# Dynamic Quantification of Activity Recognition Capabilities in Opportunistic Systems

Marc Kurz, Gerold Hölzl, Alois Ferscha
Institute for Pervasive Computing
Johannes Kepler University of Linz, Austria
{kurz, hoelzl, ferscha}@pervasive.jku.at

Hesam Sagha, José del R. Millán, Ricardo Chavarriaga
STI-CPN-CNBI
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{hesam.sagha, jose.millan, ricardo.chavarriaga}@epfl.ch

***Abstract*—Opportunistic activity and context recognition systems draw from the characteristic to use sensing devices that just happen to be available instead of pre-defining them at the design time of the system in order to achieve a recognition goal at runtime. Whenever a user and/or application states a recognition goal at runtime to the system, the available sensing devices configure an ensemble of the best available set of sensors for the specified recognition goal. This paper presents an approach to show how machine learning technologies (classification, fusion and anomaly detection) are integrated in a prototypical opportunistic activity and context recognition system (referred to as the *OPPORTUNITY Framework*). We define a metric that quantifies the ensemble's capabilities according to a recognition goal and evaluate the approach with respect to the requirements of an opportunistic system (e.g. to compute an ensemble's configuration and reconfiguration at runtime).**

***Index Terms*—Activity and Context Recognition, Opportunistic Sensing, Sensor Networks.**

## I. Introduction

Activity and context recognition in pervasive computing applications interpret environmental data, acquired by (wireless) sensing devices, in terms of inferring activities and more generally the context ([1], [2]) of persons and subjects in real world environments. All activity and context recognition systems have shared one major problem so far: the sensor deployment is application-specific and thus the mapping from sensor signals to context and activities has to be known at *design time*. Therefore, the concept of opportunistic activity and context recognition systems draws from the characteristic to use sensing devices that just happen to be available rather than pre-defining the sensing infrastructure at design time of the system [3], [4]. A user and/or application states a recognition goal to an opportunistic system at *runtime* and the currently available sensing devices in the environment configure themselves to a sensing *ensemble* which is the set of the most appropriate available sensors. An essential requirement for this configuration is therefore quantification of the ensemble's capabilities according to a recognition goal. This metric indicates the degree to which a sensing ensemble or a single sensor is suited to execute a goal (i.e. the *degree of fulfillment - DoF*). Furthermore, a second metric, the *Trust Indicator* (*TI*), can influence the ensemble configuration process at runtime. This metric indicates the plausibility of the currently delivered sensor data. If for example the sensor is rotated, moved, or faulty, the ensemble is reconfigured if necessary. As sensor-based activity and context recognition systems work with different machine learning (ML) technologies [5] (e.g. feature extraction, classification, fusion, ...), we have implemented a prototypical framework (referred to as *OPPORTUNITY Framework*) (see [6], [7] and [8]) which uses a rich dataset recorded in a kitchen scenario ([9] and [10]) for simulating environmental measurements in a breakfast scenario (the *OPPORTUNITY Dataset*). The research challenge of this paper is to investigate the question how and to what extent ML technologies can be applied to compute metric values that quantify the capabilities of a sensing ensemble according to a recognition goal. This is done by using the OPPORTUNITY Framework and its capabilities of applying sensor abstractions in the form of playback sensing devices [8] as a prototypical implementation and development base for evaluating and testing approaches of an opportunistic system which simulates a kitchen that is equipped with more than 70 sensing devices of 10 modalities (the details of the kitchen dataset can be found in [9] and [10]).

The rest of the paper is structured as follows. Section II describes the OPPORTUNITY Framework and the algorithmic approach showing how different ML technologies can be utilized to calculate the degree of fulfillment and the trust indicator for a sensing ensemble according to a recognition goal. Section III describes a test scenario, utilizing the kitchen/breakfast dataset to evaluate the opportunistic goal recognition and quantification mechanism. The paper closes with a conclusion in Section IV.

