# Detecting and rectifying anomalies in body sensor networks

Hesam Sagha, José del R. Millán, Ricardo Chavarriaga
*CNBI-STI-EPFL*
*Lausanne, Switzerland*
{*hesam.sagha, jose.millan, ricardo.chavariaga*}*@epfl.ch*

***Abstract*—Activity recognition using onbody sensors are prone to degradation due to changes on sensor readings. The changes can occur because of degradation or alteration in the behavior of the sensor with respect to the others. In this paper we propose a method which detects anomalous nodes in the network and takes compensatory actions to keep the performance of the system as high as possible while the system is running. We show on two activity datasets with different configurations of onbody sensors that detection and compensation of anomalies make the system more robust against the changes.**

***Keywords*-Activity recognition, classifier fusion, anomaly detection and compensation, intelligent sensor nodes, onbody sensor network**

## I. Introduction

Degradations and faults of sensor readings are one of the major problems in sensor networks resulting in decreases of performance [1]. Degradation of the sensor signal may be caused by changes in environmental conditions, change of the position (e.g. accelerometer), etc. Also, signal degradation in classification systems is not restricted to sensory reading signals, the behavior of the sensor may change with respect to the others in the network. For example, when an on-body accelerometer sensor rotates or slides, it may not affect the sensor reading characteristic but its behavior may change. Hereafter we call all types of degradation as 'anomalies'.

In order to cope with the anomalies one way is to detect its type and take a compensatory action to alleviate it. By knowing the type of anomaly it may be possible to compensate it [2] or estimate features which are robust against that type of anomaly [3]. For example in the case of accelerometers, knowing that a sensor has been rotated we can rotate back the signals and process them as before [4]. But detection of the anomaly type may not be feasible in real world applications where many unknown disturbances are involved. The other way is to use adaptive classifiers to keep the performance [5]. Other alternative is to estimate characteristics of the signals and test whether incoming signals differ from the estimated characteristics; in this case, any deviation suggests some types of degradation [6], [7]. As mentioned before, behavior changes cannot be detected using these approaches; for example in a walking scenario with some mounted accelerometers on feet, if a sensor is misplaced, it may still give similar signals but its relation to other sensors is not the same as before. Another approach – based on classifier fusion architecture– detects anomalies by comparing the outcome of single-channel classification with the overall fused decision [8]. We showed that anomalies can be detected efficiently and by removing the anomalous sensors from the fusion we could improve the classification accuracy.

The goal of this paper is to propose a method to rectify the anomalous part of the network by replacing their values with their corresponding expected values. It exploits the aforementioned detection mechanism with a probabilistic approach for imputing missing data [9]. The idea behind is to use the information of correlation between the classifiers to detect the anomalous behaviours of classifiers with respect to the other classifiers and rectify them. This helps the classifier fusion to use all the information which has been stored during training phase. We evaluate this combined approach on two activity datasets using different sensor configurations to empirically show the robustness of the approach in different situations.

## II. Methods

To rectify the anomalies in a classification problem we use two stages; on the first stage anomalous parts are recognized and then at the second stage we replace the corresponding values with meaningful ones. To do so, we use the classifier fusion architecture which allows more independence between sensors, i.e. sensors can be removed or replaced without change in other parts of the network.

### *A. Anomaly Detection*

In order to recognize anomalous parts in the network, one can estimate some statistics on the raw sensor signals and any deviation from them is considered as an anomaly. In this category Chandola et al. surveyed methods for anomalous data [1], but the methods only tell whether the whole pattern of data is anomalous or not, unable to distinguish exactly which part of the data is abnormal. CUmulative SUM (CUSUM) approach estimates up to 4th moment of the signal and detects changes of the distribution over time [6]. In problems where the goal is to classify signal from a large number of sensors, another approach is to recognize the changes in the classifier outputs [8]. The

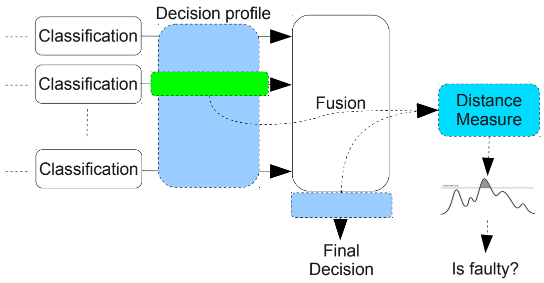


Figure 1. Detection of anomalies in the network. The blue boxes are the output of the classifiers and fusion.

