2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing

A Nonintrusive and Single-Point Infrastructure-Mediated Sensing Approach for Water-Use Activity Recognition

Lisha Hu, Yiqiang Chen, Shuangquan Wang
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
{ hulisha, yqchen, wangshuangquan}@ict.ac.cn

Liping Jia
School of Software
Beihang University
Beijing, China
lpjia86@sina.com

***Abstract*—Recent years, a variety of infrastructure-mediated sensing methods have been proposed to recognize activities of daily living (ADLs). However, due to the inconvenience such as high-cost, difficult-to-install, intrusive and applicable to the house with specific architectures alone, existing water-use activity recognition methods cannot be widely used into people's houses. In this paper, a single-point infrastructure-mediated sensing technique is proposed for water-use activity recognition. A single 3-axis accelerometer sensor is attached to the surface of the main water pipe in the house to detect and collect the vibration signals of the main water pipe. These signal data are then processed through six modules in the proposed activity recognition system. Four classes of water-use activities (Bathing, Flushing toilet, Cooking and Washing) are classified by the system and experimental results show that our system can recognize about 70.37% water-use activities.**

Keywords-water-use activity recognition; machine learning; infrastructure-mediated sensing; nonintrusive

I. INTRODUCTION

Monitoring daily activities of elders is still a challenge in the ubiquitous computing research domain. Existing approaches are typically expensive, intrusive or difficult to install. Among the techniques proposed by now, infrastructure-mediated sensing has been recognized as a low-cost and nonintrusive activity recognition technique. The main idea behind infrastructure-mediated sensing technique is: A person's daily activities can be recognized by monitoring the infrastructures (e.g. water, electric, heat, ventilating, air conditioning [1-7]) in the house.

A number of infrastructure-mediated sensing approaches have been proposed. Patel et al present a system in [4] to detect the presence of batteryless tags at home through the power lines. The batteryless tags can be excited by the energy rich transients occurred by turning on a light switch or television in response to the electrical transients. A single plug-in module monitors the power line for the presence of the tags when these tags are excited. However, their detection range is of short range (30-50 cm) and the read distance along the power line is also fairly short (3-4 meters). He also proposes a human movement detection approach by detecting and recording the pressure variation of sensors mounted on the air filter [3]. He detects the pressure variation from pressure sensors mounted on the air filter and classifies movement events occurring in the house, such as walking through a doorway or opening a door. To achieve this goal, the air filter of HVAC has to be instrumented with five pressure sensors to detect airflow in all directions. Lymberopoulos et al present a methodology of counting the number of times elder go to the washroom one night to monitor elder's health condition [8]. Dozens of camera and motion sensors are instrumented in the whole house to monitor an old person and collect the data, which is a bit more intrusive and might upset the owner.

Several infrastructure-mediated sensing approaches for water-use activity recognition have been proposed recently, since we can infer the user's daily activities based on his/her water usage patterns. A water-use activity recognition technique is proposed by Fogarty in [9] by deploying four microphones on the surface of water pipes near the inlets and outlets. Although this technique is easy-to-install, the signals collected are easy to interferential by sounds of ambient environment, such as the central air conditioning unit. Froehlich et al propose HydroSense, another infrastructure-mediated single-point sensing technique, in [10]. HydroSense can support the identifications of various activities at individual water fixtures as well as the estimation of the amount of water being used at each fixture by analyzing continuously the pressure within the water infrastructures in the house. Therefore, the disadvantage of HydroSense is: the pressure sensor can only be installed and utilized in the house with hose bib. Most of the houses have to be re-constructed in order to work in with the HydroSense. Thomaz et al propose a technique of combining single-point, infrastructure-mediated sensing with a vector space model learning approach for high-level activity recognition in [11]. Their work has been considered to be the first one of employing the method for inferring high-level water-use activities. The same as above, the infrastructure of the house has also to be remodeled in order to work in with the installation of the pressure sensors.

