Personal models for eHealth - improving user-dependent human activity recognition models using noise injection

Pekka Siirtola, Heli Koskimäki and Juha Röning
Biomimetics and Intelligent Systems Group
P.O. BOX 4500, FI-90014, University of Oulu, Oulu, Finland
pekka.siirtola@ee.oulu.fi, heli.koskimaki@ee.oulu.fi, juha.roning@ee.oulu.fi

***Abstract*—In this paper, a noise injection method to improve personal recognition models is presented. The idea of the method is to build more general recognition models for eHealth using a small original data set and by expanding the area covered by training data using noise injection. This way, it is possible to train models that are less vulnerable to changing conditions, and thus more accurate, but still the data gathering phase can be non-burdensome. The proposed method was tested using two classifiers (linear discriminant analysis and quadratic discriminant analysis) and three human activity recognition data sets collected using inertial sensors of a smartphone. Two of these data sets are open data sets. The results show that noise injection improves the true positive recognition rates. With first data set the improvement varies from 1.3 to 2.0 percentage units, with second from 1.4 to 4.5 percentage units, and with third the highest improvement was 2.5 percentage units. Moreover, the results show that the method improves precision and reduces false positive rates. In addition, experiments were made using different training set sizes to show that the improvement in true positive rate is bigger if the original training data set is small. In this study, the method is experimented using human activity data sets but it is not limited to this application area and can be used with any time series data.**

***Index Terms*—Activity recognition; eHealth; noise injection; smartphone; inertial sensors; accelerometer; personal models;**

I. Problem statement and related work

In eHealth it is often important to recognize what a person is doing or a condition of a person. However, people are different, and therefore, a user-independent recognition model that provides accurate recognition results for one person does not necessary provide as high results for other person. There are many possible reasons for this phenomena. For instance, person's can be different: physical characteristics, health state or gender can have an effect to the recognition results. On the other hand, external factor such as weather, terrain and location can cause problems to recognition models. In addition, in the real-life many other unseen contingencies can happen and the training data set used to train the recognition models cannot include all of these. This means that model that seem to work really well when tested with testing data do not work as well when it is used online, in real-time, real-life applications [1].

In the case of human activity recognition, often the recognition is done using user-independent models and good recognition rates have been achieved (for instance [2], [3], [4], [5]). However, it has been shown that user-independent models do not work accurately for instance if trained with healthy study subjects and tested with subjects who have difficulties to move [6]. Personal models could solve this problem. In fact, it has been shown that user-dependent models are more accurate than user-independent ones [7]. However, in [8] it was shown that this is not always the case. In the study, it was shown that often user-dependent models fail. The reason for this was that models were not general enough. It would be possible to improve the generalization of the model by collecting more data, but this is especially burdensome when models are user-dependent. This because it would require an extensive, separate data collection session for each user. If the aim is to build a commercial application for the masses, this is far from ideal situation.

If the recognition model is not general enough, it cannot react to the changing conditions. This means that when conditions slightly change, a feature vector extracted from unseen data corresponds a point in the feature space that does not have any members from the pool of events defined by the training data set, and the accuracy of the model decreases. However, recognition models can be made more general using regularization methods [9]. In this study regularization is obtained using noise injection, which has been shown to improve recognition rates especially if the training data set is small [10]. Noise injection has been used in many different application areas, for instance in computer security to test the reliability of automatic worm signature generators [11], to improve biometric models based on fingerprint, speech, and hand geometry data [12], to model carbonate reservoirs [13], in image recognition [14], and to recognize cancer patients based on magnetic resonance spectra data from biofluits [15]. In most of these studies, the noise injection is based on the study by [16] where noise injection was used with K-nearest neighbors algorithm in multilayer perceptron training. However, what is noticeable is that noise injection has been used to with wearable sensor data only to balance data ([17]) but it has not been studied how it can be used to build more general models from a small original data set.

In our study, the aim is to build general personal recognition models for eHealth applications from a small training data set. Noise injection is used to expand the area covered by training data, and in this way, make the models trained using it more general and less vulnerable to changing conditions. This is shown to improve the recognition rates, especially if the amount of the training data is small. The proposed idea is tested using three data sets, two of these are open data sets. In addition, experiments are done using two classifiers but the method can be used with any classifiers.

The paper is organized as follows: Section II presents the used data sets and the normal human activity recognition protocol. Section III introduces the proposed method to improve personal classification models using noise injection. The Section IV describes the experimental protocol and shows the results. Finally, the results are discussed in Section V and conclusions are in Section VI.

II. Data sets and methods

A. Experimental data sets

In this study, the experiments are done using three human activity data sets. Common with these three data sets is that they are all collected from smartphone sensors.

