Detecting Anomalies to Improve Classification Performance in Opportunistic Sensor Networks

Hesam Sagha, José del R. Millán and Ricardo Chavarriaga
Defitech Foundation Chair in Non-Invasive Brain-Machine Interface
Center for Neuroprosthetics, Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
{*hesam.sagha,jose.millan,ricardo.chavarriaga*}*@epfl.ch*

***Abstract*—Anomalies and changes in sensor networks which are deployed for activity recognition may abate the classification performance. Detection of anomalies followed by compensatory reaction would ameliorate the performance. This paper introduces a novel approach to detect the faulty or degraded sensors in a multi-sensory environment and a way to compensate it. The approach considers the distance between each classifier output and the fusion output to decide whether a sensor (classifier) is degraded or not. Evaluation is done on two activity datasets with different configuration of sensors and different types of noise. The results show that using the method improves the classification accuracy.**

***Keywords*-Activity recognition, classifier fusion, anomaly detection, intelligent sensor nodes**

I. Introduction

Sensor fault or degradation detection is a critical issue in the field of sensor networks and wireless sensor networks [1], [2]. The degradation or fault may come from loss of energy, change of the sensor position or due to environmental fluctuations. As an example, in the field of activity and context recognition using wearable sensors [3] degradation may come from the slippage or rotation of on-body accelerometer sensors. These changes will be reflected on the sensed data, and consequently on the overall recognition performance. We propose a method for automatic detection of changes in sensor networks for activity recognition. Using a classifier fusion approach, we show that automatic detection and removal of sensors with anomalous behavior leads to graceful performance degradation with respect to increasing levels of noise.

There are many studies regarding fault detection in control-based systems where the goal is to detect abnormal sensors. For that, one may extract a model of the system and detect faults based on residual error signal [4]. In more complex systems we can create the input-output model of the system using regression methods and detect potential faults when there is a change in the estimated parameters [5]–[7]. Cumulative Sum (CUSUM) is another approach that detects changes over time in the distribution of a measured variable [8], [9]. These methods regard only one stream of signals and they decide whether it is faulty or not. However, in a sensor network, one of them may not be faulty but its behavior can change with respect to the other parts in the network (e.g. sliding an on-body sensor, in this case sensor works normally but sensory reading may change with respect to the other sensors). There are approaches which recognize whether sensed data belongs to a distribution or not; Chandola et al. surveyed different methods on it for anomaly detection in data [10]. However, these methods focus on the recognition of faults in the whole pattern –corresponding to the feature vector built using all the sensors– therefore, if one or few sensor fails, the whole sensed data is considered as faulty.

We propose that changes in the sensor behavior with respect to other sensors in the network can be used to detect anomalies. These changes can be estimated based on the correlations measured at different levels of the recognition chain, i.e. raw data or individual features computed for each sensor. Moreover, if the *classifier fusion* approach is used, correlations can be also computed between the outputs of individual classifiers. We choose this last approach (Fig 1) because of the following advantages:

1) Each sensor comes up with a decision independent of the other sensors. Therefore failure of one sensor will not affect the classification of others.
2) This allows for the removal of individual classifiers without requiring to re-train the remaining classifiers. Moreover, there exist fusion techniques that can cope with removal without recomputing the parameters (e.g. Majority voting, Naive Bayesian fusion or Dempster Shafer).

The rest of this paper is structured as follows: Section II explains the proposed method, then we describe the experiment details in section III and finally the results of evaluation of the method on two activity datasets are provided in section IV.

II. Method

We propose a method for detection of anomalous sensor behavior in sensor networks. The rationale of the method is to characterize the relationship between each sensor and the

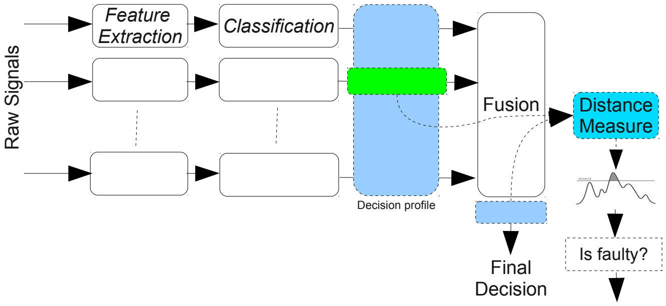


Figure 1. Schematic representation of the method. Detection process of anomalous sensors takes place at the fusion level by computing distance between corresponding classifier output and the fusion output.

overall network behavior. This relationship can be estimated at any level of the recognition chain –i.e. raw signals, feature vectors and classifier output. We propose to perform this operation at the classifier level because the values from all channels at this level are probability values, and is independent of chosen features and sensor data while at the other levels the values may be inhomogeneous using different types of sensors. To this end, we can compare the output of an individual classifier and the outcome of the fusion and whenever the difference among the two exceeds a particular threshold, the corresponding sensor is considered as faulty and removed from the network. The schematic representation of the method is shown in Fig. 1.

