system for different degrees of predictability (κ), artificial data sets are generated using the original data set and a generative probabilistic model shown in Fig. 4. The goal is to generate a data set which is predictable to a specified degree by the predicting algorithm, meaning that it results in a certain prediction accuracy. A Markov chain assumes that the process being modeled holds with the Markov property. It follows that by changing how pronounced the Markov property is in the data, the accuracy of the predictor can be set. The predictability is defined as $\kappa \in [\frac{1}{|C|}, 1]$ where a value of 1 indicates that

$$\forall i \exists j | P(c_{t+1} = c_j | c_t = c_i) = 1$$

and a value of $\frac{1}{|C|}$ indicates that

$$\forall i, j | P(c_{t+1} = c_j | c_t = c_i) = \frac{1}{|C|}$$

or that all transitions are equally likely. Setting $\kappa = \frac{1}{|C|}$ is the lower bound for predictability, as there are $|C|$ transitions leaving each state, and the probabilities of all exit

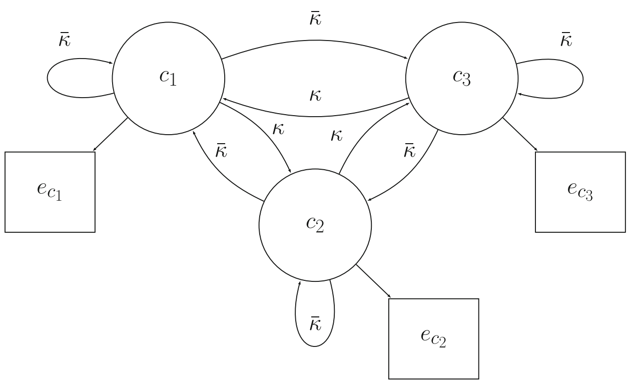

Fig. 4 Generative model for constructing an artificial data set with 3 classes (C), emissions (E), and predictability $\boldsymbol{\kappa}$

transitions must sum to one. Assigning κ a lower value than this means at least one exit transition must have a probability higher than $\frac{1}{|C|}$, increasing predictability.

Using κ, we can generate a HMM (not to be confused with the HMM used for recognition) by ordering states such that each state has 1 and only 1 transition to a different state with probability κ, and only 1 transition from a different state to itself with probability κ. All other transitions have probability $\bar{\kappa} = \frac{1-\kappa}{|C|-1}$. Simply put, as κ approaches 1, the state following the current state becomes more and more certain, and therefore easier to predict. As κ approaches the lower bound of $\kappa = \frac{1}{|C|}$, the next state becomes more random and harder to predict.

Once this model is created, traversing it generates emissions which are sample windows from the original data set for the given activity. This is demonstrated in Fig. 4 for an example 3-class dataset. Although this artificial data set does not represent a realistic pattern of the human activities in the data set, it does create a data set which is predictable to a specified degree. As will be shown later, the results are only dependent on the prediction accuracy, meaning that for realistic scenarios with identical prediction rates, the results should theoretically still hold.

4.3 Experimental process

The algorithm presented here is not application specific. It is meant to reduce the cost of embedded activity and context recognition in scenarios with repetitive temporal patterns. Each application is different in terms of the optimal trade-off between energy consumption and accuracy [28]. The following evaluation is conducted without a specific cost model, but allows the reader to evaluate the effectiveness for their application scenario at hand.

The classifiers used are the Hidden Markov Model [22] (HMM) and the k-Nearest Neighbors (kNN) [7] algorithms, as they are both easily adapted to a variable feature vector length. The algorithm requires two separate sets of training data, one to train the classifier and predictor, and a separate one to populate the Q matrix using the trained classifier. A third data set is required for evaluation.

4.3.1 Training phase

Each artificial data set is partitioned into 3 sections. The data used to train the classifier and predictor $\widetilde{D}_{\text{Train}}$ make up 60 % of the original data set D. Another 20 % $\widetilde{D}_Q$ is used to calculate the Q matrix, as using $\widetilde{D}_{\text{Train}}$ for this purpose results in overfitting, and therefore distorted loss values in Q. Finally, the last 20 % $\widetilde{D}_{\text{Eval}}$ is used to evaluate the performance of the whole system and in this experiment contains 3,595 sample windows in total.

In the first step, $\widetilde{D}_{\text{Train}}$ is used to train the classifier, either HMM or kNN, as well as the Markov chain used for prediction. In this phase, sensor selection is not conducted, and both of the classifier instances and the predictor are trained on all features $f \in F$. In the second training step, $\widetilde{D}_Q$ is used to populate the Q matrix by evaluating the recognition rate of every class c_i with every permutation of $\widetilde{S}$. Therefore, every combination in $C \times \wp(S)\backslash\{\}$ is evaluated in a separate classification phase, using all vectors for activities of the current subset.

4.3.2 Testing phase

In the testing or evaluation phase, the classifier algorithms are run on $\widetilde{D}_{\text{Eval}}$ in parallel. At each classification time step, the $\widetilde{S}_t$ resulting from the previous time step is used to generate a new feature vector $\widetilde{F}_t$. This vector is then classified, either by the HMM or kNN classification algorithm. Based on this classification, the algorithm predicts ρ probable classes $\widetilde{C}_{t+1}$ for the next time step. Next, the sensor subset $\widetilde{S}_{t+1}$ is selected such that it fulfills Eq. (1). At the end of each step, the simulation environment records the classification result using F, $\widetilde{F}_t$, the ground truth for that sample window, the sensor subset $\widetilde{S}$, the energy consumed $\text{En}(\widetilde{S}_t)$ and the predictions $\widetilde{C}_{t+1}$ for the next time step.