In this paper we propose a single-point infrastructure-mediated sensing system for water-use activity recognition by having a single 3-axis accelerometer clinging to the surface of the main water pipe in the house. Obviously, the device is easy to install or take down. Unlike the pressure sensors in [7], we do not need any re-construction projects into the house. Besides, compared with the microphone sensor utilized in [6], our accelerometer is low-price and less

978-0-7695-5088-6/13 $31.00 © 2013 IEEE
DOI 10.1109/HPCC.and.EUC.2013.304

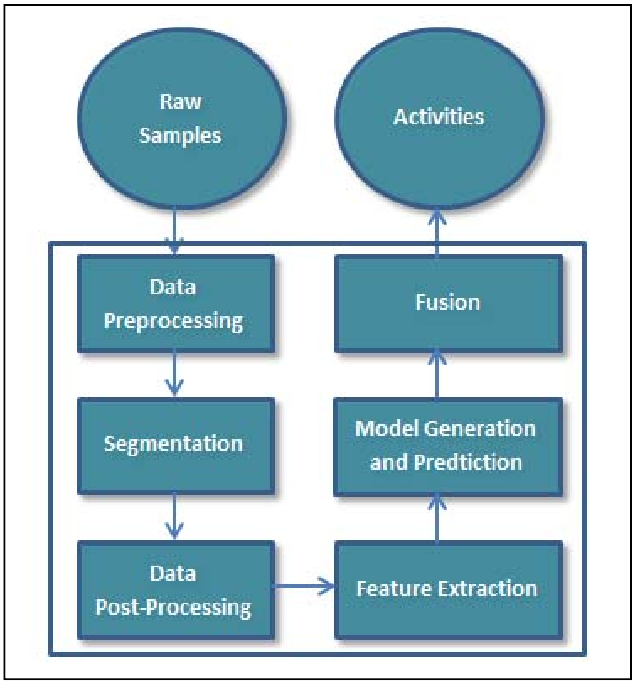

Fig.1. Block diagram of water-use activity recognition system.

sensitive to the ambient environment. Experimental results show that our system can recognize correctly all the water-use activities in the house with acceptable accuracy (about 70.37%) by utilizing such an easy-to-install and absolutely nonintrusive device.

II. WATER-USE ACTIVITY RECOGNITION

Figure 1 shows the block diagram of water-use activity recognition system. There are six modules in the system. The input of system is a set of raw **time series samples** (a series of samples at successive times). Time series samples are preprocessed first to filter noises and then segmented into several **smooth segments** (time series smooth samples) and **rugged segments** (time series rugged samples). The segmentation module is based on the fact that: If any water-use activity is in process, the samples collected from the device are usually rugged. The reason is that vibrations are generated to the pipe when the water flows in it. Each segment is post-processed to own a proper length. We care only about those rugged segments. For each rugged segment, several instances can be generated by extracting features from the samples in this segment. Each instance' affiliation with its corresponding segment is also reserved. All instances are split randomly in two sets. One set is utilized for model generation and the other for water-use activity prediction. Distinguished prediction results of instances in the same segment are fused into a single prediction result by law of 'The minority is subordinate to the majority'. At last, the output of the system is a set of time series water-use activities. Each module is further explained as follows.

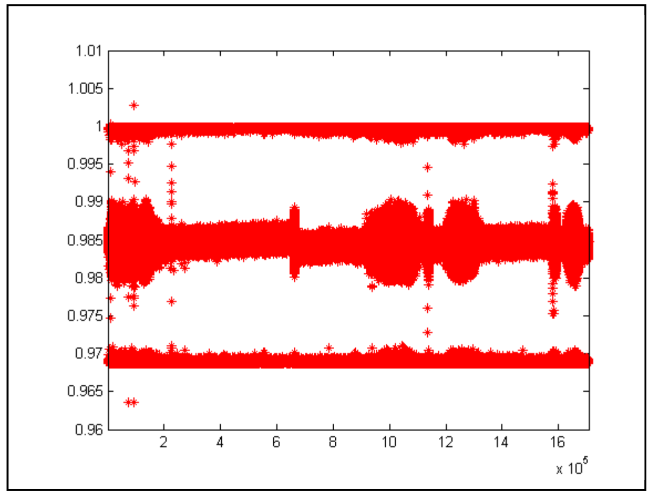

Fig.2. Raw samples.

