

Human Activities Recognition with a Single Wrist IMU via a Variational Autoencoder and Android Deep Recurrent Neural Nets

Edwin Valarezo Añazco¹, Patricio Rivera Lopez¹, Hyemin Park¹, Nahyeon Park¹,
and Tae-Seong Kim^{1*}

¹ Department of Biomedical Engineering, College of Electronics and Information,
Kyung Hee University, Republic of Korea
{edgivala, patoalejor, hmp9669, nhpark, tskim}@khu.ac.kr

Abstract. Human Activity Recognition (HAR) is an active research field because of its versatility towards various application areas such as healthcare and lifecare. In this study, a novel HAR system is proposed based on an autoencoder for denoising and Recurrent Neural Network (RNN) for classification with a single Inertial Measurement Unit (IMU) located on a dominant wrist. A Variational Autoencoder (VAE) is built to denoise IMU signals which improves HAR by Android Deep RNN. Evaluating our VAE and Android Deep RNN HAR system is done in two ways. First, the system is tested on a PC using discrete epochs of activities of daily living. Our results show that VAE improves Signal-to-Noise Ratio (SNR) of the IMU signal from 8.78 to 17.26 dB. In turn, HAR improves from 89.29% to 95.11% in F1-score and from 90.38% to 95.47% in accuracy. Secondly, the system is tested on an Android device (i.e., smartphone) using continuous activity signals. This is done by transferring the PC HAR system to an Android HAR App (i.e., Android Deep RNN). We have achieved 86.13% and 95.09% in accuracy without and with VAE respectively. Our results demonstrate that HAR can be achieved in real-time on a standalone smart device with a single IMU for lifelogging services.

Keywords: Human Activity Recognition; Denoising Autoencoder; Android Deep Recurrent Neural Networks; Mobile Application.

1. Introduction

Human Activity Recognition (HAR) is defined as a context-aware technology that maps activity data collected by IoT sensors into activity labels [1]. In recent years, HAR has gained strong attention because of its various applications [2], such as fall detection [3] [4], rehabilitation [5] [6], activity recognition in sports [7] [8], sports training analysis

* Corresponding author.

Corresponding author address: Room 719, College of Electronics and Information, Kyung Hee University,
1732, Deogyong-daero, Giheung-gu, Yongin-Si, Gyeonggi-do 17104, Republic of Korea.
Corresponding author office Tel.: +82-31-201-3731.
Corresponding author email.: tskim@khu.ac.kr.

[9] [10], energy expenditure estimation [11], and health care [12] [13]. Two issues have recently been raised regarding advanced HAR. First, in HAR studies, activity noises affect the recognition rate [14]. Then, it is necessary to use filters to denoise the motion signals. Second, most HAR studies are implemented on a PC platform. There is a strong need for a practical HAR system on a portable platform, such as smartphones.

Recently Deep Learning (DL) has been adopted to create a better representation of the signal with less noise via Denoising Autoencoder (DAE) [15] [16]. Some studies tested DAE for instance: in Heyman, et al. [17], Bayesian Feature Enhancement (BFE) and DAE were used to clean-up voice signals for speech recognition using the output of BFE as target data. In Xiong, et al. [18], wavelet transform with the scale-adaptive thresholding method was used to filter out noise and DAE to remove the residual noise from electrocardiogram signals. In both cases, additional denoising stages were used as target data for DAE. Then, the performance of DAE depends on the target. In contrast, a totally unsupervised DAE is possible (i.e., using only autoencoder for denoising). For instance, in Mohammed, et al. [14], Variational Autoencoder (VAE) produced desirable denoising performance for motion artifact from a sensor attached to clothes. Our VAE implementation focuses on denoising activity signals from a wrist Inertial Measurement Unit (IMU).