5 Evaluation

The evaluation presented here covers several months of simulation time. For each different degree of scenario *predictability* κ (MVS: from 0.125 to 0.875 step 0.125, OPP: from 0.25 to 0.875 step 0.125), a different artificial data set was generated. The *number of predicted states* ρ (MVS: from 1 to 8 step 1, OPP: from 1 to 4 step 1), the *acceptable loss parameter* λ (from 0 to 1 step 0.1), and the classifier (MVS: HMM and kNN, OPP: kNN) were permuted to evaluate the output parameters over each data set. The OPP data set was only evaluated using the one kNN classifier to maintain brevity, and as it suffices to validate the conclusions drawn from the MVS results. These results are multi-dimensional, consisting of dimensions ρ, λ, and κ, the classifier, data set, recognition rates, and energy consumption. It is impossible to impart this information in its entirety here; therefore, we will detail and demonstrate major insights with graphical excerpts.

5.1 Results of the MVS data set

5.1.1 *MVS: recognition loss*

We define recognition loss as the difference in percentage points (pp) between the reference recognition rate (in percent) with all sensors on, and the recognition rate for the novel algorithm for a given set of parameters. For a given loss parameter λ, loss of recognition decreases monotonically (meaning recognition increases) for an increasing ρ (number of states predicted). For the MVS data set, this is demonstrated by Fig. 5 for a κ of 0.125, and again in Fig. 6 for a κ of 0.875 for both the HMM and kNN classifiers. This is again evident in Fig. 8, where for a given λ, increasing ρ either reduces or leaves recognition loss unchanged.

In other words, for a specific number of classes predicted at each step (ρ), if the parameter λ which identifies how much loss is acceptable for a specific application is increased, the loss in recognition does indeed increase.

The same also applies to the acceptable loss λ, where for a given classifier and ρ, loss in recognition and energy savings increase monotonically with λ. The implication is that the acceptable loss parameter λ does indeed function as an indicator for how much recognition can be sacrificed as proposed. The monotonic behavior of recognition loss implies that for a given predictability κ, the lowest recognition loss (best recognition) is obtained by $\rho = |C|$ and $\lambda = 0$, and the highest loss (worst recognition) when $\rho = 1$ and $\lambda = 1$.

Observing accuracy loss over κ for fixed values of λ and ρ is not as clear cut. In Fig. 7, varying κ affects recognition for $\rho = 4$ using the HMM, where the trend in recognition loss is decreasing as κ increases, although not monotonically (compare $\kappa = 0.125$ with $\kappa = 0.875$ for $\lambda = 0.8$).

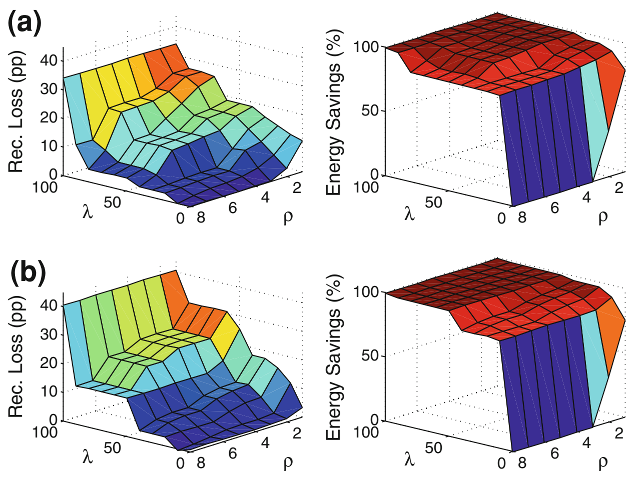

Fig. 5 Recognition loss and energy savings for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\kappa} = 0.125$

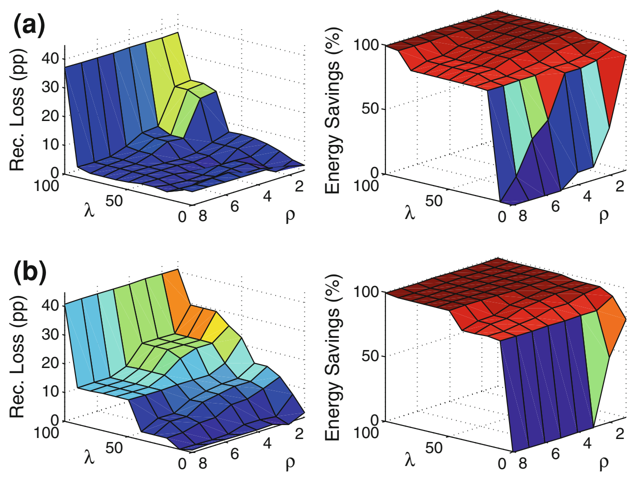

Fig. 6 Recognition loss and energy savings for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\kappa} = 0.875$