We assume that the classifiers output as well as the fusion outcome are probabilistic output vectors (soft labels). Then anomalous behavior is detected based on the distance between these two vectors. We use Mahalanobis [11] distance which is more robust against classifier bias with respect to Euclidean distance. This distance is computed as follows,

$$\mathcal{D}_{sc'} = (\mathbf{o}_s - \mathbf{f})^T \Sigma_{sc'}^{-1} (\mathbf{o}_s - \mathbf{f}), \qquad (1)$$

where $\mathbf{o}_s$ is the output of classifier s and $\mathbf{f}$ is the outcome of the fusion. $c' \in [1..C]$ is the recognized class after classifier fusion. The covariance matrix, Σ_{sc} represents the correlation between the classifier s and the fusion output, for class c. This matrix is estimated from the training data set using the correctly classified samples.

$$\Sigma_{sc} = E((o_s^c - f^c)^T (o_s^c - f^c)) \qquad (2)$$

where o_s^c and f^c are the output of the classifier s and the output of fusion for the specific class c, respectively and $E(.)$ is the mathematical expectation.

During system operation a classifier whose distance to the fusion output exceeds a given threshold is considered as faulty. Thresholds were set individually for each classifier and class such that $\Theta_{sc} = k\ max(\mathcal{D}_{sc})$, where $k > 0$, and $max(\mathcal{D}_{sc})$ is computed on the training set. The value of k can be chosen according to the distribution of distances, in the reported experiments it was chosen empirically to 4. Notice that removal of a sensor may change the outcome of the fusion process. For this reason, we apply the process iteratively where we use the fusion outcome with the current available sensors. If the distance of one classifier exceeds the threshold, the corresponding sensor is removed from the network, and the fusion of the remaining sensors is performed. We compare then the distances to the re-computed output and keep removing faulty sensors until no one exceeds its threshold.

In order to have a better estimation of the sensor behavior, at time t we use the average distance $\tilde{\mathcal{D}}_{sc}$ over a window of n preceding samples before comparing with the threshold. Moreover, we also fix an upper bound in the distances to avoid large values. A unique bound was set empirically for all classes and sensors.

This leads us to the following algorithm:

```
begin
  $\Sigma_{s,c} \equiv$ Covariance matrix for classifier $s$ and class $c$
  $\Theta_{sc} \equiv$ Distance threshold for classifier $s$ and class $c$
  $nS \equiv$ minimum number of sensors for fusion
  $\tilde{\mathcal{D}}_{sc} \equiv$ Average distance over specified window
  $f(\mathcal{S}) \equiv$ Fusion of classifiers in set $\mathcal{S}$
  $\mathcal{H} \equiv$ set of healthy sensors (initially $\mathcal{H} = [s_1..s_n]$)
  $\mathcal{A} \equiv$ set of anomalous sensors (initially $\mathcal{A} = \phi$)
  while $nSensor > nS$
      $\mathbf{f} = fusion(\mathcal{H})$
      $c' = argmax_c(\mathbf{f})$
      Compute distance $\tilde{\mathcal{D}}_{sc'}\ \forall s \in \mathcal{H}$
      if $\tilde{\mathcal{D}}_{sc'} < \Theta_{sc}$
        exit
        else
          $\mathcal{A} \equiv$ append $argmax_s(\tilde{\mathcal{D}}_{sc'})\ to\ \mathcal{A}$
          $\mathcal{H} \equiv$ remove $argmax_s(\tilde{\mathcal{D}}_{sc'})\ from\ \mathcal{H}$
          $nSensor = nSensor - 1$
      end
  end
```

III. Experiment

We evaluated the performance of the algorithm on two activity datasets using different sensor configurations. Each dataset contains data from body mounted sensors while human subjects perform different actions. In order to emulate changes in the sensor network we artificially added different levels of noise and assessed the method performance to detect noisy channels, as well as the recognition performance.