method characterizes the relationship between each classifier (corresponding to one or several sensors) and the overall behavior of the network. At any time, the behavior of the network corresponds to the output labels of all classifiers. Doing so, the classifier fusion architecture provides the feasibility to define a measure for each classifier representing the distance between the classifier output and the fusion output. Here, we consider the cases which the classifier outputs are soft labels, i.e. a probabilistic vector, $\mathbf{o}_s$, representing the probability of the data vector to belong to each class. The constructed matrix of these vectors is called Decision Profile (DP) and is the size of $N \times C$ matrix, where $N$ is the number of classifiers and $C$ is the number of classes. During the training phase we estimate a threshold on the distance for each classifier and while running the system, if the measured distance exceeds the threshold it is considered as anomaly. It is empirically shown that the method is independent from classification and fusion method. The schematic representation of the method is shown in Fig. 1.

To compute the distance between classifier output and fusion output, the Mahalanobis distance [10] is used since it is more robust against classifier bias. We assume that the classifiers' output as well as the fusion outcome are probabilistic output vectors. This distance is computed as follows,

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

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

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

where $o_s^c$ and $f^c$ are the output of the classifier $s$ and the output of fusion for the specific class $c \in [1..C]$, respectively and $E(.)$ is the mathematical expectation. Note that the selection of only correct classified samples also helps to ignore the wrong decisions after fusion and it improves the performance.

Thresholds are 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 the maximum distance for class $c$ and classifier $s$ on the training set. When the distance between a classifier and fusion is larger than the corresponding threshold for the chosen class, we label the classifier as anomalous. If one classifier is labeled as anomalous we can remove it from the network, not to use it anymore.

When a classifier is removed, the final fusion decision may change. So, to detect all the anomalous parts we do the removal process iteratively till all the distances of the remaining classifiers are below the threshold or the minimum number of sensors is reached.

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

### B. Value Generation

Once the anomalous parts of the classification network have been discovered, one possibility is to remove them from the classification chain. The other alternative is to replace their values by other meaningful values. The replacement process can be at the signal level, feature level or classification level. In [9] it is shown that replacement at classification level in a classifier fusion architecture has better effect. At that level classifiers have made their own decisions and values are in the same space as they all correspond to class probabilities.

When the anomalous parts are detected we aim to replace the probabilistic classifier output vector for each of the anomalous channels. We call the matrix form of these vectors as $DP^{Test}(anomal)$, and we want to regenerate the values using the healthy parts, $DP^{Test}(hlty)$. To this end, a Gaussian distribution is estimated using the DPs of training data, and the anomalous values are replaced by the mean values of the conditional distribution, taking into account the current values of the available vectors. To do so, first we re-order DP to be a column vector $\overline{DP}$ of size $N * C$, and we estimate the covariance between each element.

Then, when encountering anomalous values, we infer and impute the mean $\mu_{a|b}$ of conditional distribution,

$$\overline{DP}^{Test}(anomal) = \mu_{a|b} = \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (x_b - \mu_b) \tag{3}$$

where $\mu$ and $\Sigma$ are the mean vector and covariance matrix of data computed on training data. $\mu_a$ and $\mu_b$ are subvectors of $\mu$ and they are the mean of the anomalous and healthy values, respectively. $x_b$ is the vector of values of the healthy classifiers, $\overline{DP}^{Test}(hlty)$, $\Sigma_{ab}$ is the covariance between anomalous values and healthy values, and $\Sigma_{bb}$ is the covariance between healthy values. This regeneration phase helps to bring back the lost information with the help of available information. These information will be used in the classifier fusion process.

### C. Computational workload

For the first step, to recognize the anomalies, there is at most $S-1$ iterations of computing distances, where $S$ is the number of classifiers. At each iteration we compute $S$ distances between each classifier and the fusion output. The cost of calculating of each distance is two vector subtractions with size $C$ (number of classes) and one multiplication of three matrices with sizes $1 \times C$, $C \times C$, and $C \times 1$. Finally the order of computation is $O(S^2C^3)$.