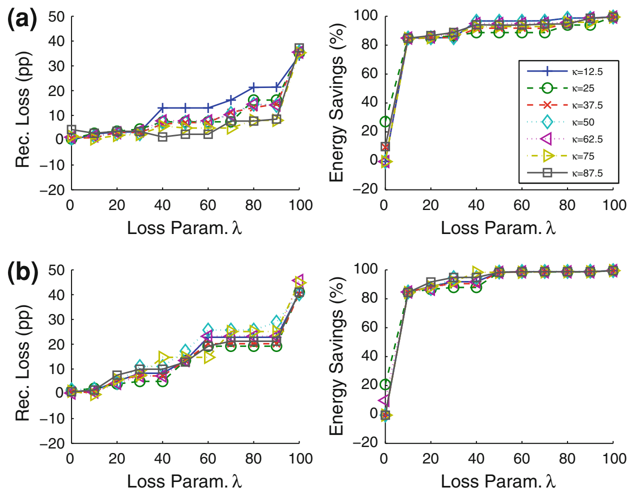

Fig. 7 Recognition loss and energy savings for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\rho} = 4$

For the kNN classifier, the effects of κ are minimal when compared to the HMM as seen in Figs. 5 and 6.

5.1.2 *MVS: energy consumption rates*

The energy savings are defined as the relative decrease in energy consumed over the evaluation of $\widetilde{D}_{\mathrm{Eval}}$ between the reference classifier with all sensors on and the novel algorithm. When observing Figs. 5 and 6, the acceptable loss parameter λ has a far greater influence on energy savings than either ρ or κ. Figures 7 and 8 demonstrate this by showing very little differentiation in energy savings for either κ or ρ, respectively. In all cases, a relatively small values of λ ($\approx$0.1) suffice for large energy savings (>80 %).

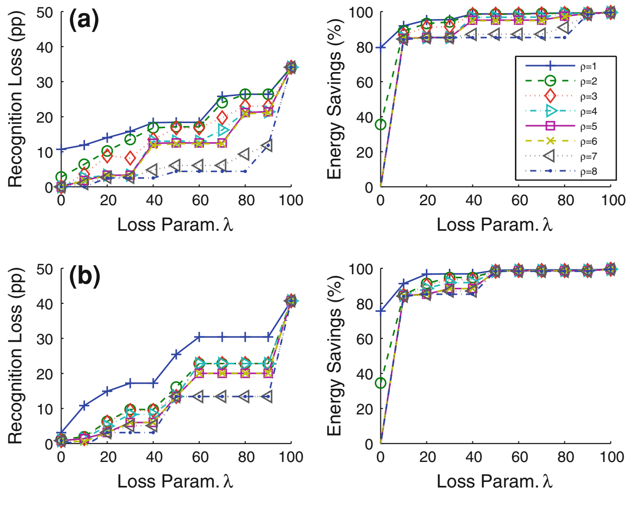

Fig. 8 Recognition loss and energy savings for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\kappa} = 0.125$

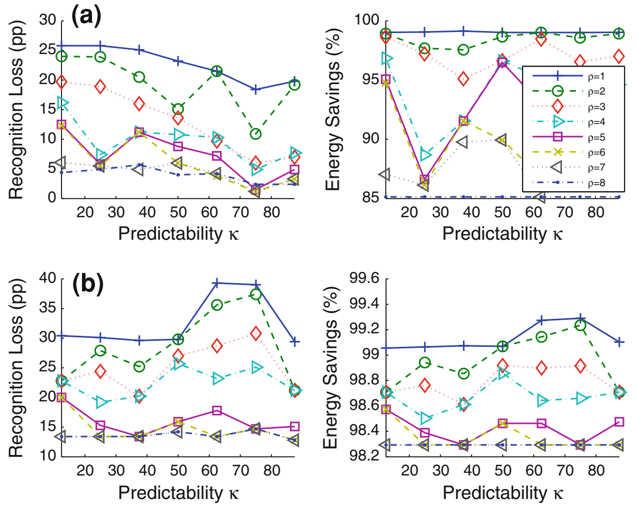

Fig. 9 Recognition loss and energy savings for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\lambda} = 0.8$

All of the images displaying the results show a rapid increase in energy savings for even small acceptable loss values. This increase is caused by the light sensor, which consumes an order of magnitude more energy than the vibration sensor, for example, see Table 3. The upper bound for energy savings, as well as for recognition loss, is given by using the cheapest sensor only, namely the MVS vibration sensor [14]. The light sensor is the first to be shut off, creating the steep climb over low values of λ seen clearly in Figs. 5 and 6.

Another slight increase can be seen around $\lambda = 0.5$ corresponding to the acceleration sensor. Shutting off this sensor, however, causes large increases in recognition loss. In other words, the algorithm filters out those sensors first which contribute little, but cost a lot.

5.1.3 MVS: classifier comparison

The kNN classifier performance for the reference case (all sensors on) remained stable across κ with recognition rates between 79.6 and 80.1 %. On the other hand, reference recognition rates for the HMM varied in performance from 70.5 % for $\kappa = 0.125$ to 81.6 % for $\kappa = 0.875$, indicating that the recognition rates of the HMM are quite dependent on the predictability of the scenario. This can be seen again in Fig. 7, where recognition losses vary little for all values of κ for the kNN classifier, but are further spread out for the HMM classifier. However, the recognition rate for the HMM is consistently higher than for the kNN classifier for the same parameters. This can be seen when comparing the top left and bottom left images in Figs. 5 and 6.