## II. Application of ML-Technologies for Quantification

### A. Introduction of the OPPORTUNITY Framework

The OPPORTUNITY Framework - implemented with Java and the OSGi module system - is meant to be (i) a prototypical implementation of a mobile opportunistic activity and context recognition system, and (ii) a first step towards a ready-to-use middleware for building opportunistic activity and context recognition applications for different domains. One major characteristic and thus a crucial requirement of an opportunistic system is the identification and configuration of a group of sensors together with their (trained) machine learning technologies (e.g. feature extraction, classification, fusion) that

978-1-4244-8329-7/11/$26.00 ©2011 IEEE

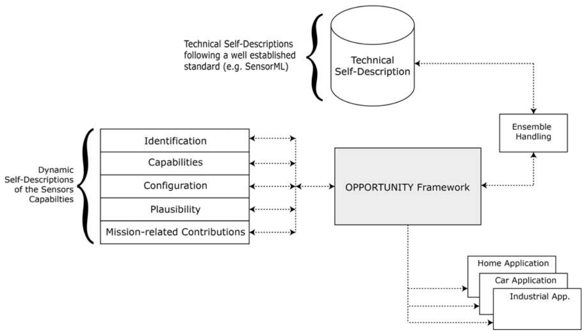


Fig. 1. The principle of sensor self-description in an opportunistic activity and context recognition system.

is best suited to achieve a given recognition goal. An example of such a goal in a kitchen is the recognition of the modes of locomotion (e.g. walking, standing, sitting, ...) of a subject. The sensor systems in the OPPORTUNITY Framework are able to describe themselves by providing a description file in a form of an XML document. In addition to the technical description of each sensor (e.g. weight, communication interface, processor and memory unit, ...), the self-description also includes a highly dynamic part which is mainly used for the ensemble configuration process. Figure 1 illustrates this principle of the sensor's self-descriptions (see [6] and [7] for further details).

In detail, the ensemble (re-) configuration process can be segmented into three steps:

i) *Identifying Candidates:* This is probably the easiest part of the ensemble configuration. A sensing mission that is derived from an abstract recognition goal contains machine-readable descriptions of what shall be recognized. Based on these descriptions, the ensemble configuration process starts by simply reading and parsing the *capabilities*-part (available in form of so-called *ExperienceItem*s, see Fig. 2) of the self-descriptions of the available sensor nodes to generate a set of possible candidates for the ensemble. This set of possible candidates is used in the next step - *Ensemble Structuring* - to define the sensor nodes that will be integrated and configured in the ensemble. The available sensor nodes are the set $S$, the set of candidates is called $S_C$, and is equal or a subset of all available sensors ($S_C \subseteq S$).

ii) *Ensemble Structuring:* This step considers the quality of the sensor nodes to be configured in the ensemble according to the recognition goal. From the candidates, the ensemble is structured with respect to the degree of fulfillment of (i) each individual sensor node, (ii) combinations of sensor nodes, and (iii) the experience of the contributions of the sensor node in past sensing missions. Therefore, the *capabilities*-part and the sensing mission-related part of the self-descriptions are used (stored as a so-called *ExperienceItem*). A metric is calculated for every sensor node according to the sensing mission indicating the sensor's performance value for this mission. This value describes to what extent the sensor (or a set of sensors) can contribute to the requirements of a sensing mission, and how valuable the sensor would be if integrated in the ensemble. Based on this value the set of sensors that is best suited for a given sensing mission can be configured into an ensemble. Let $S$ be the set of sensors that are available and capable to execute the purposes defined in the sensing mission (no matter to what extent). Let $S_E$ be the set of sensors that comprise the ensemble for a given sensing mission, whereas $S_E$ is a subset of $S_C$ or equal to $S_C$ ($S_E \subseteq S_C \subseteq S$). As already mentioned, the outcome is a set of (yet still not configured) available and connected sensor nodes that are best suited to execute a mission ($S_E$).

iii) *Ensemble Configuration:* Given the set $S_E$ of sensors that are best suited for the sensing mission. Within this step this ensemble has to be configured, which includes necessary network configurations to enable communication between the sensor nodes. Furthermore, a node may be of type *OnlineSensor* (a piece of software that delivers online available data and is also referred to as *Sensor*) [8], this may require a connection to a webservice or another online resource to be established. Other operations within this step deal with inline code (to be executed during ensemble configuration) that is provided in the self-description or invoking stubs of objects that are necessary for configuring a sensor node. Furthermore, the required ML techniques, such as feature extraction, classification, and fusion methodologies are configured and loaded within this step (e.g. a trained classifier to recognize a set of classes can be activated). The outcome of this step is a functioning and working ensemble of sensor nodes - the ensemble - that executes the required recognition goal to some extent.

