Repair Assessment of Sensor Node Failures for Activity Detection*

Krasimira Kapitanova, Enamul Hoque, Jingyuan Li, Daniele Alessandrelli*, John A. Stankovic, Sang H. Son, and Kamin Whitehouse

University of Virginia, Charlottesville, USA
{krasi, eh6p, jl3sz, stankovic, son, whitehouse}@virginia.edu
* Scuola Superiore Sant'Anna, Pisa, Italy
d.alessandrelli@sssup.it

Abstract. Accurate and reliable detection of activities of daily living (ADL) is extremely difficult and it becomes even harder due to sensor node failures caused by hardware degradation, node displacement, inaccurate readings, and environmental changes. Once a fault is detected, we propose to evaluate its severity by using data replay analysis to determine if this fault could have caused errors in the past. This approach significantly decreases the number of necessary sensor repairs and maintenance dispatches.

1 Introduction

It is extremely difficult to guarantee continuous and reliable activity detection in wireless sensor network (WSN) applications. A major challenge with activity detection is that there are numerous sensors which are likely to move or exhibit failures due to hardware degradation, inaccurate readings, and environmental changes. These problems will be even more important in the future, as sensors built into objects become more ubiquitous. As the numbers of sensors increase, even a large mean time to failure (MTTF) per sensor can translate to small MTTF for large groups. In addition, because of the inherent redundancy of these systems, not all failures will be equally severe.

In this paper, we propose an approach to help maintain sufficient activity detection accuracy in the presence of sensor failures. The key insight behind our approach is that only those sensor failures that affect the high-level application-level behavior of the system need to be repaired. The main difference between our approach and others, such as [1–3], is that it uses a top-down failure detection mechanism. Therefore, instead of detecting and reporting low-level hardware faults, our approach relies on application semantics to determine the severity of application-level faults. In addition, bottom-up approaches like those described above that rely on analysis of the raw data, are not easily applicable to binary, event-triggered sensors, such as motion sensors or reed switches.

* This work was supported in part by NSF Grants CNS-0834299, IIS-0931972, and CNS-1017363.

The basic approach is to constantly monitor the accuracy of activity detection at run time. If a failure occurs, we assess the severity of that failure by determining its impact on the activity detection accuracy. If the detection accuracy remains above a predefined threshold, a repair is not necessary. If the detection accuracy deteriorates significantly, the system needs to react in one of two ways: 1) It can update the existing classifier so that it no longer depends on the failed sensor nodes, 2) If the accuracy still remains below the specified threshold, maintenance should be performed.

We have performed preliminary evaluation using a publicly available dataset [4]. Our results show that our approach significantly decreases the number of node repairs and maintenance dispatches. We also compare the activity detection accuracy of different classifiers, such as naive Bayes, hidden Markov model (HMM), and hidden semi-Markov model (HSMM), under node failures.

The expected main contributions of our work are that we would be able to i) determine which nodes have failed, ii) maintain high ADL detection accuracy even under node failures, and iii) substantially reduce the number of repairs and maintenance dispatches. In this paper, we only discuss the last two contributions. To the best of our knowledge, this is the first work to study the accuracy of activity detection algorithms in the presence of sensor node failures.

2 Approach

We use the data set presented in [4]. This data set consists of sensor events and activities for 28 days of a 26-year-old man living alone in a three-room apartment where 14 switch sensors are installed. There are a total of 2120 sensor events and 245 activity instances. The subject himself annotated seven different activities that include *Leave House*, *Toileting*, *Showering*, *Sleeping*, *Prepare breakfast*, *Prepare dinner*, and *Taking a drink*. We use this annotation as the ground truth. Each day is divided into fixed-length timeslots and the activity performed in each timeslot is classified based on the sensor firings within that particular timeslot.

To calculate the accuracy of a classifier, we divide the entire data set into training and testing sets, and train the classifier with the training set. We use the trained classifier to label the activity during each timeslot of the testing data. Then we calculate the $M \times M$ confusion matrix C where M is the number of different activities and C_{ij} represents the number of timeslots that belong to activity i (as specified in the ground truth) but are classified as activity j. From this table, we calculate number of True Positives (TP), False Positives (FP), and False Negatives (FN) for each activity. We define *Precision* = TP/(TP+FP) and *Recall* = TP/(TP+FN). Finally, similarly to van Kasteren et al [4], we calculate the *F-Score* using Equation (1). We use the *F-score* of each activity to measure the accuracy of the classifier in detecting that particular activity.

$$\text{F-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{1}$$

After training a classifier, while using it for detecting an activity, we may detect that one or more sensors are broken. In this work, we assume that we can detect each sensor failure and our goal is to calculate the minimum number of sensor repairs and maintenance dispatches so that the detection accuracy remains acceptable. If we continue to use the same classifier when one or more sensors fail, then the accuracy of the classifier may deteriorate, because we have trained the classifier with data that includes events from the failed sensors. Our approach is to dynamically update the classifier by re-training it with an updated training set after detecting that n sensors have failed. We update the training set by removing events from all of the n sensors that are currently broken. Then we use this re-trained classifier for detecting the activity. When calculating accuracy for n sensor failures, we construct training and testing sets for each of the $\binom{N}{n}$ (N is the total number of sensors) combinations of n sensors being removed. Then we calculate the *F-score* for each testing set and use it to evaluate the accuracy of the classifier in detecting this activity when n sensors have failed.