The current popularity of wearable technology creates an invaluable opportunity to build a portable HAR system. However, building an HAR system using DL on a portable device such as smartphones require some considerations. Most of all, large and complex classification algorithms could not be implemented on a portable device because of the computational burden on limited computing power and memory. There are some recent works trying to implement HAR systems on portable devices. For instance, Lane and Georgiev [19] used a low-power Deep Neural Network (DNN) on a CPU paired with a mobile device. In their system, DNN worked with a cloud system (i.e., activity label was inferred on PC and then sent to the smartphone via the Internet). However, the cloud system with a web connection is limited to not be a standalone HAR system. To overcome the limitations of the cloud service computation, some studies implemented their classification algorithms in a standalone mode (i.e., running on a smartphone) [20] [21] [22] [23] [24]. However, conventional classification approaches are used, instead of DL. For instance, Jongprasithporn, et al. [22] applied thresholds values to classify standing, walking, and running activities. Due to the threshold values were fixed for the three activities, complex activities could not be recognized. The other studies used Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN) with hand-extracted features to differentiate sitting, standing, lying, walking, running, and cycling. DL should improve the performance over the mentioned conventional classification approaches and avoid the needed of hand-extracted features [25].

In this study, we have developed a HAR system with a single IMU at a dominant wrist based on VAE and Android Deep RNN. Our contributions focus on solving the problems of noise in HAR signals and implementation of a reliable HAR system on portable devices. To solve the problem of noise in the HAR signals, a denoising stage uses VAE based on Convolution Neural Network (CNN) and Recurrent Neural Network (RNN). VAE enhances the IMU signal by approximating the spatial-temporal features to Gaussian distributions. The importance of the denoising stage lays in the fact that IMUs are prone to motion artifacts and sensor noise. To implement a reliable HAR system we used RNN because it outperforms other algorithms in HAR [25] [26]; to

make the HAR system portable avoiding the limitations of cloud systems (e.g., connection delays), the HAR system was implemented under Android (i.e., Android Deep RNN) whereas most DL RNNs are implemented only for a PC [27] [28] [29]. RNN running on Android increases the scope of our system to any smart device using Android, even if the smart device has not internet connection. Our results show that HAR accuracy increases with the denoised IMU signal compared to the raw signal without VAE. Furthermore, the inference time of our Android Deep RNN on Android smartphone is fast enough to implement real-time lifelogging App.

2. Methodology

The proposed system is represented in Fig. 1. Human activities are sensed with a single IMU worn on one dominant wrist. In our HAR system, VAE denoises a tri-axial accelerometer and tri-axial gyroscope signals. Then the denoised IMU signals are classified into activity labels via Android Deep RNN on the PC and Android smartphone platforms. In this study, we recognize Activities of Daily Living (ADL) including standing, walking upstairs, walking downstairs, walking, running, cycling, and Nordic walking.

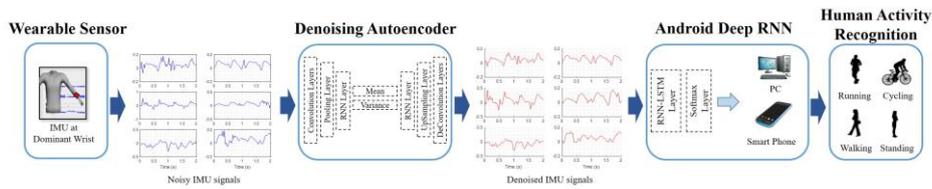


Fig. 1. HAR System workflow.

2.1. ADL Database

In this study, we used a public database (DB), Physical Activity Monitoring for Aging People (PAMAP2) [30] [31]. PAMAP2 includes raw information using three IMUs located on the chest, the wrist, and the ankle, with a sampling frequency of 100 Hz. Each IMU provides the information of tri-axial acceleration (ms^{-2}), tri-axial gyroscope (rad/s), and tri-axial magnetometer (μT). The DB includes data of nine subjects performing 12 ADLs, including ironing, vacuum cleaning, walking, walking up and downstairs, sitting, lying, and standing. High-level dynamic activities have also included, such as Nordic walking, running, cycling, and rope jumping.

For this study, we have selected the following seven ADLs: *Standing* consists of just standing without interaction with another person or standing while the subject is talking. *Walking Upstairs* and *Walking Downstairs* (Ascending and Descending stairs) were carried out in a five-floors building. *Walking* consists of walking outdoors with moderate speed steps. *Nordic Walking* is performed on asphaltic terrain using asphalt pads on the walking poles. *Cycling* is conducted outdoors with a real bike. *Running*

corresponds to jogging outdoors with suitable speed. These ADLs are selected because they should be well reflected on the wrist sensor and most of the subjects perform these activities for all the trials.