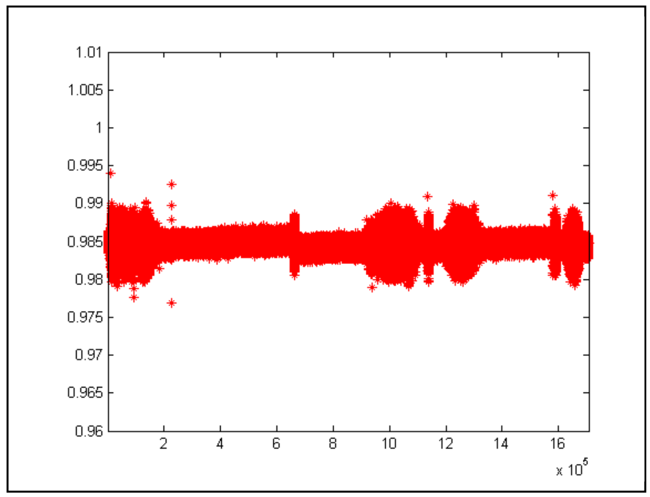

Fig.3. Raw samples.

A. Data Preprocessing

Normally, there exist some noises in the raw time series samples which should be filtered out. The median filter technique is employed in this data preprocessing module and the filter window is set to 3. Figure 2-3 show the comparison of time series samples before and after the data preprocessing module.

B. Segmentation

The segmentation module is aimed at segmenting both the rugged segments and smooth segments from time series samples.

First, **sample windows** (time series samples of size SIZE_WINDOW1) are generated on the set of time series samples according to the sliding window mechanism (window size: SIZE_WINDOW1, step length: SIZE_STEP1); Second, we annotate each sample window to be rugged or smooth based on whether its standard deviation is no less than a threshold (SIZE_THRE1) or not. At last, a rugged (or

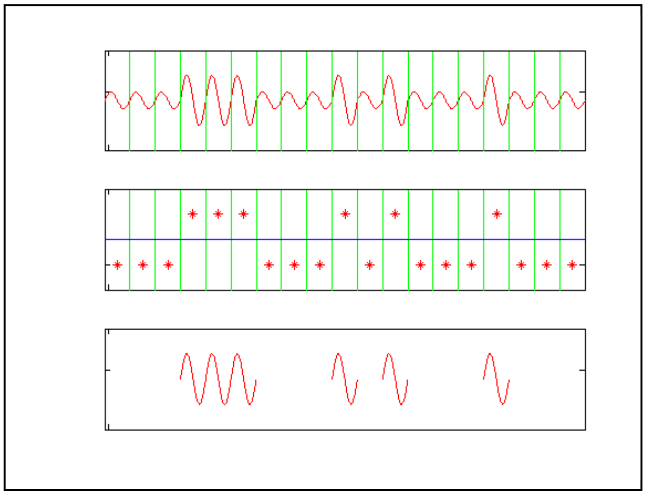

Fig.4. Segmentation procedure

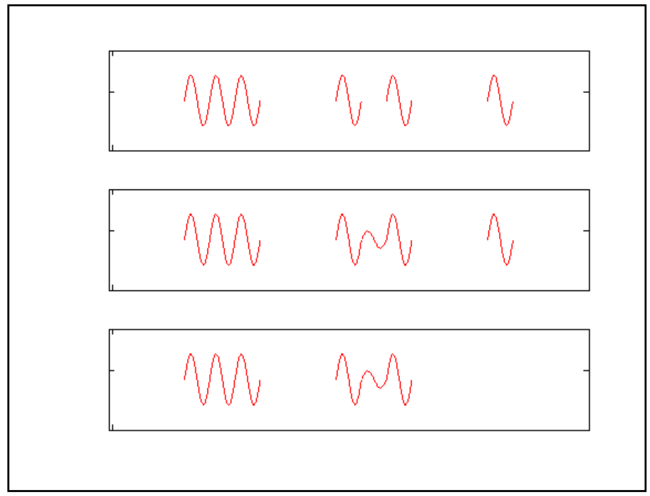

Fig.5. Data post-processing procedure

smooth) segment is defined as a time series rugged (or smooth) windows.