Open data set presented in [18] contains data from 10 study subjects and from seven physical activities (walking, sitting, standing, jogging, biking, walking upstairs and walking downstairs). The data were collected from five body locations but in this study only the data collected trouser's pocket were used. The data was collected from an accelerometer, a gyroscope, and a magnetometer using sampling rate of 50Hz. From now on this data set is referred as SAD (Sensors Activity Data set).

Another open data set used in this study is presented in [19]. This data set contains data from 30 study subjects and from six activities (standing, sitting, lying, walking, walking downstairs and walking upstairs). The data were collected using a smartphone that was located at the study subjects waist. The data was collected from an accelerometer, and a gyroscope using sampling rate of 50Hz. From now on this data set is referred as SBHAR (Smartphone-Based Human Activity Recognition data set).

The third used data set is not open and it is originally used in [20]. It is collected from seven study subjects and from five activities (walking, running, cycling, driving a car and idling). The data were collected using a smartphone that was located at the study subjects trouser's pocket. The data was collected from an accelerometer using sampling rate of 40Hz. From now on this data set is referred as DHA (Daily Human Activity data set).

There are big differences in the sizes of these data sets. In SAD there is on average 20 minutes per test subject, in SBHAR 11 minutes, and in DHA 27 minutes.

B. Human activity recognition process

Activity recognition process is divided into three main phases (data collection, training and activity recognition), and each of these can be divided into subphases, see Figure 1. In this section, it is explained what methods this study uses in different stages.

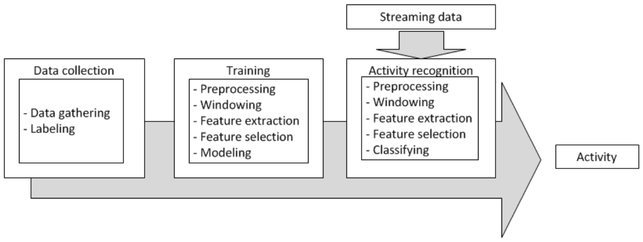


Fig. 1. Activity recognition process is divided into three main phases, and each of these can be divided into subphases [21].

Activity recognition was done using a sliding window technique. The signals from the sensors were divided into equal-sized smaller sequences, also called windows. Different window sizes were used with different data sets. Data sets that were found to be easy to classify were split to smaller windows that more challenging data sets as it has been shown that longer window size produces better results [22]. Window size of 1.0 seconds was used with SAD data set, 1.5 seconds with DHA, and 6.0 seconds with SBHAR. From these windows, features were extracted and finally the classification of the sequences was done using models trained based on these features. The features used in this study included for instance standard deviation, minimum, maximum, median, and different percentiles (10, 25, 75, and 90). Moreover, the sum of values above or below percentile (10, 25, 75, and 90), square sum of values above of below percentile (10, 25, 75, and 90), and number of crossings above or below percentile (10, 25, 75, and 90) were extracted and used as features. In addition, features from frequency domain, for instance sums of smaller sequences of Fourier-transformed signals, were extracted. Altogether 50 features were extracted from each available signal, which means 50 features were extracted from DHA (as it consists from accelerometer signal only), 100 features from SBHAR (accelerometer and gyroscope signals), and 150 features from SAD (accelerometer, gyroscope and magnetometer signals).

In order to achieve the highest possible recognition rates, the most descriptive features for each model were selected using a sequential forward selection (SFS) method [23]. Moreover, to reduce the number of misclassified windows, the final classification was done based on the majority voting of the classification results of three adjacent windows. Therefore, when an activity changes, a new activity can be detected when two adjacent windows are classified as a new activity.

It was decided to do the experiments using LDA (linear discriminant analysis) and QDA (quadratic discriminant analysis) as classifiers as in our previous studies (for instance [24], [20] and [25]) we have noticed they are not only accurate but also computationally light, and therefore, sufficient to be implemented to smartphones and used 24/7. In addition, they are fast to train. LDA is used to find a linear combination of features that separate the classes best. The resulting combination may be employed as a linear classifier. QDA is a similar method, but it uses quadratic surfaces to separate classes [26].

In the last stage of activity recognition process, using the trained recognition model, an unknown streaming signal can be classified. Before its class label can be defined, new signal must be processed in the same way as training data was processed when recognition models were trained, see Figure 1. Therefore, at first, streaming data is pre-processed and windowed. Then, selected features are extracted from the window and these are given as input to the trained recognition model to obtain the predicted activity class. Note that when new data is classified, in each stage, the same parameters must be used that were used to train the models [27].