2.2. Denoising Autoencoder

Autoencoder (AE) is a feed-forward neural network that reproduces its input as output. It includes an encoder network that generates a featured representation of the input in the latent space. Then the decoder network decodes the feature representation back to the input dimension [32].

VAE is a variant of autoencoder that is able to denoise motion signals. As autoencoder, VAE encodes the input data into a feature-representation vector in the latent space. However, in VAE, the feature representation vector is constrained to Gaussian distributions [33]. Also, VAE uses two loss functions, a combination of Kullback-Leibler divergence (KL) and Mean Square Error (MSE). The first loss function forces the feature-representation vector in the latent space to follow a Gaussian distribution, measuring the relative entropy between the approximate posterior and the prior probability density function [33]. The second loss measures the similarity between the output and the input.

Our VAE model is presented in Fig. 2. It uses a combination of CNN and RNN with Long Short-Term Memory (i.e., LSTM) layers as in [14]. Nevertheless, our VAE implementation has two main differences from the previous approaches [34] [14] [33]. The first difference lies in our VAE architecture. Our VAE has three convolutional layers and three LSTM layers in the encoder. The latent space has two dense layers. The decoder has three LSTM layers and three deconvolutional (ConvTranspose) layers. The second difference is the loss function. It is a weighted combination of KL and MSE similar to Mohammed and Tashev [14]. The weighted loss function is designed by analyzing the effect of KL and MSE in the Signal-to-Noise Ratio (i.e., SNR). Eq. 1 describes the weighted loss function used to train VAE. E_{KL} measures the KL divergence between the prior and posterior probability density functions, described by the feature representation vector z in the latent space. E_{MSE} measures the MSE error between the output and input signals; γ is the weight value, in this work γ value is 5.

$$\mathcal{L} = \gamma * (E_{MSE}[(\hat{y} - y)^2]) + \frac{1}{1000 * \gamma} * (E_{KL}[q(z|x)||p(z)]) \quad (1)$$

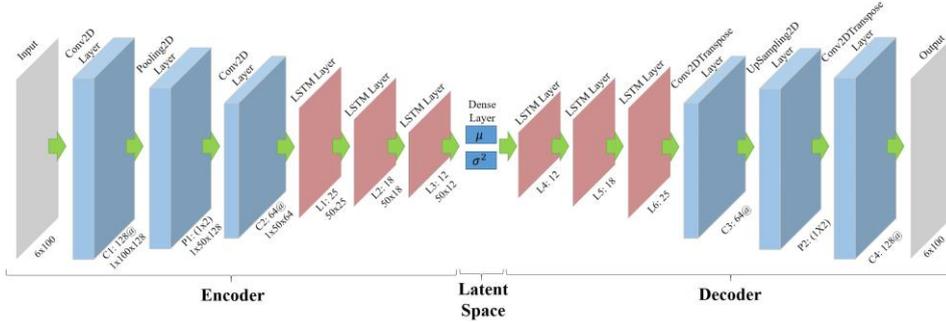


Fig. 2. Variational Autoencoder.

2.3. Android Deep RNN

RNN is characterized by recurrent connections between hidden units. These recurrent connections generate a temporal memory in which the previous state of the network is stored. This lets RNN take decisions based on the previous stage of the network and the current input [26].

Our Android Deep RNN uses the classic LSTM cell proposed in [35]. LSTM produces internal paths regulated by gates, where the gradient can flow for long durations. These regulated paths allow propagation of the error in deep networks. For weight optimization, Stochastic Gradient Descent (SGD) was used. SGD uses a momentum factor to determine how fast the algorithm converges. SGD is described by Eqs. 2 and 3.

$$W_{t+1} = W_t + v_{t+1} \tag{2}$$

$$v_{t+1} = \beta \times v_t - \alpha \times \nabla \mathcal{L}(W_t) \tag{3}$$

where α is learning rate, W_t weight matrix, \mathcal{L} the loss function, and β the momentum factor. The loss function is Negative Logarithmic Likelihood. It is described by Eq. 4, where n corresponds to the number of classes and P_y^i is the probability of the class i .