On the other hand, the kNN classifier appears to be consistently better at conserving energy than the HMM classifier, as seen in Fig. 8 when comparing energy savings of the HMM and kNN classifiers for $\lambda = 0.1$ or $\lambda = 0.6$, for example. Figure 6 demonstrates that this is also evident for other values of κ. Both Fig. 5 and Fig. 6 indicate that the energy consumption of the kNN classifier is also less for higher values of ρ, staying constant where the HMM energy savings fall off. Figure 9 confirms this (noisily) by indicating higher savings for the kNN classifier compared to the HMM, and less variance over κ for higher values of ρ.

5.1.4 MVS: prediction rates

One potential issue which was mentioned earlier is that incorrect predictions can lead to incorrect sensor configurations and incorrect classifications. The apparent problem

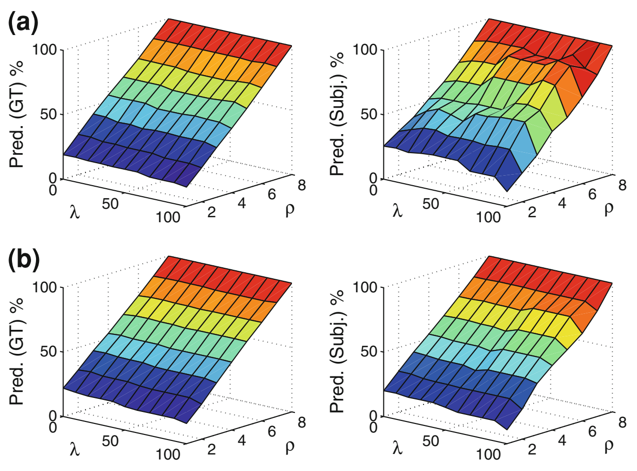

Fig. 10 Prediction against ground truth and classification for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\kappa} = 0.125$

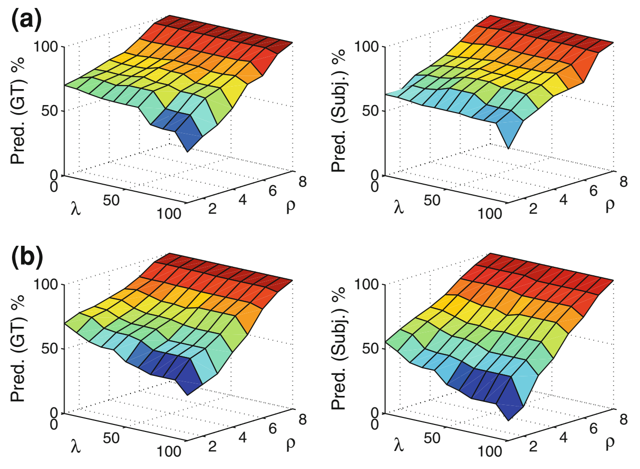

Fig. 11 Prediction against ground truth and classification for the MVS Data Set and the HMM (**a**) and kNN (**b**), $\boldsymbol{\kappa} = 0.875$

is that this can then again lead to another incorrect prediction, fueling the cycle. To evaluate the effects of these phenomena, the prediction rates of the system were also evaluated with respect to κ, ρ, and λ.

Figure 10 shows the prediction accuracy with respect to ρ and λ for the (a) HMM and (b) kNN classifiers with a predictability of $\kappa = 0.125$. On the left-hand side, the prediction accuracy with respect to the ground truth is shown, while on the right the accuracy with respect to system classifications is displayed. The latter indicates the correctness of prediction as seen from the subjective point of view of the algorithm's own classifications. When observing these graphs, the first thing which is clear is that the prediction accuracy is heavily dependent on ρ, or the number of states predicted. The linear relation between ρ and prediction accuracy is to be expected. For $\lambda = 0$, the ratio of correct to incorrect predictions should range from κ for $\rho = 1$, to 1 for $\rho = |C|$, where since all classes are predicted, the prediction is always correct. This is evident in Fig. 10, where the deviance in prediction accuracy with respect to the expected value of κ is due to misclassification. For comparison, Fig. 10 displays the same information for $\kappa = 0.875$, where the linear behavior is still evident but with an increased offset for $\lambda = 0$. Theoretically, this offset should be proportional to κ, or 87.5, where the difference is due to recognition errors.

For lower values of κ, the acceptable loss parameter λ has little effect on the accuracy of the prediction algorithm. This is due to the fact that as κ approaches $\frac{1}{|C|}$, predictions approach random; therefore, errors caused by increasing loss in recognition do not affect the randomness of the prediction. As κ increases, the effects of λ also increase, as can be seen when comparing the left column of Fig. 10 with the left column of Fig. 11. Furthermore, these effects are stronger for the kNN classifier as opposed to the HMM classifier, as the later has an internal Markov chain which stabilizes the prediction.

When comparing the left columns of Figs. 10 and 11 with their respective right columns, it is evident that the system's subjective evaluation in terms of its own prediction performance is fairly accurate with respect to its actual performance evaluated against ground truth. Only for high values of λ is there a noticeable discrepancy, where the discrepancy increases as the predictability κ increases.

5.2 Results of the OPP data set

The activity recognition data set from the OPPORTUNITY project [24] was also evaluated using the prediction-based method to confirm initial results from the MVS data set [11]. For this purpose, the results of the kNN classifier alone are sufficient, and therefore, the HMM results have been omitted for brevity.