The first dataset corresponds to a car manufacturing scenario; It contains data from 8 subjects performing 10 recording sessions each (except one subject who recorded 8 sessions only) [12]. The sensors are accelerometer, rate gyro, and magnetic sensors mounted on different parts of the body (hands, upper and lower arms, and chest). We grouped sensors into seven packages considering their physical locations, where each package contains a set of sensors (accelerometer, rate gyro and magnetic sensors). There are 20 classes of activity including open hood, close hood, open door, close door and etc. We used one-subject-out cross validation for evaluating the performance.

The second dataset, termed the *Opportunity dataset*, contains data for daily home activities in a breakfast scenario [3]. The data were recorded in a highly instrumented environment set up in a room with two doors, a kitchen and a table in the center. For the present simulations we performed classification based on five motion-jacket sensors while one subject performed sequential low-level actions (e.g. Open Fridge, Close door, Reach cup). We report classification performance of 24 classes. Each sensor is composed of accelerometer, rate gyro and magnetic sensors positioned on upper and lower arms and chest. Note that the sensors placed on the jacket and therefore they may move slightly during the recording, also at each run the sensors may not be precisely located at the same location as before. We tested the method on one subject using 3-fold cross validation.

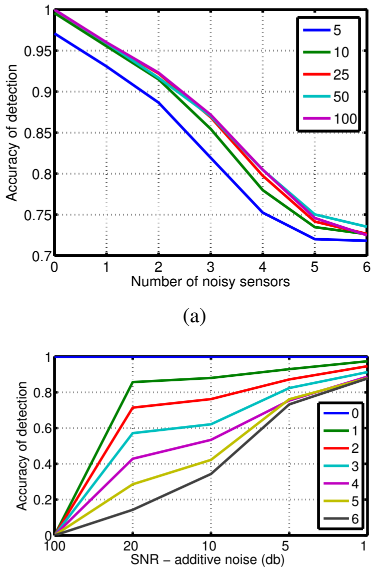
For each case we design an ensemble of classifiers whose decisions are combined using a classifier fusion technique. Linear Discriminant Analysis (LDA) classifier is chosen for classification using mean and variance as features. Since the method leads to the removal of anomalous sensors, a fusion technique that can cope with these removals should be used (e.g. Dempster-Shafer, Decision Template, Product Rule), we present results obtained using naive Bayesian fusion and Dempster-Shafer [13] for fusion.

Due to space limitations we only report the results using accelerometers and gyros; Similar results were obtained when including the magnetic sensors. We simulate different configurations of sensors –and corresponding number of classifiers– as follows:

1) *Config1*: Using accelerometers only. Each sensor feeds one classifier. Therefore, there are 7 classifiers for the car manufacturing dataset and 5 classifiers for the Opportunity dataset.
2) *Config2*: Each classifier receives input from an accelerometer and gyro located at the same position (i.e. sensor package). The number of the classifiers is same as in the Config1.
3) *Config3*: Using accelerometers and gyros but each *type* of sensor feeds a different classifier. Then, there are 14 classifiers for the car manufacturing dataset and 10 classifiers for the Opportunity dataset.

We used two different types of noise in the simulations: rotational and additive noise. In the former case, the level of rotation is randomly chosen between 0° and 90° with steps of 10°. In the case of additive noise, the Signal Noise Ratio (SNR) was varied between 100, 20, 10, 5, and 1 db. For each condition we perform 10 repetitions with different sets of noisy sensors, and at each repetition we compute the accuracy for detecting anomalous sensors as well as the activity classification accuracy. Detection accuracy is computed as the number of true detections,

$$acc = \frac{TruePositive + TrueNegative}{Number\ Of\ Sensors} \tag{3}$$



(a)

(b)