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^n \log(P_y^i) \tag{4}$$

Fig. 3 shows our Android Deep RNN, the main difference with previous works are: first, our Android Deep RNN can run on Android smartphone on standalone mode (i.e., inference on Android device), while most of the previous HAR systems are implemented only for PC [29] [28] [27]. Second, the architecture of our Android Deep RNN, for instance Milenkoski, *et al.* [36] built an HAR system using three LSTM layers with 64 cells each layer. In contrast, our network is lighter than them in terms of number of layers. It has only a single RNN-LSTM layer and a single hidden layer to reduce the memory usage on Android smartphone. The single RNN-LSTM layer has 100 cells

corresponding to 2 seconds data; the hidden layer has 120 nodes; at the end of our network a softmax layer with seven nodes, corresponding to each class.

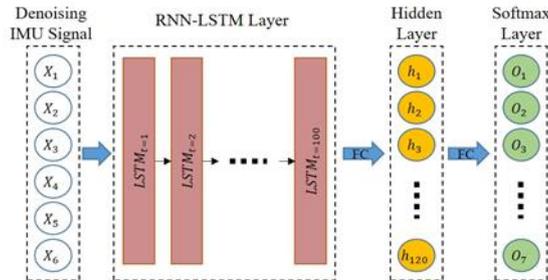


Fig. 3. Android Deep RNN Architecture. The six-channels input matches with the RNN-LSTM layer. The remaining layers are fully connected.

2.4. VAE and Android Deep RNN Implementation

In this study, VAE was implemented on a PC platform using Keras and TensorFlow. The convolutions and the RNN-LSTM layers were implemented to transform the spatial-temporal information of the IMU signals into a family of Gaussian distributions in the latent space with mean μ and variance σ^2 . The mean μ and variance σ^2 are represented as dense layers (see Fig. 2). Mini-batch approach was used to train VAE with a mini-batch size of 100. Adam Optimizer was used with a learning rate of 0.0008. The kernels size and the hidden nodes are same for each encoder-decoder equivalent layer.

The Android Deep RNN was implemented using Deeplearning4J library [37]. The RNN-LSTM layer of our HAR system classifies the data as many-to-one, after an epoch (i.e., 2 seconds data) the model infers a single class label. Truncated Backpropagation Through Time (Truncated-BPTT) was used to train the Android Deep RNN. Truncated-BPTT is a variant of BPTT that restricts the downside to a specific number of hidden nodes (i.e., the gradient is not back-propagated to all hidden nodes). To train our HAR system the mini-batch approach was used with a mini-batch size of 113. Weight initialization was done using a random number generator. Hyperparameters such as learning rate and number of hidden nodes were fixed by analyzing the behavior of these in the loss function. The learning rate was set as 0.09 and the number of training steps was fixed after setting the learning rate and to avoid overfitting. The training of the Android Deep RNN was done using IntelliJ IDE on PC.

An Android app was created using Android Studio IDE with the Application Program Interface (API) 23. API 23 lets the app run on Android 6.0 or higher. The app loads the network architecture shown in Fig. 3 with the trained weights. Then, the inference is made on an Android smartphone without fine-tuning or retraining of the network on smartphone.

The smartphone used in this study is a Samsung Galaxy S7 with the following technical specifications: Android version 7.0, 4GB of RAM, Octa-core processor of

4x2.3 GHz and 4x1.6 GHz. The PC specification includes Processor Intel(R) Core (TM) i5-7500 CPU@ 3.40GHz, 8GB of RAM, and NVIDIA GeForce GTX 1050 Ti.

2.5. Data Preparation

In this study, we utilized activity information from a tri-axial accelerometer and a tri-axial gyroscope on a dominant wrist. We created a discrete epoch dataset, where each epoch is segmented data with a time window of two seconds.