Several rugged and smooth segments can be generated after the segmentation module. Each segment's segmentation result is different with the result(s) of its neighbor(s).

Fig 4 illustrates the segmentation procedure. We show a string of time series samples (marked in red) in the above sub-graph of Fig 4. Let the parameters SIZE_WINDOW1 and SIZE_STEP1 be fixed, several sample windows can be generated and shown in the above sub-graph, in which each two neighboring sample windows are separated by a vertical green line. For each of the sample windows, we compute the standard deviation corresponding to the samples in it, and mark the value of standard deviation as a red star in the middle sub-graph. The blue horizontal line in the middle sub-graph represents the threshold (SIZE_THRE1). Six sample windows, whose corresponding standard deviation values are above the blue horizontal line, represent the rugged windows and the segments and five smooth segments are generated. Only the rugged segments are shown in the sub-graph below.

C. Data Post-Processing

The data post-processing module is to make all the rugged segments generated in the last module more complete and precise. The goal of this module is to deal with two circumstances below:

- Time series samples of a water-use activity are segmented into not one, but two or more rugged segments (these rugged segments are separated from each other by small-sized smooth segments) in the last module.
- Some rugged segment is too 'short' (samples contained in the segment are not enough) to contain any kinds of water-use activities.

Two thresholds are employed in this module to deal with these two circumstances.

First stage post-process procedure: In the first circumstance, any smooth segment (in between two rugged segments), whose corresponding samples are no more than a threshold (SIZE_THRE2), is re-annotated to be rugged segments. After that, all the neighboring rugged segments make up a long rugged segment.

Second stage post-process procedure: In the second one, any rugged segment (in between two smooth segments), whose corresponding samples are no more than another threshold (SIZE_THRE3), is re-annotated to be smooth segments. After that, all the neighboring smooth segments make up a long smooth segment.

We give an example to illustrate the post-processing procedure (Fig 5). The above sub-graph in Fig 5 describe the four rugged segments after the segmentation module, which can also be seen in the below sub-graph of Fig 4. Let the parameters SIZE_THRE2 and SIZE_THRE3 be fixed. According to the first stage post-process procedure, the invisible smooth segment in between the second and third rugged segments (the space in between them) in the above sub-graph in Fig 5, is re-annotated to be rugged segment and merge into a long rugged segment (the middle sub-graph in Fig 5); Afterwards, the last rugged segment in the middle sub-graph is re-annotated to be the smooth segment and became invisible in the behind sub-graph in Fig 5 after the second stage post-process procedure. The number of segments is changed into two at the end of data post-processing procedure.

Since most of the water-use activities are embedded in the rugged segments, we care only about those rugged segments generated and all the smooth segments are discarded. In the following modules, all the samples utilized are from the rugged segments.

D. Feature Extraction

Instances are generated by utilizing the sliding window mechanism (window size: SIZE_WINDOW2 step size: SIZE_STEP2) again on each rugged segment. Before that, we observe the true labels (water-use activities) of all the samples in each rugged segment, in which there might exists

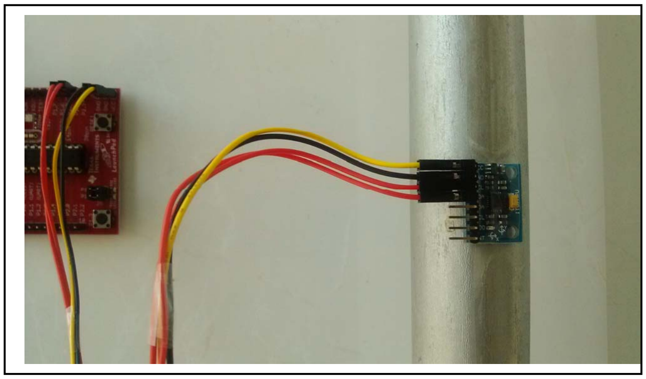

Fig.6. Devices used for data collection.