5.2.1 OPP: recognition loss

The behavior of the recognition loss for the OPP data set with a predictability of $\kappa = 0.25$ is displayed in Fig. 12. Remember, for this data set, this value represents random ordering as there are only 4 activity classes, as opposed to 8 classes in the MVS data set. Here, a plateau in recognition loss can be clearly seen at 12.2 pp for all values of the loss parameter $\lambda \geq 0.2$. This same plateau can be seen for $\kappa = 0.875$ in Fig. 13 as well.

This plateau behavior is generated by the cheapest sensor in terms of energy cost, in this case the magnetic field sensor (see Table 3). As opposed to the MVS data set, this sensor alone provides relatively high accuracy, indicating that increasing λ quickly leads the system to select that sensor alone as the lowest energy sensor configuration for achieving the required recognition.

5.2.2 OPP: energy savings

Similar to recognition loss, energy savings also plateau at 94.8 % for values of $\lambda \geq 0.2$, which is also the optimum

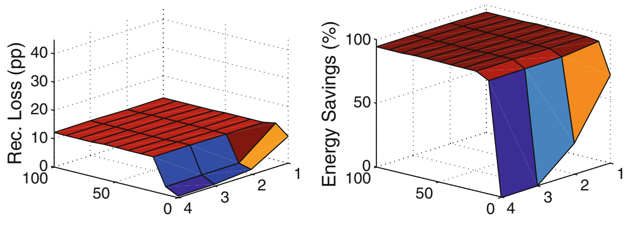

Fig. 12 Recognition loss and energy savings for the OPP Data Set and kNN with $\boldsymbol{\kappa} = 0.25$

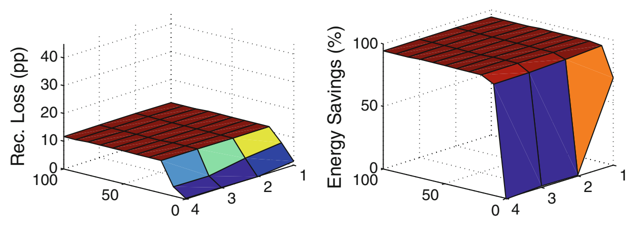

Fig. 13 Recognition loss and energy savings for the OPP Data Set and kNN with $\kappa = 0.875$

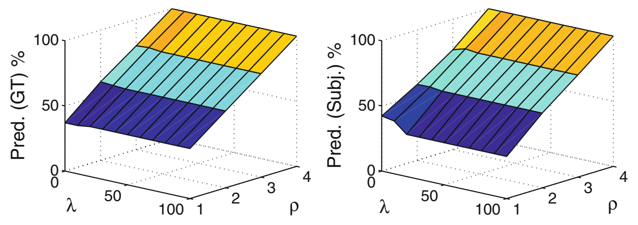

Fig. 14 Prediction against ground truth and classification for the OPP Data Set and kNN classifier with $\kappa = 0.25$

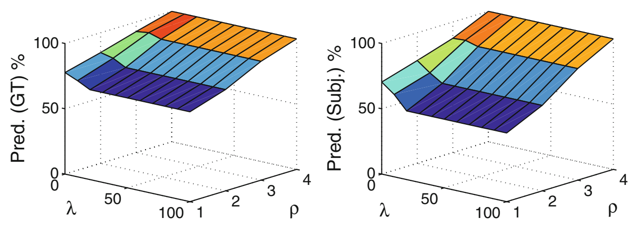

Fig. 15 Prediction against ground truth and classification for the OPP Data Set and kNN classifier with $\kappa = 0.875$

for energy savings. This can be seen for both $\kappa = 0.25$ in Fig. 12 and for $\kappa = 0.875$ in Fig. 13. Again, this is caused by the loss parameter quickly dropping below the rates achievable using the single cheapest sensor, yielding an optimal configuration of only that sensor.

5.2.3 OPP: prediction rates

Figure 14 displays the prediction rates for the OPP data set with a kNN classifier against ground truth (left) and system classifications (right) for $\kappa = 0.25$. This is the case where the distribution is random for future activities, where all 4 activities are equally likely. For $\rho = 1$, prediction is steady at approximately 38 %, above the expected 25 % given by the generated data set. This is caused by recognition error which may have a different predictability inside of a single class due to the generation process.

The same information is presented in Fig. 15 for $\kappa = 0.875$. For $\rho = 1$, prediction values begin at around 78 % for $\lambda = 0$, but drop off to around 68 % once the sensor set is reduced to the compass alone for values of $\lambda \geq 0.2$. Again, the values increase linearly from that point to 100 % as ρ increases toward the maximum value of 4.

In both Figs. 14 and 15, it can be seen that the subjective evaluation of the prediction value for the system is fairly accurate. Here, an increase for lower values of λ can be seen, as well as a reduction in prediction values. This reduction is around 10 to 15 pp for $\rho = 1$ and $\kappa = 0.875$, which falls off to 0 for $\rho = |C|$. Intuitively, this can be interpreted as the system correctly predicting the next activity, but judging this as a false positive due to misclassification.