One part of the dynamic self-description is referred to as *ExperienceItem*. Each self-description (for one sensor) can store an arbitrary number of *ExperienceItem*s. This piece of information stores the configuration and capabilities for a single sensor or an ensemble for different recognition goals. Figure 2 shows an example of a simple *ExperienceItem* for an accelerometer sensor that is located on a person's back and is able to recognize the class `WALK` with a *Nearest Centroid Classifier (NCC)* and a certain configuration (provided in the *ExperienceItem* as link to a *JSON* file). In Fig. 3 an example of such a JSON file is provided, which defines the configuration for a simple NCC classifier by providing a snapshot of the trained state. The necessary ML techniques, like the feature extraction, the classifier, and the unpacker (which extracts the required channel from the sensor data stream) are loaded dynamically in the *Ensemble Configuration step* by using the *reflection* capabilities of Java. This example defines that the accelerometer channel of a single sensor (the sensor to which this self-description belongs) with the defined configuration is capable of recognizing `WALK` at a DoF of $0.75$. The term `WALK` describes the associated class and is defined in the system's

```xml
<characteristics name="ExperienceItem">
 <swe:DataRecord>
  <swe:field name="trainingDataSet">
   <swe:Text>
    <swe:value>matthias_R_1_2.arff</swe:value>
   </swe:Text>
  </swe:field>
  <swe:field name="method">
   <swe:Text>
    <swe:value>NCC_Classifier</swe:value>
   </swe:Text>
  </swe:field>
  <swe:field name="sensors">
   <swe:DataRecord>
    <swe:field name="count">
     <swe:Quantity>
      <swe:value>1</swe:value>
     </swe:Quantity>
    </swe:field>
    <swe:field name="sensorList">
     <swe:DataRecord>
      <swe:field name="sensor">
       <swe:DataRecord>
        <swe:field name="type">
         <swe:Text>
          <swe:value>ACC</swe:value>
         </swe:Text>
        </swe:field>
        <swe:field name="location">
         <swe:DataRecord>
          <swe:field name="Object">
           <swe:Text>
            <swe:value>Person</swe:value>
           </swe:Text>
          </swe:field>
          <swe:field name="Position">
           <swe:Text>
            <swe:value>BAC</swe:value>
           </swe:Text>
          </swe:field>
         </swe:DataRecord>
        </swe:field>
       </swe:DataRecord>
      </swe:field>
     </swe:DataRecord>
    </swe:field>
   </swe:DataRecord>
  </swe:field>
  <swe:field name="labellist">
   <swe:DataRecord>
    <swe:field name="WALK">
     <swe:Text definition="dof">
      <swe:value>0.75</swe:value>
     </swe:Text>
    </swe:field>
   </swe:DataRecord>
  </swe:field>
  <swe:field name="jsonConfiguration">
   <swe:Text>
    <swe:value>mj_BAC_ACCEL_MEANVAR_NCC.json</swe:value>
   </swe:Text>
  </swe:field>
  <swe:field name="requiredFeature">
   <swe:Text>
    <swe:value>FX_LocomotionAccFeatures</swe:value>
   </swe:Text>
  </swe:field>
 </swe:DataRecord>
</characteristics>
```

Fig. 2. An exemplary ExperienceItem as part of a sensor self-description.

ontology that acts as knowledge base and provides a vocabulary and also specifies the semantic relations between the terms for an application. In our test and evaluation implementation we have defined an ontology that provides the terms and relations of the aforementioned kitchen/breakfast scenario (the *OPPORTUNITY Dataset*, see [9] and [10]). *ExperienceItem*s

```json
{
  "centroids" : [
    [-8.4965,-2.2945,2.5368,0.29659],
    [-9.2508,-2.3725,1.7388,1.3654],
    [-9.3933,-1.9461,1.0396,0.13042],
    [-4.4928,7.9204,-2.242,0.13042]
  ],
  "centroid_labels" : [3,1,5,4],
  "number_of_instances" : [1310,433,476,95],
  "cloud_size" : [
    [1.0241,0.89795,1.8205,0.2136],
    [0.32604,0.56007,0.71504,0.55195],
    [0.26634,0.57792,1.1387,0.11843],
    [1.1746,0.56141,1.5778,0.2331]
  ]
}
```

Fig. 3. An example for a JSON file which provides the configuration of a classifier used for activity recognition.

which provide the configuration details of an ensemble that consist of more than one sensor (thus makes use of multi sensor fusion) is very similar to the example in Fig. 2. Besides the count of sensors and the required channels, it additionally contains information about the fusion method and its training configuration (if necessary). How the DoF is calculated, and how the changes in the sensor data-stream can be detected and used in form of a Trust Indicator to re-configure an ensemble is explained in the following section.