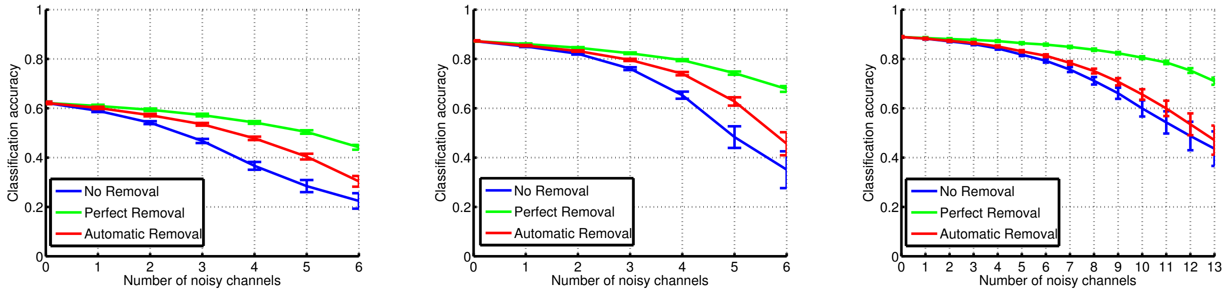
Figure 2. Recognition of anomalous sensors on the car manufacturing dataset (*Config1*; additive noise). 2(a) Effect of window length. 2(b) Effect of noise level for different number of noisy sensors.

In the experiments we chose the threshold coefficient,Θ_{sc} and upper bound coefficient empirically, and is kept the same for all the configurations of each dataset.

IV. Results

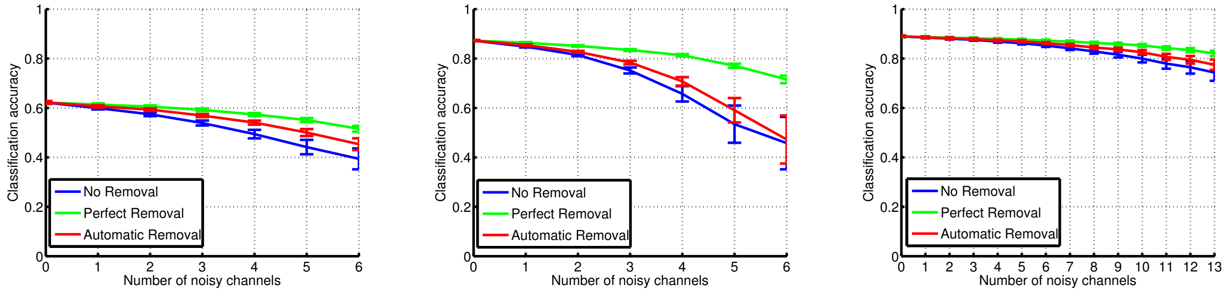
Fig. 2(a) shows the accuracy on detecting the noisy sensors on the car manufacturing dataset (*Config1*, additive noise). It can be seen that the proposed method is able to accurately identify the sensors to which additive noise was added; recognition accuracy decreases from $\succ$95% to $\sim$73% as the number of noisy sensors increases from 1 to 6. Since the method uses the average distance over a time window there may be trade off between the accuracy and the detection latency. Simulations show that recognition accuracy is rather constant with respect to the window size and similar performance is obtained when using windows of at least 25 samples (c.f. Fig. 2(a)). A window size of 50 samples was used in the remaining simulations presented in this paper.

Fig. 2(b) shows the accuracy of detection of anomalous channels with respect to the level of noise. Unsurprisingly, the performance of the detection decreases for lower levels of noise, since the added noise is comparable to the noise



(a) *Config1*:Acc (b) *Config2*:Acc+Gyro (packed) (c) *Config3*:Acc+Gyro (separate)

Figure 3. Classification performance –rotational noise– on the car manufacturing dataset.



(a) *Config1*:Acc (b) *Config2*:Acc+Gyro (packed) (c) *Config3*:Acc+Gyro (separate)

Figure 4. Classification performance –additive noise– on the car manufacturing dataset.

level inherently present in the data (please remind that we use data recorded during actual performance of human activities).

Upon detection of the faulty sensors, the corresponding classifiers are removed from the fusion. Fig. 3 and 4 show the classification accuracy in the car manufacturing scenario for rotational and additive noise, respectively. We compare the performance of the proposed model (*Automatic removal*), with the performance if no action is taken (*No removal*). As a comparison, we also report the performance when classifiers affected by noise are manually removed from the classification fusion (*Perfect removal*). In this case we remove the classifier only if it gives a different decision with respect to noiseless condition. So, this performance corresponds to the upper bound of a detection system. In all cases, *Automatic removal* outperforms the *No removal* condition showing that the proposed approach consistently yields more robust classification.

Fig. 5 shows the performance with respect to the level of noise and the number of noisy sensors using accelerometers, gyros and magnetic sensors in a packed mode. It is clear that *Automatic removal* performs better than *No removal* as the level of noise and number of affected channels increases.