a single or more sub-segments of water-use activities. The feature extraction module is executed on each sub-segment. Eight features (0.25-quantile, 0.5-quantile, 0.75-quantile, mean value, standard deviation, quadratic sum, zero-crossing, spectral peak) are extracted from a window of sample values in each axis (x, y or z axis in the accelerometer device). In all, there are 24 features for each instance. Besides, the label of each instance is the water-use activity going on during its corresponding sample collection process.

A set of instances are generated after the feature extraction module. Each instance' affiliation with its corresponding segment is reserved for the next modules.

E. Model Generation and Prediction

All the instances are split into two sets (the training set and the testing set) with approximately the same size. Instances in the same segment are assured to be put into the same set, since we do not want any water-use activity to be apart.

Support Vector Machine (SVM) is employed for model generation. We utilize the Gaussion kernel as our kernel function. Two parameters---the kernel parameter and the penalty parameter---need to be set before we start the learning process. In the end, a classifier is learnt on the training set.

The classifier is then employed to predict the labels of instances in the testing set (testing instances). These prediction results are recognized as SVM's prediction labels for the testing instances.

F. Prediction Results Fusion

As mentioned above, every rugged segment can be recognized as once water-use activity in our activity recognition system. However, SVM's prediction labels for the testing instances of the same rugged segment might be different and have more than one kind. Therefore, it is necessary to fuse these instances' various prediction labels into the same prediction label.

The prediction results fusion module is done by law of 'The minority is subordinate to the majority'. Specifically, for each water-use activity, we count the number of testing instances in a rugged segment whose prediction labels are this class. The dominant water-use activity in the segment

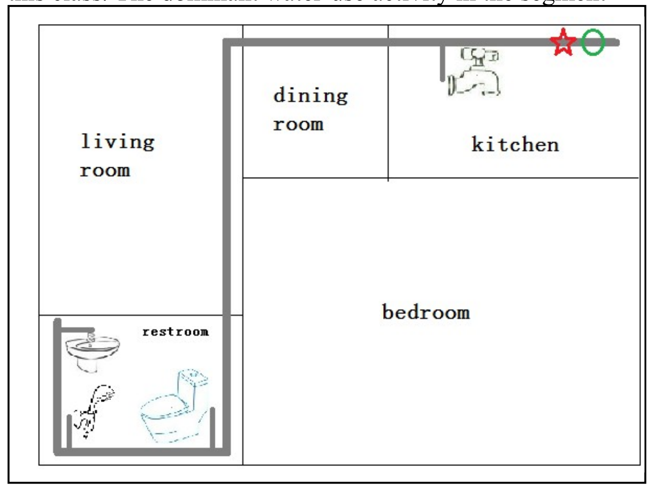

Fig.7. The structure of the water pipe

corresponds to the most testing instances. In the end, the prediction labels of all instances in the segment are replaced by the dominant water-use activity. The prediction results of the rugged segment are fused finally.

III. EXPERIMENTATION AND EVALUATION

All the experiments are executed in MATLAB 7.11.0 environment running in Core 2 Quad, 3.20-GHZ with 3-GB RAM. LIBSVM toolbox [12] is employed for SVM training and prediction.

A. Data Collection

Fig 6 shows the accelerometer (MPU6050) and how it is attached to the main water pipe. The accelerometer has a range of ±2g and is connected to a MCU (MSP430G2553) which can read the recorded data and communicate with a laptop. The data is transferred by a serial port with 9600 baud rate. The sampling rate of the device here is set to be 85. Each sample is saved as a vector of 3-axis accelerometer readings as well as the time information.