III. Improving the human activity recognition process by adding noise

As already told, training data can never contain information about all the things that can happen in real-life. Therefore, the recognition model can never be perfect. In our study, the idea is to expand the area covered by training data by noise injection, and in this way, make the models trained using it more general and, therefore, more accurate as they are less vulnerable to changing conditions.

In our approach, to expand the area covered by the training data, noise injection is applied to 5-fold cross-validation, see Figure 2. To build user-dependent models, firstly features are extracted are from the collected personal data set. Secondly, five copies are made from this feature matrix, and each of these copies equals to one fold in cross-validation. To avoid over-fitting, data set in each copy needs to be different. In the approach presented in this paper, this differentiation is obtained by injecting noise to the data of each copy, and to ensure that each copy is different, different amount of noise is injected to the each copy. Recognition models are then trained based on these created data sets using cross-validation where each copy equals one fold in cross-validation.

It is explained in Algorithm 1 how noise is injected. As an input the noise injection algorithm gets the feature matrix extracted from data set A, labels related to it, and the amount of noise that will be injected. Noise injection is done class-wise: the amount of noise injected to features is dependent of how much the values of certain feature vary within the class. If the amount of noise injected would not be done class-wise but by data set-wise, the impact of noise injection would be relatively much bigger for classes where values of features do not vary much compared to classes where values vary more. This would make the resulting data set imbalanced. In the noise injection algorithm, for each instance of the feature matrix a random value between 0 and 1 is generated, and this value is multiplied with maximum noise and the variance related to this feature with same class label. Variance of feature is calculated as a remainder of percentiles 95 and 5. Maximum and minimum are not used to avoid the effect of outliers. The value from this multiplication is added to the original value of matrix instance, and when this is done to each instance of the feature matrix, the result is a feature matrix with noise injected.

IV. Experiment protocol and results

The proposed method was tested using the experiment protocol presented in Figure 3. For this purpose, the data from each subject was divided into half so that both halves contained as many observations from each activity. These halves were used to train and test the recognition models: 50% of the data was used for training and testing (referred as data set A in Figure 3) and the other 50% for validation (referred as data set B in Figure 3). This means that same data was never used both training and validation. After the data were split, Algorithm 1 was applied to the data set A to obtain five copies of data with noise injected. Cross-validation was applied to these, as well as SFS to select the 20 best features. These were mostly features from accelerometer signal and only a couple features from other signals were selected. Using these features, the final recognition model was trained. This model was then validated using data set B.

To show the benefit of the proposed method, noise injection was tested with different noise levels (0.05, 0.10, 0.15, 0.20, 0.25, and 0.30). For instance, noise level 0.20means that data set with noise injection includes 20% more variance than the original data set (Algorithm 1). As a reference, these results were compared to traditional method, where noise was not injected. Figure 4 shows how models without noise injection were trained. In this case, copies were not made from data set A as they would be identical. Instead, data was randomly divided into five parts, and then the recognition model was trained from these by training models using 5-fold cross-validation. Then again SFS was used to select 20 features. These tests were made to all three data sets (SAD, SDHAR, DHA) presented in Section II-A and experiments were done using LDA and QDA classifiers. For each noise level and classifier combination the experiment was performed 10 times.

To make experiment section even more extensive, the proposed method was also tested using data set B as training and data set A for validation. Cases where the first 50% of the data is used for training are referred as SAD 1, SDHAR 1, and DHA 1 and cases where the second 50% is used for training are referred as SAD 2, SDHAR 2, and DHA 2.

The results for both cases are presented in Table I and they are shown using TPR (True Positive Rate), FPR (False Positive Rate) and PR (precision) metrics a proposed in [28]. The equations used were:

$$TPR = \frac{TruePositives}{TruePositives + FalseNegatives}, \qquad (1)$$

$$FPR = \frac{FalsePositives}{TrueNegatives + FalsePositives}, \qquad (2)$$

$$PR = \frac{TruePositives}{TruePositives + FalsePositive}. \qquad (3)$$

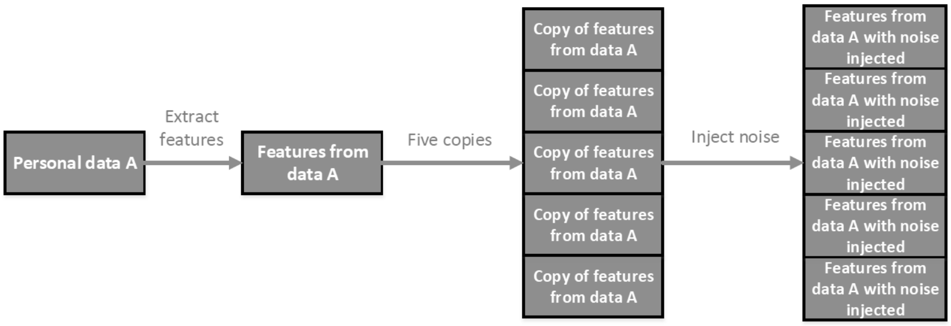

Personal data A
Extract features
Features from data A
Five copies
Copy of features from data A
Copy of features from data A
Copy of features from data A
Copy of features from data A
Copy of features from data A
Inject noise
Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected

Fig. 2. In the proposed method, several copies are made from the personal data set, and noise is injected to these copies.

```
Algorithm 1: Noise injection algorithm.
input : Feature matrix $matrix$, Class labels $labels$, Maximum noise level $noise\_max$
output: Feature matrix with noise injected $matrix\_n$
for $i \leftarrow 1$ to $5$ do
    $noise = (i-1) * noise\_max/4$;
    for $j \leftarrow 1$ to $unique(labels)$ do
        $ind = find(labels == j)$;
        $f = matrix(ind, :)$;
        for $k \leftarrow 1$ to $columns(f)$ do
            $matrix\_n(ind:k) = f(j) + randn(1:length(f)) * (prctile(f(j), 95) - prctile(f(j), 5)) * noise$;
        end
    end
end
return $matrix\_n$;
```

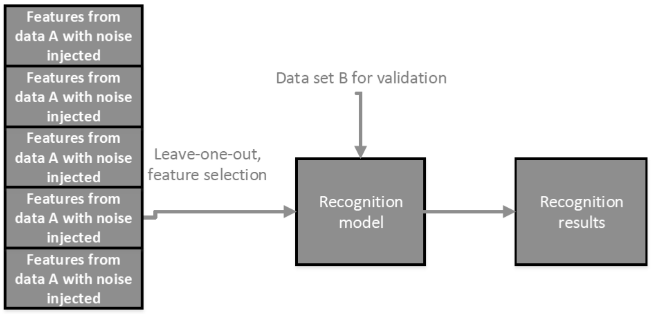

Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected
Features from data A with noise injected
Leave-one-out, feature selection
Data set B for validation
Recognition model
Recognition results

Fig. 3. Experimental protocol to test noise injection. Data set A was used for training and data set B for validation.

V. Discussion

It can be seen from the results shown in Table I that the proposed method does improve the true positive recognition rates. With SAD data set the improvement varies from 1.1 to 2.0 percentage units, with SBHAR from 1.4 to 4.5 percentage units, and with DHA the highest improvement is 2.5 percentage units and in one case (DHA 1 with LDA) there is no improvement. Moreover, in most cases noise injection reduces the variance between results. In addition, in each case the proposed improves precision. This improvement is especially notable with SBHAR data set because recognition rates with SBHAR are lower than with SAD and DHA. Therefore, these results are easier to improve. False positive rate is also lower with noise injection than without. When the results with and without noise injection from the tested 12 scenarios are compared in more detail, it can be noted that according to the paired t-test the improvement is statistically significant in 9 cases out of 12. As a conclusion, the results of Table I show that the models trained using the presented method are more general than the ones trained without adding any noise, and therefore, more accurate.

The results presented in Table I show that the improvement using SBHAR is a lot bigger than with other two data sets. The reason for this is that SBHAR data set has a lot less data per test subject than DHA and SAD. In SAD there is on average 4844 instances per test subject, in SBHAR 441 instances, and in DHA 1064 instances. In fact, in the Figure 5 it is shown how the true positive recognition rates of DHA and SAD data set change when less data is used for training. In Table I 50% of the data was used for training and the experimental results shown in Figure 5 are obtained using 25%, 15% and 12% of data from DHA 1 and SAD 1 for training. The trained models were validated using DHA 2 and SAD 2 data sets. These experiments were performed using noise levels 0.05, 0.15 and 0.25 and compared to case where no noise was added.

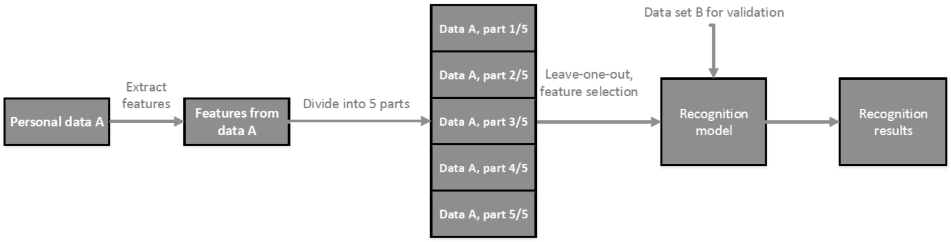


Fig. 4. As a reference, experiments were using data set where noise was not injected as a training data.