6 Discussion and insight

In Figs. 5 and 6, non-zero energy savings are present, even when $\lambda = 0$. Intuitively, setting $\lambda = 0$ means that any loss in recognition is unacceptable. For certain classes, some sensors are so insignificant that shutting them off results in an error increase so small that it is approximated to 0 by floating point arithmetic operations. Here, the expensive light sensor is useless for most classes and can be shut off with no loss as long as none of the classes requiring it is predicted. As more states are predicted, however, classes requiring that sensor are more frequently predicted, regardless of their occurrence rate, increasing consumption with no effect on recognition.

The situation when $\rho = 1$ is extremely volatile, since only the single most likely future class is predicted. Figures 5 and 6 show that $\rho = 1$ has significant negative effects for all non-zero values of λ. False classifications result in false predictions, resulting in false classification again. Introducing a confidence value at this point may allow the system to recognize error occurrence and correct by switching sensors back on when confidence or probabilities for all classes are low. The low recognition rates for $\rho = 1$ indicate that for all real scenarios, $\rho = 1$ should probably not be considered. For higher values of κ such as 0.875 in Fig. 6, predicting as few as two states at each step can be sufficient.

The kNN classifier was more resistant to noise with respect to predictability within the data set. As κ decreases, Fig. 7 indicates that for a fixed loss parameter λ, the recognition loss expands faster for the HMM than for the kNN. In Fig. 9, the energy consumption for a fixed ρ grows faster and becomes more erratic for the HMM when compared to the kNN. The HMM algorithm models the activities as a Markov process [22], meaning that unpredictable feature vectors not only affect prediction, but classification as well. The kNN algorithm is only affected by κ through incorrect sensor activations which reduce recognition.

Both classification algorithms are influenced by lower values of κ due to sub-optimal sensor activations. The effect can be counteracted by increasing the number of classes predicted ρ, improving recognition accuracy but reducing the gain in energy. The parameter ρ controls the balance between risk and reward. High values of ρ mean less risk but a smaller payoff, and lower values increase the win in energy at the cost of recognition. The predictability of a scenario can be easily obtained for real scenarios by taking the accuracy of the prediction algorithm over the training data. Once κ is known, ρ can be configured to counteract it and select an appropriate risk level using Fig. 9 as a heuristic.

Once the risk and reward trade-off between ρ and κ has been found, the loss parameter λ can be assigned to optimize the amount by which recognition may be reduced, and thereby the amount of energy which is conserved. For example, assuming a κ value of 0.5, $\rho = 3$ to counteract and a loss parameter of $\lambda = 0.2$, a kNN classifier incurs a loss of less than 1.5 pp but save 84.8 % of energy consumed for the MVS data set and 2.8 pp and 89.9 % for the OPP data set.

One caveat is that due to the nature of prediction-based optimization, the system may perform badly for recognition of important but rare and unpredictable events or activities. This will arise if those activities require special sensors to distinguish them from other activities. Since the events cannot be easily predicted, the important sensors will not be correctly activated. A possible solution to this could be to exclude the required sensors from the set of sensors which can be deactivated, leaving them on at all times. Also, one could use expert knowledge to hard-code the circumstances under which the events occur into the system if possible. If some activities are more important than others in general, investigating the integration of activity importance weights into the activity-sensor weights in the Q matrix could provide an interesting avenue of research.

Another interesting aspect which has not been addressed here is that often times the sensors of the mobile device are also used for purposes other than activity recognition alone, such as is the case with mobile phones. Under these circumstances, it is not advisable to switch these sensors off using an algorithm which does not take user preferences into account. In theory, the algorithm could be easily adapted to account for sensors which are in use by a user application $\widetilde{S}_{\text{app}}$. At each prediction step, Eq. (1) can be adapted to only search the Q matrix for sensor configurations which are a superset of $\widetilde{S}_{\text{app}}$. In this way, the optimal sensor subset can be selected given that the subset $\widetilde{S}_{\text{app}}$ is activated.

Here, we have evaluated the performance of the algorithm for a specified and fixed λ during runtime. Practically speaking, there is no reason why the acceptable loss cannot be changed dynamically during operation. This could have advantages for applications with mobile phones, where requirements on the energy source are also dynamic in nature. For example, when the phone is connected to a power supply, λ can be set to 0 as the power source is effectively unlimited. However, as the battery level approaches a critical level, λ can be increased to extend the battery life as long as possible. Alternatively, devices could also try and recognize patterns in the daily lives of users [31], and set λ to appropriately account for the time until the device will probably be recharged.

One issue which has not been addressed is that although λ is proportional to how much recognition will be sacrificed, it does not provide an exact amount. The actual loss is a function of λ, the predictability κ, the number of predicted class ρ, and the number of total classes $|C|$, as well as the reference recognition rate when all sensors are on. The implication is that at training time, actual losses in recognition are unknown, as these are not only dependent on system parameters but also on the reference recognition rate for the given activities. One solution is that when gathering training data, a small amount can be set aside for parameter tuning before the system is put online.

7 Conclusion

We proposed a novel method for saving energy while recognizing human activities using embedded and wearable sensing systems. We conducted a survey of existing techniques which revealed a taxonomy of approaches to this problem. Based on that taxonomy, we introduced the novel method for sensor system control which uses prediction to further conserve energy. Human beings are repetitive and periodic creatures; therefore, what we do can be predicted to a certain extent. Sensors which are not needed to decipher probable activities from each other can be turned off, conserving energy without greatly impacting recognition rates. The algorithms are simulated using preexisting data sets [14, 24], which are used to generate artificial scenarios with specific degrees of predictability. The standard classification (Hidden Markov Model and k-Nearest Neighbors) and prediction (Markov Chain) algorithms were used for the evaluation so as to make the effects of the novel methods more pronounced.