The IMU signals were preprocessed as follows: The activity data was downsampled to 50 Hz, as was suggested in [38] for HAR. The gravity effect was removed by a high-pass Butterworth filter, with a cutoff frequency of 0.25 Hz [39]. The activity data were normalized between -1 to 1. Finally, the sliding window approach was used with an overlap of 50% and a window length of two seconds. The total number of epochs per activity are 610 epochs for standing, 1470 epochs for walking, 700 epochs for running, 1020 epochs for cycling, 1340 epochs for Nordic walk, 640 epochs for ascending stairs, and 500 epochs for descending stairs.

In addition, the performance of our HAR system on a smartphone was evaluated by a real-time simulation with continuous epochs of activities. The activity datasets of discrete and continuous epochs were created from seven subjects in PAMAP2.

2.6. Validations

Validation of our VAE was done by analyzing the SNR improvement. Eq. 5 describes mathematically the SNR. The SNR was calculated by taking the Root-Mean-Square level (i.e., RMS) of the signal of interest involving certain activities and the RMS level of noisy background signals (i.e., no activities performed).

$$SNR = 20 * \log_{10} \frac{RMS(\text{Signal of Interest})}{RMS(\text{Noise})} \quad (5)$$

The following performance metrics are considered to assess our HAR system [40].

$$\text{Recall or Sensitivity} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{Precision or Positive Predictive Values (PPV)} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (8)$$

$$\text{Negative Predictive Values (NPV)} = \frac{TN}{TN + FN} \quad (9)$$

$$F1 - \text{Score} = 2 \times \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (10)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

To validate the performance of our HAR system two evaluation methods were used. The first evaluation is done using the discrete epochs dataset on a PC. The performance of the HAR system was evaluated through a five-fold. The second validation methodology is a continuous epoch evaluation that runs on an Android smartphone. This test evaluates the feasibility and performance of a real-time HAR system under a standalone mode.

3. Results

3.1. Denoising Performance of VAE

Fig. 4 shows the comparison of some representative epochs from different activities before and after VAE. The effect of VAE is shown as a reduction of the high-frequency components in the denoised activity signals. The overall SNR is improved from 8.78 to 17.26 dB.

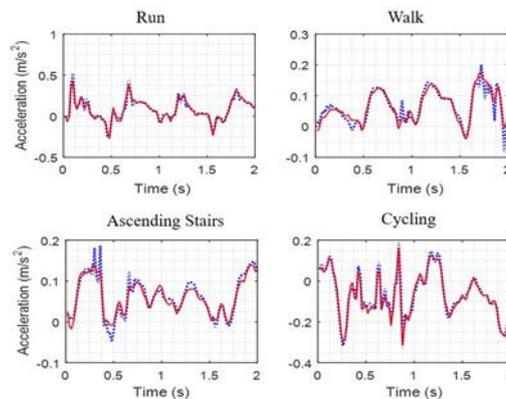


Fig. 4. Raw epochs (dotted in blue) vs. denoised epochs (solid in red) via VAE.

Fig. 5 shows the precision and recall values of our HAR system without and with VAE. With VAE there is an increment in precision for standing from 91.7% to 96.62%, for cycling from 80.07 to 98.85%, for ascending stairs from 86.42% to 88.76%, and for descending stairs from 78.98% to 87.57%. Also, there is an increase in recall for standing from 80.55% to 95.15%, for cycling from 90.94% to 96.62%, for ascending stairs from 80.46% to 89.27%, and for descending stairs from 90.85% to 96.73%. The remain ADLs have a minor improvement in precision and recall.

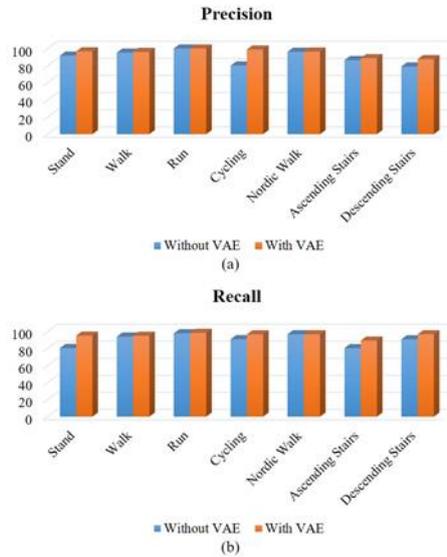


Fig. 5. Comparison of precision and recall with and without VAE. (a) the precision value of our HAR system. (b) the recall value of our HAR system.