Once a set of sensors have been detected as anomalous, the inverse matrix $\Sigma_{bb}^{-1}$ should be computed once and after that the computation cost for each pattern is three matrix multiplication with the sizes $N_a \times N_b$,$N_b \times N_b$ and $N_b \times 1$, plus one matrix addition with size $1 \times N_a$ and one matrix subtraction with size $N_b \times 1$ where finally results to $O(N_a N_b^3)$. Here we are estimating a vector of size $C$ for each anomalous classifier, so the final computational order for this phase is $O(N_a N_b^3 C^4), \quad N_a + N_b = S$.

Note that these computations are done at the classifier output level and the pace of the whole process depends only on the number of decisions the classifiers make in an interval of time, not the sampling rate of individual sensors. i.e. the feature extraction method has a great role here.

Using classifier fusion architecture imposes to have a central node for final decision. The sensors provide data (as raw data, feature or classifier output in case of intelligent sensor nodes) for decision making and note that the whole mentioned workload are inside the center and sensor nodes are independent of that.

## III. Experiment

We evaluate the performance of the proposed algorithm on two activity datasets using different sensor configurations. For each case we design an ensemble of Gaussian Mixture Model (GMM) classifiers. Each GMM has two units per class and Bayesian fusion is used. It should be noticed that both the anomaly detection and imputation methods do not impose any specific type of classifier as long as it provides a probabilistic output.

The first dataset (*car manufacturing dataset*) corresponds to a car manufacturing scenario [11], [12], as shown in Fig. 2. Seven IMU sensor nodes are mounted on hand, torso, lower and upper arm of both hands. We use data from 8 subjects performing 10 recording sessions each (except one subject who recorded 8 sessions only). There are 20 classes corresponding to the activities like Open hood, Close hood, Open door, Close door and etc.

Figure 2. Car manufacturing scenario. Seven IMU sensor nodes are located on the body during recording data, [11].

The second dataset, termed the *Opportunity dataset*, contains data for daily home activities in a breakfast scenario [13]. Five IMU sensor nodes are located on back, lower and upper arm of both hands. The data were recorded in a highly instrumented environment set up in a room with three doors, a kitchen and a table in the center. For the present simulations we performed classification based on 24 low-level actions for one subject (like Open fridge, Close fridge, Open dishwasher,...). The subject performed activities in 5 sessions.

Each node contains 3-axis accelerometer, 3-axis rate gyro, 3-axis magnet sensors. In one configuration we assign a classifier to a group of the sensors which are physically together (*packed*) in the same node and in another configuration each classifier is assigned to individual sensor (*separate*). The former type assumes that the sensors which are physically located at the same place all are affected by the change, e.g. displacement, and the later type assumes that those sensors may not be affected with the same type of anomaly. For each type of configuration, we will simulate different sets of sensors. The first one uses accelerometer, the second one uses accelerometer and gyro and the last one uses all three types of sensors.

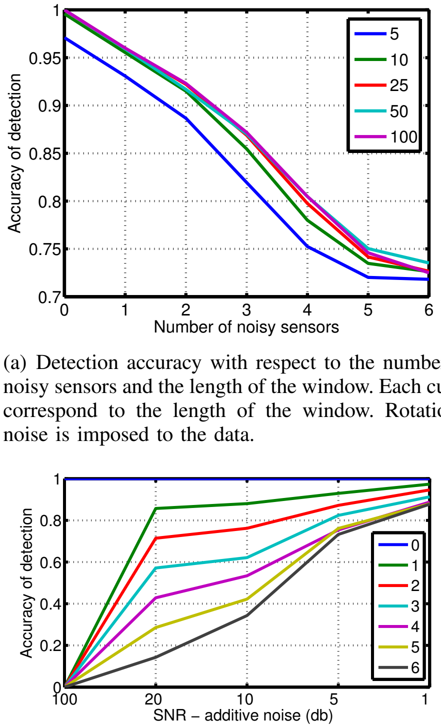
The simulated configurations are shown in the table I. Here we are using segmented data assuming there is an algorithm behind telling when is the starting and ending point of the actions and the extracted features are the mean and variance of the segmented signals. We simulated anomalies corresponding to two different types of noise: rotation or additive. 10 repetitions have been done for each configuration and type of noise. For each repetition we insert a type of anomaly to a number of randomly selected packages. To show the robustness across different sessions, data are evaluated based on one-session-out cross validation.

For rotational noise, the level of rotation is randomly chosen between -90° up to 90° degrees with steps of 10° in three dimensions. Note that, here we assume that rotation does not affect magnetic sensors. In the case of additive noise, the Signal Noise Ratio (SNR) was varied between 100, 20, 10, 5, and 1 db.