The results indicate that for highly predictable scenarios, significant savings are possible with little loss in recognition. For less predictable scenarios, losses are higher, but the predictability can be artificially increased by predicting more than one state per iteration. This, however, reduces the savings in energy achievable, but limits the loss due to missed predictions. The Hidden Markov Model which takes state transitions into account performed better the kNN which does not, although low predictability has a greater effect on recognition. Finally, energy savings and recognition loss are greatly dependent on the activities being recognized and the sensors being used. The limit is given by the recognition rate using only the cheapest sensor in terms of energy consumption. The recognition and energy consumption converge to the performance given by using only that sensor for recognition as the system is granted increasing freedom to optimize at the cost of recognition.

Although this evaluation yielded positive results, there are still some drawbacks to this approach. Applications which attempt to recognize important but unusual events could suffer greatly. This could come about if the important events require special sensors but their occurrence cannot be predicted, where the prediction error would lead to those sensors not being activated in the time of need. Counteracting this effect requires further research where several approaches have been proposed.

The results are also independent of the specific recognition and prediction algorithms used, depending only on the prediction and recognition rates achieved. The simulation results can therefore be applied to a new scenario once the predictability (prediction accuracy) has been found. Although the acceptable loss parameter does indeed control how much loss is incurred, the function for this dependence is not straightforward and some training data should be used for parameter fitting. In this way, the system designer can ensure that the actual loss in recognition incurred meets application requirements. For a scenario with predictability of 0.5 and 3 classes predicted at each time step, a loss parameter of 0.2 with the kNN classifier would incur a recognition loss of less than 1.5 pp but save 84.8 % of energy consumed for the MVS data set and 2.8 pp and 89.9 % for the OPP data set.

An avenue of further research would be to integrate a reliability measure into the classification and prediction processes. Using such a method could allow the system to identify system configurations which do not allow reliable classification of activities, that is, when necessary sensors are not active due to incorrect prediction, and take necessary measures. Another open research question is how the system would perform using low-level prediction based on the sensor data streams as opposed to an activity history. This would further decouple the performance of the recognition and prediction processes and could prevent the negative feedback loop between incorrect classifications and predictions which occurred in some extreme situations. Finally, although the experiment was carefully designed to simulate real conditions, the work presented here would benefit from online experiments, for example, with a mobile phone, to verify these results in live scenarios. Currently, we are planning a semi-supervised approach to evaluating the algorithm under real conditions, as this has proven effective with other approaches to conserving energy for embedded recognition [34].

References

1. Au L, Bui A, Batalin M, Kaiser W (2012) Energy-efficient context classification with dynamic sensor control. Biomed Circ Syst IEEE Trans 6(2):167–178
2. Benbasat AY, Paradiso JA (2007) A framework for the automated generation of power-efficient classifiers for embedded sensor nodes. In: SenSys '07: proceedings of the 5th international conference on embedded networked sensor systems. ACM, New York, pp 219–232
3. Berchtold M, Budde M, Gordon D, Schmidtke H, Beigl M (2010) Actiserv: activity recognition service for mobile phones. In: Wearable Computers (ISWC), 2010 international symposium, pp 1–8
4. Bharatula N, Stager M, Lukowicz P, Tröster G (2005) Power and size optimized multi-sensor context recognition platform. In: Wearable Computers, 2005. Proceedings. Ninth IEEE international symposium, pp 194–195
5. Bharatula NB, Ossevoort S, Stäger M, Tröster G (2004) Towards wearable autonomous microsystems. In: Pervasive, pp 225–237
6. Bharatula NB, Stäger M, Lukowicz P, Tröster G (2005) Empirical study of design choices in multi-sensor context recognition systems. In: IFAWC: 2nd international forum on applied wearable computing, pp 79–93
7. Bishop CM (2006) Pattern recognition and machine learning, 1st ed. 2006. corr. 2nd printing edn. Springer, Berlin
8. Bouten C, Koekkoek K, Verduin M, Kodde R, Janssen J (1997) A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. Biomed Eng IEEE Trans 44(3):136–147
9. Cakmakci O, Coutaz J, Laerhoven KV, Werner Gellersen H (2002) Context awareness in systems with limited resources. In: In Proceeding of the third workshop on artificial intelligence in mobile systems (AIMS), ECAI 2002, pp 21–29
10. Gao L, Bourke A, Nelson J (2012) Activity recognition using dynamic multiple sensor fusion in body sensor networks. In: Engineering in Medicine and Biology Society (EMBC), 2012 annual international conference of the IEEE, pp 1077–1080
11. Gordon D, Czerny J, Miyaki T, Beigl M (2012) Energy-efficient activity recognition using prediction. In: Wearable Computers (ISWC), 2012 16th international symposium, pp 29–36
12. Gordon D, Hanne JH, Berchtold M, Miyaki T, Beigl M (2012) Recognizing group activities using wearable sensors. In: Puiatti A, Gu T (eds) Mobile and ubiquitous systems: computing, networking, and services, lecture notes of the institute for computer sciences, social informatics and telecommunications engineering. Springer, Berlin, vol 104, pp 350–361
13. Gordon D, Hanne JH, Berchtold M, Shirehjini A, Beigl M (2012) Towards collaborative group activity recognition using mobile devices. Mobile Networks and Applications pp 1–15
14. Gordon D, Schmidtke H, Beigl M, von Zengen G (2010) A novel micro-vibration sensor for activity recognition: potential and limitations. In: Wearable Computers (ISWC), 2010 international symposium, pp 1–8
15. Gordon D, Sigg S, Ding Y, Beigl M (2011) Using prediction to conserve energy in recognition on mobile devices. In: Pervasive computing and communications workshops (PERCOM Workshops), 2011 IEEE international conference, pp 364–367
16. Könönen V, Mäntyjärvi J, Similä H, Pärkkä J, Ermes M (2010) Automatic feature selection for context recognition in mobile devices. Pervasive Mob Comput 6(2):181–197