It can be noted that in the most cases the improvement is the bigger the less we have data. For instance using DHA data set and LDA classifier, true positive rate improves almost 5%-units when noise injection is used (Figure 5(a)). This means that the presented noise injection method enables training of accurate personal models using less data than we normally would need. Therefore, when the the presented method is used to train personal recognition models, the personal data gathering session does not need to be long. This is a huge improvement to the current situation as the problem with personal recognition models is that they require personal data for training which means that it is necessary to arrange a personal data gathering session, which can be frustrating and difficult to arrange. However, the results presented in Table I and Figure 5 show than when the presented noise injection method is used, the data gathering session can be shorter and less painful.

The noise injection algorithm was tested with six different parameters that defined how much noise was added to the training data. Using SAD data sets, the highest recognition rates were obtained when only small amount of noise was added while with DHA and SDHAR performed better when more noise was added. It should be further studied what is the reason for this, it can be related for instance to the types of the recognized activities, used sensors or the amount of data.

VI. Conclusions

In this paper, a noise injection method to improve the personal recognition models was presented. The idea of the method is to build more general recognition models by expanding the area covered by training data by using noise injection. This way, it is possible to train models that are and less vulnerable to changing conditions, and thus more accurate and more usable in eHealth applications. The method was tested using three human activity recognition data sets and two classifiers (LDA and QDA). The trained models were personal, half of the data of these data sets were used for training and the other half for validation.

The results shown in Table I show that personal models can be improved and made more general using the noise injection algorithm presented in this study. With SAD data set the improvement varies from 1.3 to 2.0 percentage units, with SBHAR from 1.4 to 4.5 percentage units, and with DHA the highest improvement was 2.5 percentage units. Moreover, the method improves precision and reduces false positive rates. In addition, the results of Figure 5 show that recognition rates of DHA and SAD data sets improve even more when less than half the data was used for training. It was noted that the improvement obtained using the presented noise injection method is in most cases the bigger the smaller the original training data set is. This means that it is possible to train accurate personal models with small training data set if noise injection is used. Therefore, when the noise injection is used the data gathering session for personal models can be very short, which means that it is less disruptive for the user.

It should also be noted that while in this study the presented method is used with human activity recognition data sets, it is not dependent on the application area and can be used with any time series data sets. In addition, although in this study LDA and QDA classifiers were used because they are accurate and fast to train. However, the presented method is model independent, and therefore, it can be used with any classification algorithm.

Acknowledgment

The authors would like to thank Infotech Oulu for funding this work.

References

[1] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors." *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[2] M. Kose, O. Incel, and C. Ersoy, "Online human activity recognition on smart phones," in *Workshop on mobile sensing: from smartphones and wearables to big data (colocated with IPSN)*, 2012, pp. 11–15.

[3] P. Siirtola and J. Röning, "Reducing uncertainty in user-independent activity recognition - a sensor fusion-based approach," in *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016)*, February 2016, pp. 611–619.

[4] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10, 2010, pp. 71–84.

[5] J. Leppänen and A. Eronen, "Accelerometer-based activity recognition on a mobile phone using cepstral features and quantized gmms," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 3487–3491.

[6] M. Albert, S. Toledo, M. Shapiro, and K. Kording, "Using mobile phones for activity recognition in parkinsons patients," *Frontiers in neurology*, vol. 3, pp. 1–7, 2012.

[7] G. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *AAAI Workshop on Activity Context Representation: Techniques and Languages*, 2012, pp. 98–104.

TABLE I

User-dependent recognition rates with and without noise injection using three data sets, and LDA and QDA classifiers.

SAD 1

	TPR	FPR	PR
LDA			
without noise	86.8% (1.1)	1.9% (0.2)	80.9% (2.3)
noise 5	88.6% (0.5)	1.6% (0.1)	82.6% (1.5)
noise 10	**88.8%** (0.6)	1.6% (0.1)	82.3% (1.1)
noise 15	88.6% (0.5)	1.6% (0.1)	82.1% (1.5)
noise 20	88.3% (0.6)	1.7% (0.1)	83.0% (1.5)
noise 25	88.3% (0.9)	1.7% (0.1)	83.0% (2.1)
noise 30	88.1% (1.0)	1.7% (0.1)	82.0% (2.0)
QDA			
without noise	90.6% (0.4)	1.3% (0.1)	92.3% (1.2)
noise 5	**91.7%** (0.8)	1.2% (0.1)	93.1% (0.7)
noise 10	91.4% (0.7)	1.2% (0.1)	92.4% (0.7)
noise 15	91.6% (0.6)	1.2% (0.1)	91.2% (0.6)
noise 20	91.1% (0.5)	1.3% (0.1)	89.4% (0.8)
noise 25	89.7% (1.3)	1.5% (0.2)	87.7% (1.3)
noise 30	89.6% (0.5)	1.5% (0.1)	85.8% (1.5)