All the data are collected in an apartment of about 50 square meters. The structure of the water pipe in the apartment can be seen in Fig 7. The thick and thin grey lines represent the main water pipe and branches of the main pipe. The green circle and red star in Fig 7 are water meter and the accelerometer respectively. Since water flows into the family through the water meter first, we attach the accelerometer next to it to capture the vibration signals of the pipe. It is obvious that there are four kinds of water-use activities in this apartment: Bathing, Flushing toilet, Cooking and Washing.

Samples of four water-use activities (bath, flush toilet, cook, wash) and none water-use activity are collected by the device. When collecting all these samples, we also record their true labels manually with the help of a specific App running on the android mobile phone. There are four buttons corresponding to four water-use activities in the screen of the phone when the App is running. When a certain water-use activity begins or ends, the button corresponding to the activity is pressed manually and the time is recorded in the

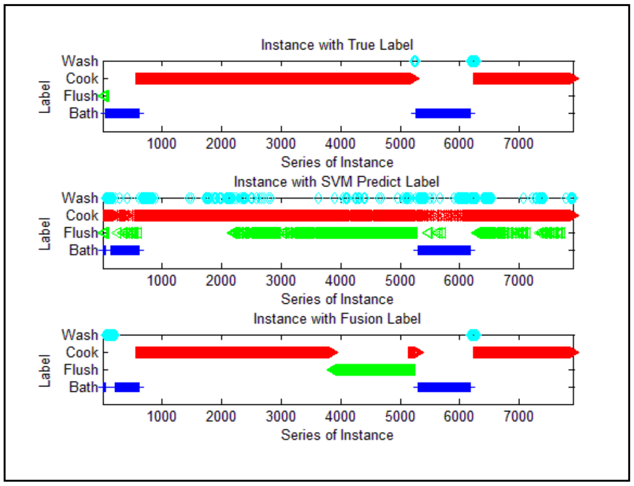

Fig.8. Water-use activity recognition results.

phone for this specific water-use incident. Samples' labels can be obtained by comparison of their time information with the records.

The data collection procedure goes on for 7 days, usually from 6 pm to 11 pm. In all, more than 31.31 hours data (about 9 million samples) are collected.

B. *Experimental Results*

All the parameters are set as follows. The number of samples in a sample window (SIZE_WINDOW1) is 40; the step length (SIZE_STEP1) of the sliding window is 20.

The threshold (SIZE_THRE1) of the standard deviation is calculated based on samples of 1.5 hour none water-use activity collected separately in the data collection procedure. First, sample windows are generated for these none water-use activity data according to the segmentation procedure above. Second, the standard deviation of samples in each sample window is calculated. At last, the threshold (SIZE_THRE1) is set to be the value which is larger than ninety-nine percent of these standard deviation values.

Two thresholds (SIZE_THRE2, SIZE_THRE3) in the data post-processing procedure are 100 and 1000 respectively. In other words, any smooth (or rugged) segment in between two rugged (or smooth) segments, which consists of no more than 100 (or 1000) samples, will be re-annotated to be rugged (or smooth) and combined with its two neighboring rugged (or smooth) segments. In all, there are 54 segments (26 times Bathing, 2 times Flushing toilet, 22 times Cooking and 4 times Washing) generated, which contain the samples from all the water-use activities.

The window size (SIZE_WINDOW2) and step length (SIZE_STEP2) of the sliding window for feature extraction are 100. After feature extraction, we generate 22104 instances in the end. That is, there are 2979, 135, 18492 and 498 instances for Bathing, Flushing toilet, Cooking and Washing respectively.