In the experiments the threshold coefficient, $\Theta_{sc}$, and upper bound coefficient, $\Gamma$, are set empirically to 4 and 5, respectively, and are kept the same for all the configurations

| Dataset | Opportunity | | Car manufacturing | |
|---|---|---|---|---|
| configuration | packed | separate | packed | separate |
| Acc | *config11*(5) | - | *config11*(7) | - |
| Acc+Gyro | *config12*(5) | *config22*(10) | *config12*(7) | *config22*(14) |
| Acc+Gyro+Magnet | *config13*(5) | *config23*(15) | *config13*(7) | *config23*(21) |

Table I
NAME OF THE CONFIGURATIONS AND THE NUMBER OF CLASSIFIERS FOR EACH ONE FOR BOTH DATASETS.



(a) Detection accuracy with respect to the number of noisy sensors and the length of the window. Each curve correspond to the length of the window. Rotational noise is imposed to the data.

(b) Detection accuracy with respect to the level of noise. Each curve corresponds to the number of noisy sensors.

Figure 3. Anomaly detection accuracy

of each dataset.

## IV. RESULTS

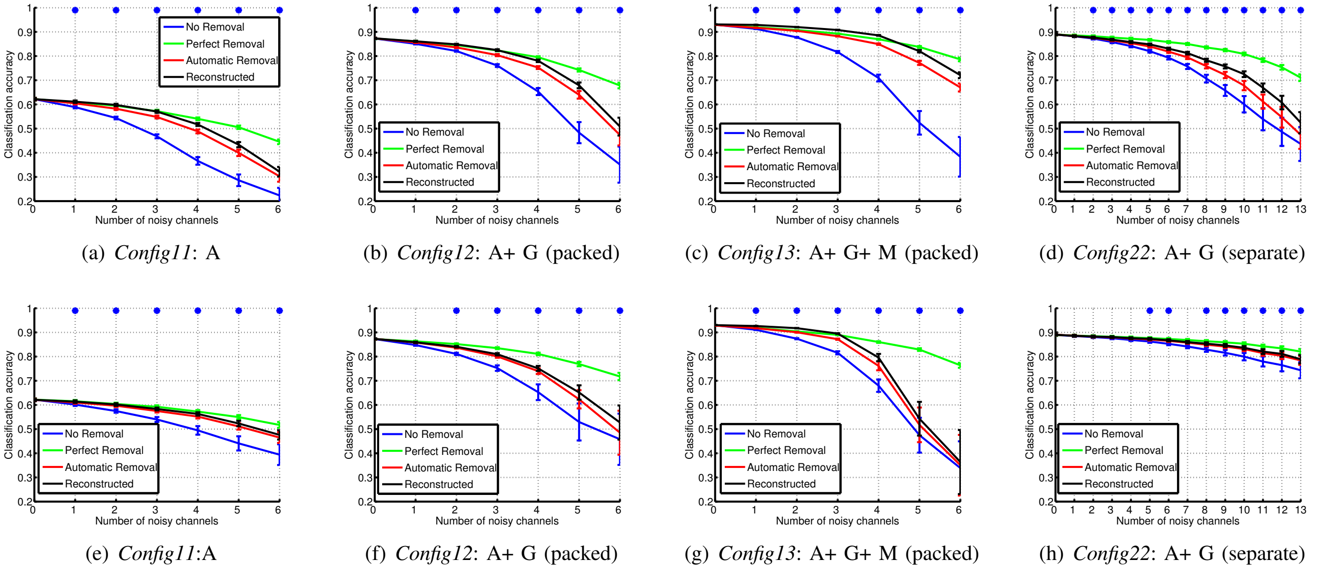
We first report the accuracy on detecting the anomalous sensors, Fig 3(a). We assess this with respect to the number of noisy sensors as well as length of the averaging window. Unsurprisingly, the accuracy decreases when the number of noisy sensors increases. Also it can be seen that an averaging window of 50 samples yields already the maximum performance. On the other hand, Fig 3(b) shows how the detection process is affected by the level of noise. With low level of noise, it is not able to detect well which is a good point since low level of noise doesn't harm classification performance.

We compare four different conditions for each configuration:'No Removal' refers to accuracy based on no anomaly detection and compensatory reaction, while 'Automatic Removal' means removing anomalous classifiers from the fusion without imputing any value. 'Perfect Removal' is the case where we know which classifiers are faulty and we remove them from the chain. In a real scenario it is not possible to do so without any supervision. This case somehow shows the upper bound of performance.