To assess the independency of the approach with respect to the fusion method, we show another simulation of the configuration *Config1* using Dempster-Shafer classifier fusion. Obtained results, shown in Fig. 6, follow the same trend as with the Naive fusion where the *Automatic removal* performs better than the original sensor network.

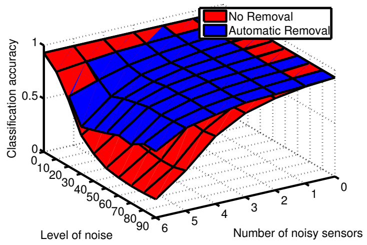


Figure 5. 3D view of classification accuracy on the car manufacturing dataset for different levels of rotational noise using accelerometers, gyros, and magnetic sensors on the car manufacturing dataset.

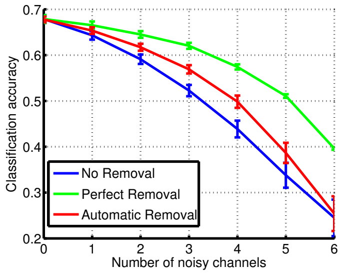
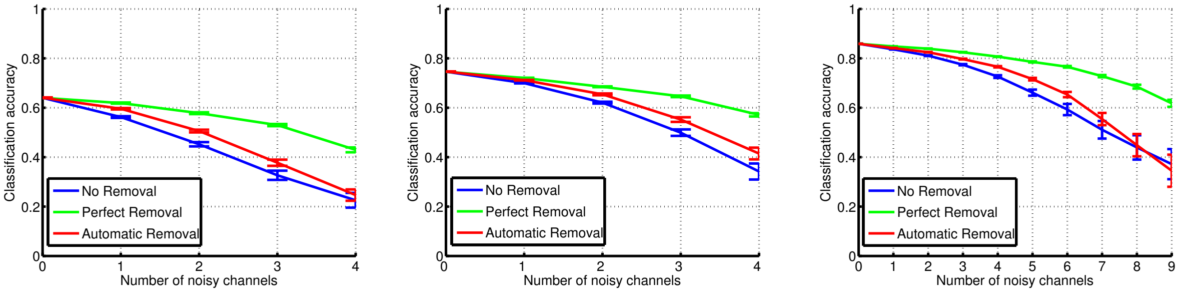
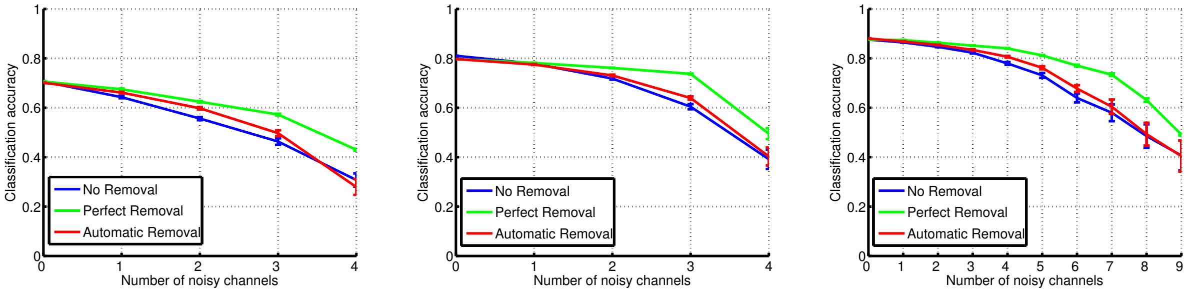


Figure 6. Classification accuracy on the car manufacturing dataset using Config1 and Dempster-Shafer classifier fusion.



(a) *Config1*:Acc (b) *Config2*:Acc+ Gyro (packed) (c) *Config3*:Acc+Gyro (separate)

Figure 7. Classification performance using Naive Bayesian fusion on the *Opportunity* dataset (rotational noise).



(a) *Config1*:Acc (b) *Config2*:Acc+Gyro (packed) (c) *Config3*:Acc+Gyro (separate)

Figure 8. Classification performance using Dempster-Shafer fusion on the *Opportunity* dataset (rotational noise).

Simulations using the Opportunity dataset show similar results. Fig. 7 and 8 show the results for different configurations on this dataset applying Naive Bayes fusion and Dempster-Shafer fusion, respectively. As before, for both fusion techniques, *Automatic removal* of anomalous sensors results in a more graceful degradation of performance when compared to the original sensor network.