It is obvious to see that instances from different classes are extremely unbalanced from several hundreds to more than ten thousands. Therefore, we employ the weight SVM

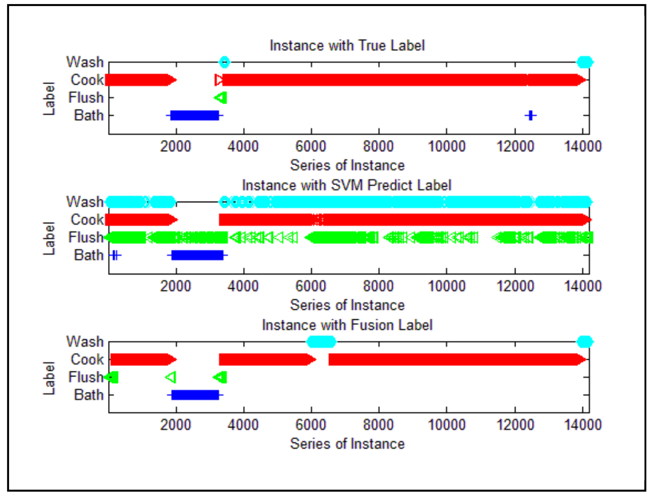

Fig.9. Water-use activity recognition results.

in the LIBSVM toolbox to learn the classifier. The weight for each class is set to be the ratio of all instance numbers to instances in this class, which is the commonly used method for unbalanced classification problems.

All the instances are split into two sets. Instances in the same segment are assured to be in the same set. One set is used for training and the other one is for testing. Therefore, we can attain the classification results of all the instances (Fig 8-9).

Fig 8 shows the comparison results between SVM prediction results and fusion results on the two sets. Numbers in horizontal axis represent orders of the instances. Symbols in vertical axis represent 4-class water-use activities ("Bathing", "Flushing toilet", "Cooking" and "Washing"). The above sub-graph in Fig 8 (or Fig 9) depicts true labels of the instances. The prediction results of SVM on the testing instances are shown in the middle sub-graph. At last, the final fusion results are shown in the sub-graph below.

According to the prediction results of SVM in Fig 8-9, the Precision and Recall for the 4-class can be seen in Table 1-2.

From the middle sub-graph in Fig 8 we can see that, the prediction results of SVM classifier are chaotic for the instances. Many instances are misclassified into the Flushing toilet and Washing classes. To be more specific, the Precisions of SVM classifier on these two classes are only 1.54% and 14.57% (see Table 1).

Such phenomenon happened owning to the fact that: there are only a few instances in these two classes, leading to their weights (175 for Flushing toilet and 39 for Washing) much larger than the weights of others (9 for Bathing and 1 for Cooking) in learning the classifier. As a result, the classifier became to bias the small-sized classes. Unfortunately, the Recalls of SVM classifier on the two classes are not high either (40.74% and 47.18%). The same phenomenon can also be found in the middle sub-graph of Fig 9.

TABLE I. WATER-USE ACTIVITY RECOGNITION RESULTS ON PRECISION AND RECALL

	Precision		Recall	
	SVM	*Fusion*	*SVM*	*Fusion*
Wash	14.57%	41.09%	47.18%	74.65%
Cook	95.51%	98.46%	73.95%	78.66%
Flush	1.54%	0%	40.74%	0%
Bath	97.33%	95.97%	78.92%	86.96%

After the fusion module, the prediction results became much more regular in the sub-graphs of Fig 8-9. Moreover, both Precisions and Recalls of the fusion results increase to a great extent for the three classes (Bathing, Cooking and Washing) compared with the results of SVM classifier (Table 1-2). For instance, the Precision and Recall increase by 182.02% and 58.22% in the Washing class, which shows that the fusion module can correct and improve the prediction results greatly.

Now we explain the prediction results in the Flushing toilet class: The Precision and Recall decrease from 1.54% and 40.74% to 0% after the fusion module in the first set, but increase from 3.72% and 56.79% to 22.19% and 100% in the second set. Since the instances in the Flushing class are all classified correctly in both sets (Fig 8-9), these two different results are mainly based on the two Recall values (40.74%, 56.79%) of SVM classifier. There is only one rugged segment for the Flushing toilet class in each set. The Recall (56.79%) in the second set represents that: About 56.79% instances in the segment are classified correctly by SVM classifier. In other words, the Flushing Toilet is the dominant label in this segment. Therefore, the Recall increases to 100% percent after the fusion model.