The classification accuracies for the car manufacturing dataset are shown in Fig. 4 and for Opportunity datasets in Fig. 5 for both types of noise. When there is no noise the classification performance for Skoda for *config12*, *config13*, *config22* are about 87%, 93% and 89% respectively. Also for Opportunity dataset for *config12* , *config22* and *config23* it is about 74%, 85% and 95% respectively. Unsurprisingly, performance increases when a larger number of sensors is used for classification. This performance consistently drops as noise is added to the sensor readings. Removal of sensors detected as anomalous (*red traces*) compensate for such noise yielding graceful performance degradation for all configurations. Moreover imputation of the estimated values (*black traces*) significantly improves the classification accuracy with respect to the removal process only. Statistical significance is shown with asterisks on Figs 4-6 (using t-test with $p < 0.05$)

These results show that detecting the anomalies and rectifying their values using the mentioned algorithms improves the recognition accuracy. Reconstruction always is as good or better than only removing the sensor. The method is useful when there is a running system without supervision and continuously classifies data and there is a set of sensors where each sensor corresponds to a classifier and by classifier fusion scheme the final decision is achieved.

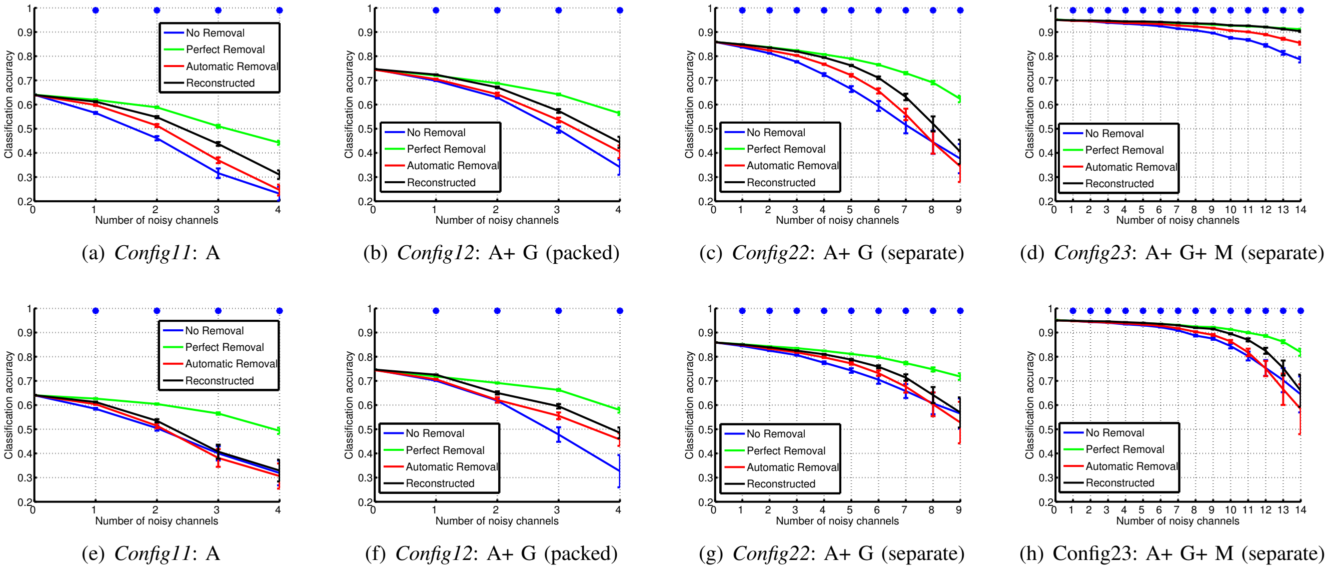
In order to assess the performance when other types of fusion mechanism are used, we simulated *config13* for car manufacturing dataset with Dempster-Shafer, majority voting and sum rule fusion. In case of majority voting, after reconstruction of decision profile we pick the index of the maximum value of each probability vector for each classifier. Figure 6 shows the results. We found that with the fusion methods which use hard labels (e.g. majority voting, Bayesian fusion), the level of improvement after reconstruction is higher, while for the others it is not but it never degrades the performance. That is because the conditional distribution approach is not able to construct exactly the desired decision profile but still the generated