### *B. Quantification and Application of Ensemble Capabilities*

As the delivered environmental sensor data can change due to faulty, shifted, or disappearing sensors (e.g. a sensor node might run out of power) a configured and running ensemble is not meant to be a fixed configuration over an extended period of time. As anomalies in the sensor data can be detected, the reconfiguration of ensembles at runtime is an important requirement in an opportunistic system. The initial DoF is more or less a static value that is calculated during the training phase of a sensor for a defined recognition chain. The information about how well such a sensor configuration/recognition chain (consisting of an unpacker, a feature extraction (FE) method, a classifier, and sensor fusion) is suited to recognize a class is stored in an *ExperienceItem* as part of the sensor's self-description. The initial DoF is calculated from the recognition rate, the confidence, the precision and the accuracy in the training phase which can be extracted by comparing the predicted class against the ground truth. The problem of dynamic changes of this DoF value is that within a running activity and context recognition system one usually does not have the ground truth available. Therefore, this generation and dynamic changing of the DoF value has to work according to the ML-technologies that are available at runtime and does not rely on ground truth classes. In [11] a methodology is presented to detect anomalies in the classifier recognition chain of a fusion method. Whenever a classifier delivers faulty and anomalous data, the system recognizes this and quantifies the amount by which the sensor data has changed. This value is also referred to as the *Trust Indicator* (TI) of a sensor. The anomaly detection algorithm (see [11] for further details) relies on the Mahalanobis distance [12] between the classifier

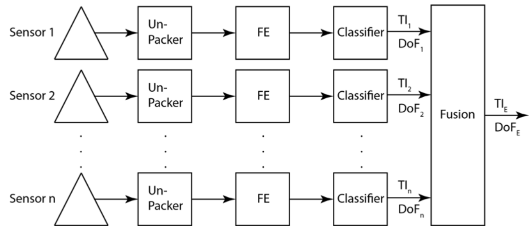


Fig. 4. Configured Recognition Chains together with the *Trust Indicator* (TI) and *Degree of Fulfillment* (DoF) values.

output classes and the fusion output. The system therefore can detect anomalies for every single sensor, with respect to its recognition chain. This results in the *Trust Indicator* value ($TI \in [0...1]$), where 1 means that the classifier output is plausible at 100%.
The outcome of the anomaly detection algorithm is used to quantify the ensemble's ability to recognize a goal depending on the plausibility of the configured sensing devices. The following values and metrics are used for this calculation (see Fig. 4 for illustration):

- $n$ = number of Recognition Chains ($RC$), whereas a RC consists of a single sensor and its corresponding unpacker, a feature extraction unit, and a classifier.
- $RC_s$ = Recognition Chain of sensor $s$.
- $DoF_E$ = Degree of Fulfillment of Ensemble (given by the fusion method and its n $RC$s; also calculated in the training phase, similar to the DoF of a single sensor).
- $TI_s$ = Trust Indicator value for Sensor $s$.
- $TI_E$ = Trust Indicator value for the Ensemble.
- $DoF_{ER}$ = dynamic DoF of Ensemble as calculated at runtime.

We calculate the $DoF_{ER}$, the runtime indication of the ensemble's goal recognition capabilities, as follows:

$$DoF_{ER} = DoF_E * \{TI'_E | TI''_E\}, \tag{1}$$

where either $TI'_E$ or $TI''_E$ is picked. Both values are calculated depending on the TI values of the sensors and the DoF of the ensemble:

$$TI'_E = min(TI_i), \tag{2}$$

and

$$TI''_E = \left( \frac{\sum_{i=1}^{n} TI_i}{n} \right) \tag{3}$$

The difference between $TI'_E$ and $TI''_E$ reflects to what extent the sensor with the smallest TI influences the overall result based on how reactive the system should behave on a single sensor's TI change. If for example an ensemble consists of a large number of recognition chains, one or two faulty sensors might not influence the $DoF_{ER}$ significantly. The resulting value ($DoF_{ER}$) can be compared to the initial $DoF_E$ of the ensemble to react if there is a significant difference between the two values and to reconfigure the ensemble (e.g. by removing the faulty recognition chain, or even by configuring a totally new ensemble) if necessary. This approach treats all sensors as the same, other approaches may take into account the DoF of each sensor in order to re-compute the whole DoF ($DoF_E$) upon decrease of individual TIs and/or reconfiguration. In the following section we provide tests and results that show (i) the application of the quantification of goal recognition capabilities and the ensemble (re-) configuration in the OPPORTUNITY Framework, and (ii) the difference between $TI'_E$ and $TI''_E$ and the resulting $DoF_{ER}$.