If the accuracy for n sensor failures is within $x\%$ of the actual accuracy, we consider this event of n sensor failures as insignificant to the classifier. Therefore, we do not need to repair any sensor. Otherwise, at least one sensor needs to be repaired. For such cases, for each of the $\binom{N}{n}$ cases described above, we calculate the minimum number of sensor repairs r ($1 \leq r \leq n$) so that after repairing r of the n sensors, i.e. after including sensor events from these r sensors, the accuracy rises within $x\%$ of the actual accuracy for this activity. For lack of space, we do not describe the algorithm we use to efficiently calculate the number of minimum repairs. We calculate r for each of the $\binom{N}{n}$ combinations.

3 Experimental Results

Figure 1 shows a preliminary evaluation of our approach. We monitor the detection accuracy degradation of the *Prepare breakfast* activity when node failures occur and determine the minimal number of necessary repairs. For this experiment, we assume that nodes fail in a detectable way. We chose the *Prepare breakfast* activity because of its complexity, as its detection is influenced by seven sensors: microwave, cups cupboard, fridge, freezer, pans cupboard, and groceries cupboard. We consider node repair to be necessary if the detection accuracy drops below x% of the actual accuracy. In this experiment x = 30%.

When the baseline repair technique is used, a repair is necessary every time a sensor node fails. Figure 1 shows the number of repairs when our approach is applied. We can see that when we use a naive Bayes classifier, if one node fails, we need to perform a repair in only 43% of the cases. This means that the failure of only 3 of the 7 nodes affects the application behavior and it therefore considered severe. The number of necessary repairs gradually increases when more nodes fail, but always remains smaller that the number of repairs required by the baseline approach. For example, when all 7 sensors fail, our approach determines that only 3 nodes should be repaired instead of 7. Figure 1 can also be used to reason about the decrease in maintenance dispatches. A maintenance dispatch

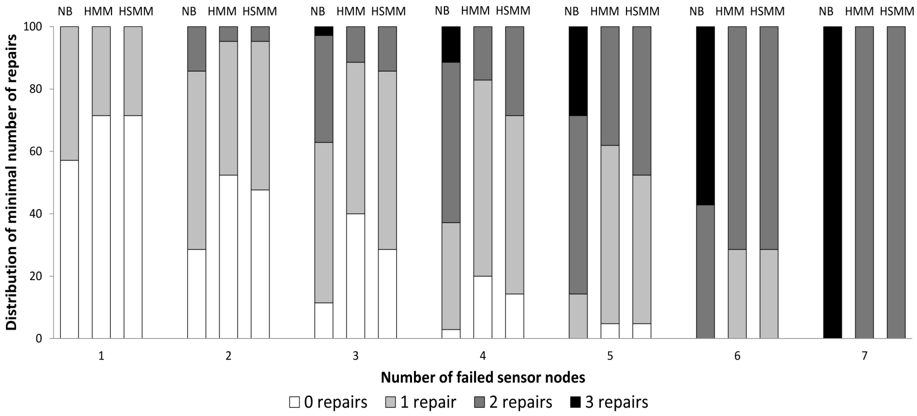

Fig. 1. Distribution of minimal number of repairs for activity *Prepare breakfast*. We use three classifiers - naive Bayes (NB), hidden Markov model (HMM), and hidden semi-Markov model (HSMM). Compared to a baseline, where a repair is necessary every time a node failure occurs, our approach shows decrease in the number of repairs and maintenance dispatches.

can be avoided when our approach determines that no repairs are needed. Therefore, with a Naive Bayes classifier, the number of dispatches is decreased by 57% when one sensor has failed, by 29% with two failed sensors, and so on.

Another observation we make is that there is a significant difference in the number of repairs when different classifiers are used. The detection accuracy degradation is worse with a naive Bayes classifier. HMM and HSMM show very similar behavior and tend to perform better than Naive Bayes. The difference in the performance of the classifiers changes with changing the threshold x. When x is small, the performance gap between naive Bayes and the Markov models is small, and they require comparable number of node repairs. With the increase of x, the performance gap also increases.

References

1. S. Ganeriwal, L. K. Balzano, and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, June 2008.
2. N. Ramanathan, T. Schoellhammer, E. Kohler, K. Whitehouse, T. Harmon, and D. Estrin. Suelo: human-assisted sensing for exploratory soil monitoring studies. In *SenSys*, 2009.
3. S. Rost and H. Balakrishnan. Memento: A health monitoring system for wireless sensor networks. In *SECON*, 2006.
4. T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *UbiComp*, 2008.