17. Krause A, Ihmig M, Rankin E, Leong D, Gupta S, Siewiorek D, Smailagic A, Deisher M, Sengupta U (2005) Trading off prediction accuracy and power consumption for context-aware wearable computing. In: Wearable Computers, 2005. Proceedings. Ninth IEEE international symposium, pp 20–26
18. Lin K (2010) Energy-accuracy aware localization for mobile devices. In: Proceedings of the 8th international conference on mobile systems applications and services MobiSys 10. In Mobisys, ACM Press, p 285
19. Lu H, Brush AJB, Priyantha B, Karlson AK, Liu J (2011) SpeakerSense: energy efficient unobtrusive speaker identification on mobile phones. Pervasive Comput 6696:188–205
20. Mayrhofer R, Radi H, Ferscha A (2003) Recognizing and predicting context by learning from user behavior. In: Kotsis WSG, Ferscha A, Ibrahim K (eds) Proceedings MoMM 2003: 1st international conference on advances in mobile multimedia. Austrian Computer Society (OCG), vol 171, pp 25–35
21. Paek J, Kim J, Govindan R (2010) Energy-efficient rate-adaptive GPS-based positioning for smartphones. In: Proceedings of the 8th international conference on Mobile systems applications and services MobiSys 10, MobiSys 10. ACM, ACM Press, p 299
22. Rabiner L (1989) A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2):257–286
23. Raffa G, Lee J, Nachman L, Song J (2010) Don't slow me down: bringing energy efficiency to continuous gesture recognition. In: Wearable Computers ISWC 2010 international symposium. IEEE, pp 1–8
24. Roggen D, Calatroni A, Rossi M, Holleczek T, Forster K, Tröster G, Lukowicz P, Bannach D, Pirkl G, Ferscha A, Doppler J, Holzmann C, Kurz M, Holl G, Chavarriaga R, Sagha H, Bayati H, Creatura M, del R Millan J (2010) Collecting complex activity datasets in highly rich networked sensor environments. In: Networked sensing systems (INSS), 2010 Seventh international conference, pp 233–240
25. Roy N, Misra A, Julien C, Das SK, Biswas J (2011) An energy-efficient quality adaptive framework for multi-modal sensor context recognition. In: IEEE international conference on pervasive computing and communications PerCom. Institute for Infocomm Research, Singapore, IEEE, pp 63–73
26. Sigg S, Gordon D, Zengen G, Beigl M, Haseloff S, David K (2011) Investigation of context prediction accuracy for different context abstraction levels. Mobile Computing, IEEE Trans (99):1
27. Stäger M, Lukowicz P, Tröster G (2004) Implementation and evaluation of a low-power sound-based user activity recognition system. In: ISWC '04: Proceedings of the eighth international symposium on wearable computers. IEEE Computer Society, Washington, pp 138–141
28. Stäger M, Lukowicz P, Tröster G (2007) Power and accuracy trade-offs in sound-based context recognition systems. Pervasive Mob Comput 3:300–327
29. Sun FT, Kuo C, Griss M (2011) Pear: Power efficiency through activity recognition (for ecg-based sensing). In: Pervasive computing technologies for healthcare (PervasiveHealth), 2011 5th International conference, pp 115–122
30. Thatte G, Li M, Lee S, Emken A, Narayanan S, Mitra U, Spruijt-Metz D, Annavaram M (2012) Knowme: an energy-efficient multimodal body area network for physical activity monitoring. ACM Trans Embed Comput Syst 11(S2):48:1–48:24
31. Van Laerhoven K, Kilian D, Schiele B (2008) Using rhythm awareness in long-term activity recognition. In: Proceedings of the 12th international symposium on wearable computers (ISWC 2008). IEEE Press, Pittsburgh, pp 63–68
32. Wang Y, Krishnamachari B, Annavaram M (2012) Semi-Markov state estimation and policy optimization for energy efficient mobile sensing. In: The 9th annual IEEE communications society conference on sensor, mesh and Ad Hoc communications and networks (SECON'12)
33. Wood A, Merrett G, Gunn S (2012) Adaptive sampling in context-aware systems: a machine learning approach. IET wireless sensor systems 2012
34. Yan Z, Subbaraju V, Chakraborty D, Misra A, Aberer K (2012) Energy-efficient continuous activity recognition on mobile phones: an activity-adaptive approach. In: 16th international symposium on wearable computers, pp 17–24