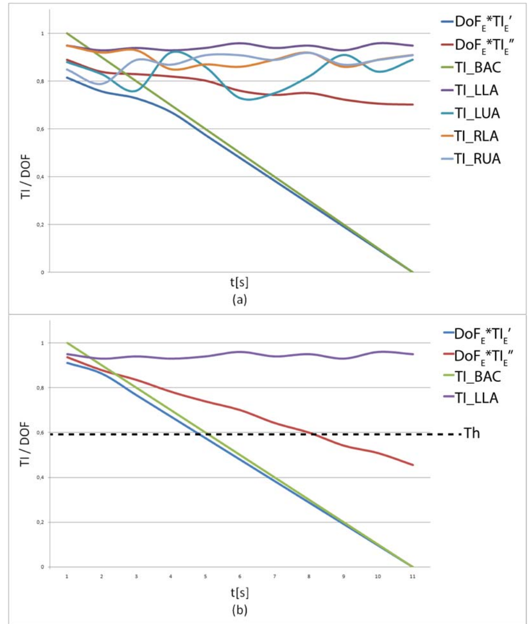


Fig. 5. Results of the dynamic calculation of the DoF according to detected anomalies in two exemplary setups with five sensors (a) and two sensors (b) as TI_BAC is artificially reduced.

## III. Evaluation and Testing

To test the approach of quantification of goal recognition capabilities in the framework we use a subset of the OPPORTUNITY dataset, namely data from an XSens Xbus Kit consisting of 5 on-body MTx sensor systems (each sensing drift-free 3D orientation, 3D acceleration, 3D rate of turn (rate gyro), and 3D earth-magnetic field). This earlier data set is provided in the OPPORTUNITY Framework (via *PlayBackSensors*, see [8]) and works with the modes of locomotion (MoL) (`WALK`, `SIT`, `STAND`, `LIE`, `STAIRS-UP`, and `STAIRS-DOWN`) as the recognition goal. The 5 sensors are located on the left/right lower/upper arm and on the back of the test subject (respectively LLA, RLA, LUA, RUA, and BAC in Figure 5).
The first test compares the two different calculation approaches for the $DoF_{ER}$. Therefore, we use 5 recognition chains (one for each sensor) consisting of the accelerometer channel of the sensor, a feature vector that uses mean and variance, a trained QDA (Quadratic Discriminant Analysis) Classifier that is able to recognize the MoL and a DecisionTemplate Fusion method that calculates the resulting class. To compare the two resulting values the TI value of the back sensor is reduced linearly until it reaches a value of 0.0 (i.e. we simulate a faulty sensor

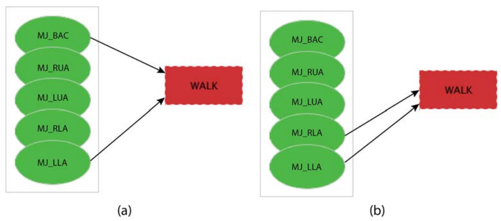


Fig. 6. Illustration of the ensemble re-configuration process in the OPPORTUNITY Framework.

by manipulation of the data-samples). Fig 5 (a) shows the resulting value of the $DoF_{ER}$ for both calculation approaches. As expected, the calculation approach that uses $TI'_E$ is much more affected by a single sensor's TI value change in contrast to the approach that uses $TI''_E$.
The second test demonstrates the reconfiguration capabilities. As shown in Fig. 5 (b) two sensors (the back sensor and the left lower arm sensor) are used (both are used with the same recognition chain as before). Again, the TI value of the back sensor is reduced linearly. The threshold ($Th$) for the system to react and reconfigure an ensemble was set to 0.6 of the $DoF_{ER}$ value (calculated with $TI''_E$). In Fig. 6 the reconfiguration process is illustrated. This Figure shows the available sensing devices in the environment and the dataflows. On the left hand side (a) the back and left-lower arm sensors are used together to recognize the class `WALK`. As the TI of the back sensor drops and the overall $DoF_{ER}$ drops below the threshold (as illustrated in Fig. 5 (b)), the system configures another ensemble (illustrated in Fig. 6 (b)). This new ensemble (left-lower arm and right-lower arm) has a lower initial value for $DoF_E$ but as the back sensor is faulty, the overall calculated value for the new ensemble is higher than the value for the ensemble including the faulty back sensor, and thus is configured for the recognition goal `WALK`.