SAD 2

	TPR	FPR	PR
LDA			
without noise	89.2% (1.3)	1.5% (0.2)	84.8% (2.0)
noise 5	90.0% (0.5)	1.4% (0.1)	85.2% (1.0)
noise 10	89.9% (0.4)	1.4% (0.1)	85.9% (1.5)
noise 15	90.1% (0.9)	1.4% (0.1)	86.1% (2.5)
noise 20	**90.4%** (1.0)	1.4% (0.1)	86.1% (1.7)
noise 25	**90.4%** (0.4)	1.4% (0.1)	87.7% (1.1)
noise 30	90.1% (0.6)	1.4% (0.1)	88.0% (1.5)
QDA			
without noise	89.9% (0.7)	1.4% (0.1)	89.7% (1.1)
noise 5	**91.2%** (0.5)	1.3% (0.1)	91.2% (1.4)
noise 10	91.1% (0.7)	1.3% (0.1)	90.0% (1.1)
noise 15	91.0% (0.6)	1.3% (0.1)	89.4% (1.5)
noise 20	90.5% (0.5)	1.4% (0.1)	87.7% (1.9)
noise 25	89.9% (0.8)	1.4% (0.1)	86.0% (1.4)
noise 30	88.9% (0.6)	1.6% (0.1)	86.0% (2.1)

SBHAR 1

	TPR	FPR	PR
LDA			
without noise	83.9% (2.5)	2.7% (0.4)	79.8% (8.0)
noise 5	84.3% (2.3)	2.6% (0.3)	79.3% (7.8)
noise 10	85.2% (1.8)	2.5% (0.3)	81.6% (7.0)
noise 15	85.2% (2.4)	2.4% (0.4)	81.9% (9.6)
noise 20	**85.6%** (2.2)	2.4% (0.4)	82.9% (8.4)
noise 25	85.5% (2.4)	2.5% (0.4)	84.5% (5.8)
noise 30	85.1% (1.8)	2.5% (0.3)	83.0% (8.3)
QDA			
without noise	80.2% (2.6)	3.3% (0.4)	76.8% (10.8)
noise 5	80.2% (2.5)	3.3% (0.4)	73.2% (11.7)
noise 10	82.2% (2.3)	3.0% (0.4)	79.4% (9.1)
noise 15	83.8% (2.4)	2.8% (0.4)	82.8% (10.0)
noise 20	84.1% (2.4)	2.7% (0.4)	82.8% (9.7)
noise 25	84.4% (2.4)	2.6% (0.4)	83.6% (7.9)
noise 30	**84.7%** (2.3)	2.6% (0.3)	84.6% (9.2)

SBHAR 2

	TPR	FPR	PR
LDA			
without noise	82.3% (2.5)	3.0% (0.4)	79.8% (9.8)
noise 5	83.5% (2.1)	2.8% (0.3)	82.5% (6.3)
noise 10	83.0% (2.3)	2.9% (0.4)	82.1% (9.7)
noise 15	83.6% (1.7)	2.8% (0.3)	84.0% (8.1)
noise 20	**83.7%** (2.0)	2.8% (0.3)	84.9% (8.1)
noise 25	83.3% (1.9)	2.9% (0.3)	85.2% (6.7)
noise 30	82.3% (2.3)	3.0% (0.4)	83.7% (7.8)
QDA			
without noise	82.5% (2.6)	3.0% (0.5)	80.1% (7.9)
noise 5	83.1% (2.8)	2.9% (0.4)	78.7% (11.0)
noise 10	84.6% (2.5)	2.6% (0.4)	85.0% (7.5)
noise 15	**85.1%** (1.7)	2.5% (0.3)	85.7% (5.0)
noise 20	85.4% (1.7)	2.5% (0.3)	86.5% (6.1)
noise 25	85.3% (1.8)	2.5% (0.3)	86.8% (7.0)
noise 30	84.7% (2.1)	2.6% (0.3)	87.0% (5.2)