(a) *Config11*: A (b) *Config12*: A+ G (packed) (c) *Config13*: A+ G+ M (packed) (d) *Config22*: A+ G (separate)

(e) *Config11*:A (f) *Config12*: A+ G (packed) (g) *Config13*: A+ G+ M (packed) (h) *Config22*: A+ G (separate)

Figure 4. Classification performance after imposing up) rotational noise down) additive noise on the car manufacturing dataset. A:Acc, G:Gyro, M:Magnetic



(a) *Config11*: A (b) *Config12*: A+ G (packed) (c) *Config22*: A+ G (separate) (d) *Config23*: A+ G+ M (separate)

(e) *Config11*: A (f) *Config12*: A+ G (packed) (g) *Config22*: A+ G (separate) (h) Config23: A+ G+ M (separate)

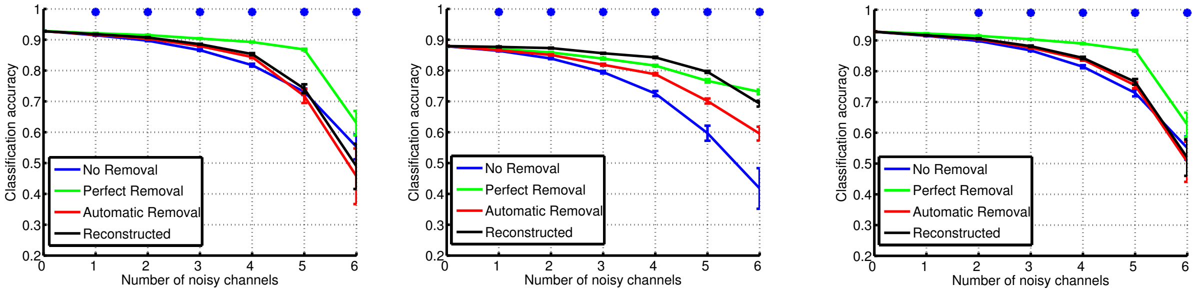
Figure 5. Classification performance after imposing up) rotational noise down) additive noise on the Opportunity dataset. A:Acc, G:Gyro, M:Magnetic

values obey the same behavior in their order, so the index of the desired maximum value in each vector remains the same.

## V. Conclusion

In real applications, the performance of activity recognition may be degraded because of some anomalies in the data. We introduced a method to detect and compensate such anomalies in classifier fusion architecture. The detection process is based on the distance between the decision of each classifier and the final decision of classifier fusion. The compensation is performed at the classifier level where each classifier has made a decision and it is based on missing value compensation. The results show that using detection and compensation process can improve the performance of the system online. If the fusion and detection process is done in a central station, upon detection of the faulty sensors and turning them off and put the burden of compensation on the centralized fusion, we can save the energy in the whole system.

Later, we want to investigate the technical requirements of implementing this particular method on (wireless) body sensor networks, as well as assessing its performance in an online manner.



(a) Dempster-Shafer

(b) Majority Voting

(c) Sum rule

Figure 6. Result of rectification of decision profile with differenct fusion methods with $config13$ on car manufacturing dataset.

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

[2] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 20–29.

[3] K. Förster, P. Brem, D. Roggen, and G. Troster, "Evolving discriminative features robust to sensor displacement for activity recognition in body area sensor networks," in *5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2009, pp. 43 –48.

[4] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket–an experimental study," in *IEEE Int Conf Pervasive Computing and Communications (PerCom 2010)*, Mannheim, Germany, 2010.

[5] K. Förster, D. Roggen, and G. Tröster, "Unsupervised classifier self-calibration through repeated context occurences: is there robustness against sensor displacement to gain?" in *Proceedings of the 13th International Symposium on Wearable Computing*. IEEE Computer Society, 2009, pp. 77–84.

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

[7] ——, "Just-in-time adaptive classifiers, part I: Detecting nonstationary changes," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1145 –1153, 2008.

[8] H. Sagha, J. d. R. Millán, and R. Chavarriaga, "Detecting anomalies to improve classification performance in an opportunistic sensor network," in *7th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, PerSens 2011*, Seattle, March 2011.

[9] ——, "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*, 2010.

[10] P. C. Mahalanobis, "On the generalized distance in statistics," 1936, pp. 49–55.

[11] 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.

[12] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from onbody sensors: Accuracy-power trade-off by dynamic sensor selection," *Wireless Sensor Networks*, pp. 17–33, 2008.

[13] 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.