## IV. Conclusion

In this paper we have introduced the concept of dynamic configuration of sensing ensembles based on sensor self-descriptions, which is the best available set of sensors for a recognition goal. To quantify the capabilities of an ensemble, we introduce the terms *Degree of Fulfillment (DoF)* and *Trust Indicator (TI)*. Furthermore, we have shown how anomaly detection of sensor data can be used in an opportunistic activity and context recognition system to dynamically quantify the goal recognition capabilities of a configured ensemble. Based on a DoF, that is calculated at the training phase of a classification or fusion method according to one or more output classes, by comparing the resulting classes with the ground truth we implemented an approach that uses anomaly detection (e.g. faulty or shifted sensors) in form of the TI to quantify the capabilities of a configured set of sensors for executing a recognition goal at runtime. According to a calculated DoF that is dependent of the TI values, the system recognizes significant changes in the sensor data and is able to (re-) configure the sensor ensemble if necessary.

## Acknowledgment

The project OPPORTUNITY acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225938.

## References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag, 1999, pp. 304–307.

[2] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.

[3] D. Roggen, K. Förster, A. Calatroni, A. Bulling, T. Holleczek, G. Tröster, P. Lukowicz, G. Pirkl, D. Bannach, A. Ferscha, A. Riener, C. Holzmann, R. Chavarriaga, and J. del R. Millán, "OPPORTUNITY: activity and context awareness in opportunistic open-ended sensor environments," in *Poster at the 1st European Future Emerging Technologies Conference (FET 2009)*, Prague, Czech Republic, April 2009.

[4] D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Tröster, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, and J. del R. Millán, "OPPORTUNITY: Towards opportunistic activity and context recognition systems," in *Proceedings of the 3rd IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2009)*. Kos, Greece: IEEE CS Press, June 2009.

[5] R. Chavarriaga, J. del R. Millán, H. Sagha, H. Bayati, P. Lukowicz, D. Bannach, D. Roggen, K. Förster, A. Calatroni, G. Tröster, A. Ferscha, M. Kurz, and G. Hölzl, "Robust activity recognition for assistive technologies: Benchmarking machine learning techniques," in *Workshop on Machine Learning for Assistive Technologies at the Twenty-Fourth Annual Conference on Neural Information Processing Systems (NIPS-2010)*, December 2010.

[6] M. Kurz, A. Ferscha, A. Calatroni, D. Roggen, and G. Tröster, "Towards a Framework for opportunistic Activity and Context Recognition," in *12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010), Workshop on Context awareness and information processing in opportunistic ubiquitous systems, Copenhagen, Denmark, September 26 - 29, 2010*, September 2010.

[7] M. Kurz, "Goal-Driven opportunistic Sensing," in *12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010), Doctoral Colloquium, Copenhagen, Denmark, September 26 - 29, 2010*, September 2010.

[8] M. Kurz and A. Ferscha, "Sensor Abstractions for opportunistic Activity and Context Recognition Systems," in *5th European Conference on Smart Sensing and Context (EuroSSC 2010), November 14-16, Passau Germany*. Berlin-Heidelberg: Springer LNCS, November 2010, pp. 135–149.

[9] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, F. Wagner, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and J. del R. Millán, "Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments," Helsinki, Finland, May 2010. [Online]. Available: http://vimeo.com/8704668

[10] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. del R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany*. IEEE Computer Society Press, June 2010.

[11] H. Sagha, J. del R. Millán, and R. Chavarriaga, "Detecting Anomalies to Improve Classification Performance in Opportunistic Sensor Networks," in *7th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing*, 2011.

[12] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings National Institute of Science, India*, vol. 2, no. 1, April 1936, pp. 49–55. [Online]. Available: http://ir.isical.ac.in/dspace/handle/1/1268