DHA 1

	TPR	FPR	PR
LDA			
without noise	**94.0%** (0.5)	1.3% (0.1)	84.7% (2.0)
noise 5	93.5% (0.6)	1.4% (0.1)	85.6% (1.4)
noise 10	93.9% (0.5)	1.3% (0.2)	85.2% (0.9)
noise 15	**94.0%** (0.3)	1.3% (0.1)	85.5% (1.5)
noise 20	**94.0%** (0.4)	1.3% (0.1)	85.5% (1.7)
noise 25	**94.0%** (0.4)	1.4% (0.2)	84.9% (3.4)
noise 30	93.6% (0.3)	1.4% (0.1)	85.6% (1.5)
QDA			
without noise	92.3% (0.8)	1.6% (0.2)	89.2% (1.3)
noise 5	93.8% (1.1)	1.2% (0.2)	92.1% (1.9)
noise 10	94.3% (0.7)	1.1% (0.3)	92.2% (1.6)
noise 15	94.4% (0.6)	1.1% (0.2)	91.3% (1.4)
noise 20	94.7% (0.7)	1.1% (0.2)	91.7% (1.3)
noise 25	94.6% (0.9)	1.1% (0.3)	91.8% (1.2)
noise 30	**94.8%** (0.3)	1.1% (0.2)	91.9% (1.1)

DHA 2

	TPR	FPR	PR
LDA			
without noise	93.9% (1.3)	1.3% (0.3)	86.2% (1.6)
noise 5	94.2% (0.8)	1.2% (0.2)	86.8% (0.7)
noise 10	**94.3%** (1.1)	1.3% (0.3)	86.6% (1.7)
noise 15	94.1% (0.7)	1.3% (0.2)	85.9% (1.0)
noise 20	94.0% (0.7)	1.3% (0.2)	86.0% (0.9)
noise 25	94.1% (0.6)	1.3% (0.2)	86.0% (0.9)
noise 30	93.5% (0.9)	1.5% (0.3)	84.9% (1.5)
QDA			
without noise	94.7% (1.0)	1.2% (0.2)	91.3% (1.8)
noise 5	95.7% (0.9)	0.9% (0.2)	92.8% (1.2)
noise 10	95.3% (0.7)	1.0% (0.1)	92.2% (1.0)
noise 15	95.7% (0.5)	0.9% (0.1)	92.7% (1.2)
noise 20	**96.2%** (0.5)	0.8% (0.1)	92.9% (0.6)
noise 25	**96.2%** (0.4)	0.8% (0.1)	93.5% (0.6)
noise 30	96.0% (0.5)	0.8% (0.1)	92.7% (0.6)

[8] H. Koskimäki and P. Siirtola, "Adaptive model fusion for wearable sensors based human activity recognition," in *Information Fusion (FUSION), 2016 19th International Conference on*. ISIF, 2016, pp. 1709–1713.

[9] A. Neumaier, "Solving ill-conditioned and singular linear systems: A tutorial on regularization," *SIAM review*, vol. 40, no. 3, pp. 636–666, 1998.

[10] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent, "Adding noise to the input of a model trained with a regularized objective," *arXiv preprint arXiv:1104.3250*, pp. 1–13, 2011.

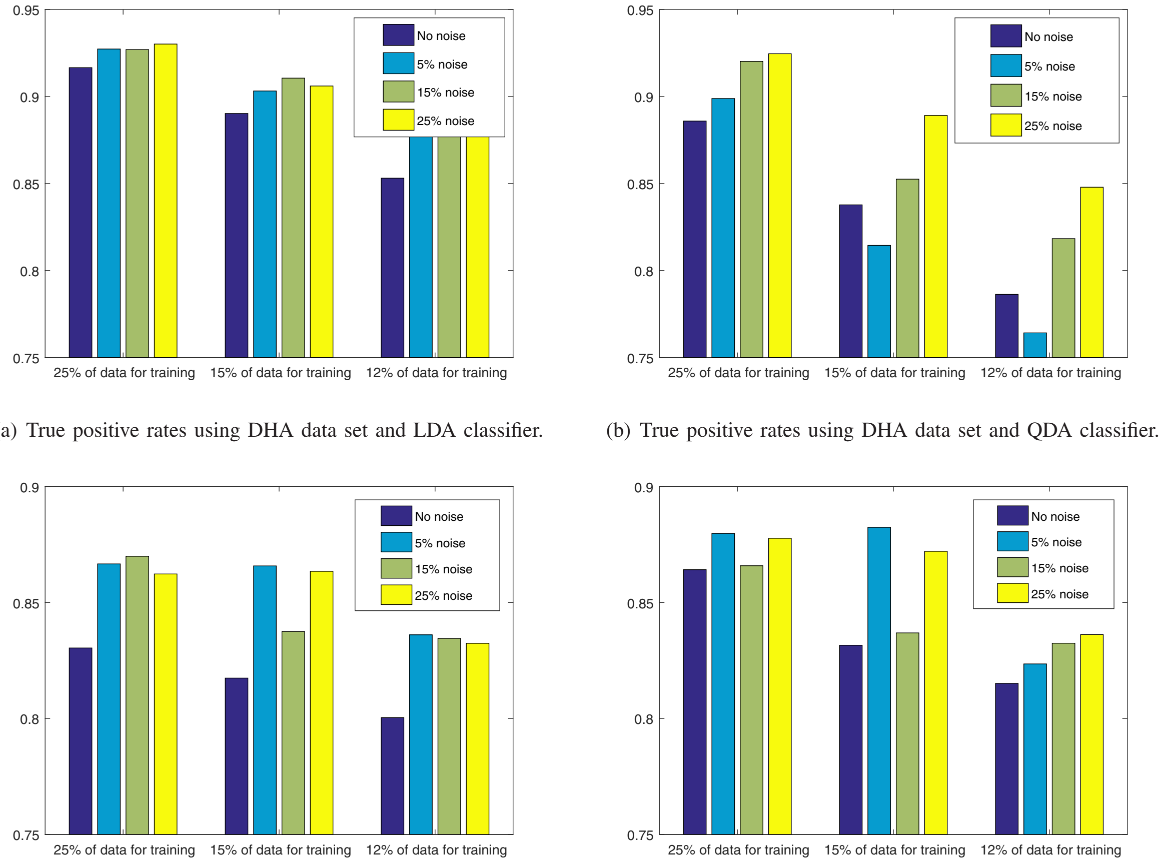
[11] R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif, "Misleading worm signature generators using deliberate noise injection," in *2006 IEEE Symposium on Security and Privacy (S P'06)*, May 2006, pp. 15 pp.–31.