While the algorithm iterates for removing faulty channels from the classifier fusion for each pattern, the output of the fusion may change. Table I provides the average number of iterations for each pattern for different configurations.

On average about one iteration is required for all configurations for small rotational noise. When the noise becomes larger, the method performs around three iterations in average in order to detect the anomalous sensors.

V. Conclusion

We have proposed a method to automatically detect anomalous channels in activity recognition networks. Upon recognition, these channels can be removed from the fusion process leading to a more robust recognition system. Simulated results using real data for activity recognition scenarios consistently show accurate recognition and graceful degradation with respect to different types of noise. The proposed method –tested on activity recognition problems– can also be applied to other multi-sensory classification problems.

The proposed approach has the advantage that it considers the relationship between classifiers (via fusion output) instead of considering only the data or the feature distribution of individual channels. In real situations, the characteristics of the data provided by a given sensor may not change, but its behavior with respect to the whole network may change, e.g. a wearable on-body sensor may be moved from one hand to the other; in this case the signal feature distribution may appear the same but its contribution for recognizing a given activity will drastically change with respect to other, unchanged sensors. In addition, related work on characterization of sensor behavior within a network suggests that this is better achieved at the fusion level than at the sensor levels [14].

The proposed method for detecting anomalies takes into account the outcome of the classifier fusion process. Therefore, it is more suited to be implemented at a centralized node (otherwise the outcome of the fusion process shall be sent back to each node increasing the communication

Table I
Average number of iterations for different configurations when the noise type is rotation.

	Config1		*Config2*		*Config3*	
Rotation level	$\leq 20°$	$> 20°$	$\leq 20°$	$> 20°$	$\leq 20°$	$> 20°$
Car manu.	1.18	3.34	1.06	3.30	1.12	3.10
Opportunity	1.35	2.28	1.35	2.81	1.24	3.58

overhead). Centralized implementations on sensor networks may send information back to anomalous sensors so that they stop transmitting their samples in order to reduce the power consumption.

It should be noticed that in the current implementation we compute the thresholds using the *correctly* classified samples on the training data. Although the use of all training samples may lead to a better estimation of the variability of the incoming data, this choice allows us to use only the data that has been observed to accurately contribute to good classification. Although, this may lead to false detections (i.e. healthy sensors recognized as anomalous) no difference was observed among the proposed method and the *No Removal* condition when tested without noise, suggesting that this effect was not significant.

In the current study the free parameters (i.e. the threshold and distance upper bound) were empirically set. However, more systematic ways can be used to estimate them such as cross-validation techniques. Future lines of research we plan to perform is the combination of the change detection with adaptation techniques that allow to overcome the changes in a supervised or unsupervised manner while keeping the changed sensors within the recognition systems.

VI. Acknowledgements

This work has been supported by the EU Future and Emerging Technologies (FET) contract number FP7-Opportunity-225938. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

References

[1] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "On-line fault detection of sensor measurements," in *IEEE Sensors*, vol. 3, no. 2, jun 2002, pp. 69–85.

[2] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[3] 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. R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Seventh International Conference on Networked Sensing Systems*, 2010.

[4] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601 – 611, 1976.

[5] R. Isermann, "Process fault detection based on modeling and estimation methods–a survey," *Automatica*, vol. 20, no. 4, pp. 387 – 404, 1984.

[6] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[7] P. Smyth, "Hidden markov models for fault detection in dynamic systems," *Pattern Recognition*, vol. 27, no. 1, pp. 149 – 164, 1994.

[8] B. F. J. Manly and D. Mackenzie, "A cumulative sum type of method for environmental monitoring," *Environmetrics*, vol. 11, no. 2, pp. 151–166, 2000.

[9] C. Alippi and M. Roveri, "An adaptive cusum-based test for signal change detection," in *ISCAS*. IEEE, 2006.

[10] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[11] 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

[12] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Computing Magazine*, pp. 42–50, 2008.

[13] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 3, pp. 418 –435, may. 1992.

[14] H. Sagha, J. d. R. Millán, and R. Chavarriaga, "A probabilistic approach to handle missing data for multi-sensory activity recognition," in *Workshop on Context Awareness and Information Processing in Opportunistic Ubiquitous Systems at 12th ACM International Conference on Ubiquitous Computing,*, Copenhagen, Denmark, September 2010.