However, the Recall (40.74%) in the first set is less than fifty percent, which means more than a half instances in the segment are misclassified. To be more specific, almost all the misclassified instances in this segment are classified into the Bathing class, leading to it being the dominate label in the segment. Therefore, instances in the segment are re-annotated to be the Washing label in the fusion module, and the Recall became 0% in the end.

In conclusion, our water-use activity recognition system works well on the basis of the performance of the classifier learnt in the model generation module. Overall, our system can recognize correctly about 70.37% water-use activities.

IV. CONCLUSION

A nonintrusive and single-point infrastructure-mediated sensing approach is proposed in this paper to recognize 4-class water-use activities in daily life. Data is collected unobtrusively by a single low-cost 3-axis accelerometer attached to the surface of the main water pipe in the house, making the installation process much more convenient. Experimental results show that our system can recognize the water-use activities in the house with acceptable prediction ability.

TABLE II. WATER-USE ACTIVITY RECOGNITION RESULTS ON PRECISION AND RECALL

	Precision		Recall	
	SVM	*Fusion*	*SVM*	*Fusion*
Wash	4.98%	27.83%	52.53%	71.07%
Cook	97.30%	98.39%	62.12%	92.73%
Flush	3.72%	22.19%	56.79%	100%
Bath	95.27%	100%	87.26%	91.13%

ACKNOWLEDGMENT

This work was supported in part by Natural Science Foun dation of China under Grant No.61070110 and No.90820303 , Beijing Natural Science Foundation under Grant No. 411205
and open project of Beijing Key Laboratory of Mobile Comp uting and Pervasive Device.

REFERENCES

[1] R.P.Evans, J.D. Blotter and A.G. Stephens, "Flow rate measurements using flow-induced pipe vibration," Fluids Engineering, vol. 126, pp. 280-285, May 2004.

[2] Y. Kim, T. Schmid and Z.M. Charbiwala. "NAWMS: nonintrusive autonomous water monitoring system," in Proceedings of the 6th ACM Conference on Embeded Network Sensor Systems, 2008, pp. 309-322.

[3] S.N. Patel, M.S. Reynolds and G.D. Abowd, "Detecting human movement by differential air pressure sensing in HVAC system ductwork: an exploration in infrastructure mediated sensing," in Proceedings of the 6th International Conference on Pervasive Computing, 2008, pp. 1-18.

[4] S.N. Patel, E.P. Stuntebeck and T. Robertson, "PL-Tags: detecting batteryless tags through the power lines in a building," in Proceedings of the 7th International Conference on Pervasive Computing, 2009, pp. 256-273.

[5] J. Froehlich et al, "A longitudinal study of pressure sensing to infer real-world water usage events in the home," in Proceedings of the Ninth International Conference on Pervasive Computing, 2011, pp. 50-69.

[6] S. Gupta, M.S. Reynolds and S.N. Patel, "ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home," in Proceedings of the 12th International Conference on Ubiquitous Computing, 2010, pp. 139-148.

[7] S.N. Patel, T. Robertson, J.A. Kientz, M.S. Reynolds and G.D. Abowd, "At the flick of a switch: detecting and classifying unique electrical events on the residential power line," in Proceedings of the 9th International Conference on Ubiquitous Computing, 2007, pp. 271-288.

[8] D. Lymberopoulos, A. Bamis and A. Savvides, "A methodology for extracting temporal properties from sensor network data streams," in Proceedings of the 7th International Conference on Mobile Systems, Applications and Services, 2009, pp. 193-206.

[9] F. James, and A.S.E.H. Carolyn, "Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity

recognition," in Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, 2006, pp. 91-100.

[10] J. Froehlich et al, "HydroSense: Infrastructure-mediated single-point sensing of whole-home water activity," in Proceedings of the 11th International Conference on Ubiquitous Computing, 2009, pp. 235-244.

[11] E. Thomaz et al, "Recognizing water-based activities in the home through infrastructure-mediated sensing," in Proceedings of the 14th International Conference on Ubiquitous Computing, 2012, pp. 85-94.

[12] C.C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Techology, 2011, http://www.csie.ntu.edu.tw/~cjlin/libsvm.