[12] K. A. Toh, X. Jiang, and W.-Y. Yau, "Exploiting global and local decisions for multimodal biometrics verification," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3059–3072, Oct 2004.

[13] F. Bahrpeyma, C. Cranganu, and B. Golchin, *Improving the Accuracy of Active Learning Method via Noise Injection for Estimating Hydraulic Flow Units: An Example from a Heterogeneous Carbonate Reservoir*. Cham: Springer International Publishing, 2015, pp. 225–244.

[14] L. Piras and G. Giacinto, "Synthetic pattern generation for imbalanced learning in image retrieval," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2198–2205, 2012.

[15] R. Somorjai, A. Janeliunas, R. Baumgartner, and S. Raudys, *Comparison of Two Classification Methodologies on a Real-World Biomedical*



(a) True positive rates using DHA data set and LDA classifier.

(b) True positive rates using DHA data set and QDA classifier.

(c) True positive rates using SAD data set and LDA classifier.

(d) True positive rates using SAD data set and QDA classifier.

Fig. 5. When the training data set is small, the improvement obtained using noise injection method is bigger.

Problem. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 433–441.

[16] M. Skurichina, Š. Raudys, and R. P. Duin, "K-nearest neighbors directed noise injection in multilayer perceptron training," *Neural Networks, IEEE Transactions on*, vol. 11, no. 2, pp. 504–511, 2000.

[17] J. Ni, T. Lambrou, and X. Ye, "Learning from life-logging data by hybrid HMM: A case study on active states prediction," in *12th international Conference on Biomedical Engineering Biomedical Engineering (BioMed 2016)*, Innsbruck, Austria, 15 16 2016, pp. 70–75.

[18] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/6/10146

[19] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[20] P. Siirtola and J. Röning, "Ready-to-use activity recognition for smartphones," in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2013)*, April 16–19 2013, pp. 59–64.

[21] O. Incel, M. Kose, and C. Ersoy, "A review and taxonomy of activity recognition on mobile phones," *BioNanoScience*, vol. 3, no. 2, pp. 145–171, 2013. [Online]. Available: http://dx.doi.org/10.1007/s12668-013-0088-3

[22] S. A. Bashir, D. C. Doolan, and A. Petrovski, "The effect of window length on accuracy of smartphone-based activity recognition," *IAENG International Journal of Computer Science*, vol. 43, no. 1, pp. 126–136, March 2016.

[23] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach*. Prentice Hall, 1982.

[24] P. Siirtola and J. Röning, "Recognizing human activities user-independently on smartphones based on accelerometer data," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 1, no. 5, pp. 38–45, June 2012.

[25] H. Koskimäki and P. Siirtola, "Recognizing gym exercises using acceleration data from wearable sensors," in *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, Dec 2014, pp. 321–328.

[26] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining*. Cambridge, MA, USA: MIT Press, 2001.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[28] J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, pp. 6:1–6:23, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1889681.1889687