3.2. HAR for Discrete Epochs on a PC Platform

The performance of the proposed HAR system without VAE is shown in Table 1 as a confusion matrix and in Table 2 as a summary of the performance metrics for each activity.

Table 1. HAR Confusion Matrix without VAE.

(%)	Stand	Walk	Run	Cycling	Nordic Walk	ASCD ^a Stairs	DESC ^b Stairs
Stand	80.55	0.00	0.00	13.68	1.52	1.22	3.04
Walk	1.26	93.97	0.00	0.50	0.50	1.51	2.26
Run	0.00	1.11	97.78	0.00	0.00	0.56	0.56
Cycling	4.15	0.00	0.00	90.94	0.38	0.75	3.77
Nordic Walk	0.32	0.97	0.00	0.97	96.76	0.65	0.32
ASCD Stairs	2.87	7.47	0.00	3.45	2.30	80.46	3.45
DESC Stairs	1.31	0.65	0.00	2.61	0.00	4.58	90.85

^a Ascending Stairs (ASCD Stairs)

^b Descending Stairs (DESC Stairs)

Table 2. Performance Metrics without VAE.

(%)	Stand	Walk	Run	Cycling	Nordic Walk	ASCD Stairs	DESC Stairs
Precision	91.70	95.17	100	80.07	96.14	86.42	78.98
NPV	95.79	98.30	99.75	98.41	99.33	97.93	99.14
Recall	80.55	93.97	97.78	90.94	96.76	80.46	90.85
Spec^c	98.38	98.65	100	96.11	99.20	98.65	97.76

^c Specificity

The confusion matrix of Table 3 shows the performance of our HAR system with VAE. There is an improvement in classification, especially for standing. The improvement is shown through a reduction in the confusion of standing against the rest of activities and through an increase in the performance metrics.

The values in the main diagonal of the confusion matrix correspond to the recall of each activity. Table 3 shows six of the seven activities (standing, walking, running, cycling, Nordic walking, and descending stairs) having the recall value up to 94.5%. Ascending and descending stairs caused some confusion due to their similarities. For these activities, the recall values are 89.27% and 96.73% respectively. On average, the F1-score increases from 89.29% to 95.11% and accuracy from 90.38% to 95.47%, if VAE is used.

Table 3. HAR Confusion Matrix with VAE.

(%)	Stand	Walk	Run	Cycling	Nordic Walk	ASCD Stairs	DESC Stairs
Stand	95.15	1.21	0.00	0.61	2.12	0.61	0.30
Walk	0.25	94.99	0.00	0.00	0.50	2.01	2.26
Run	0.00	0.55	98.34	0.00	0.00	1.10	0.00
Cycling	1.88	0.00	0.00	96.62	0.38	0.75	0.38
Nordic Walk	0.66	0.99	0.00	0.00	96.70	0.66	0.99
ASCD Stairs	1.13	4.52	0.00	0.56	0.56	89.27	3.95
DESC Stairs	0.65	0.00	0.00	0.00	0.00	2.61	96.73

Table 4 illustrates the performance metrics of our Android Deep RNN with VAE. Comparing to the results without VAE in Table 2, there is an increment in the performance metrics for all activities if VAE is used. On average, the precision increases from 89.78% to 94.88%, NPV from 98.38% to 99.23%, recall from 90.19% to 95.40%, and specificity from 98.39% to 99.24% with the use of VAE.

Table 4. Performance Metrics with VAE.

(%)	Stand	Walk	Run	Cycling	Nordic Walk	ASCD Stairs	DESC Stairs
Precision	96.62	95.95	100	98.85	96.38	88.76	87.57
NPV	98.92	98.59	99.82	99.42	99.34	98.84	99.70
Recall	95.15	94.99	98.34	96.62	96.70	89.27	96.73
Spec	99.26	98.87	100	99.81	99.27	98.77	98.73

3.3. HAR for Continuous Epochs on a Smartphone

Fig. 6 presents the recognized labels for two subjects without VAE, with VAE, and the ground-truth. Without VAE, there is confusion between standing and cycling for almost all epochs in the class of standing.

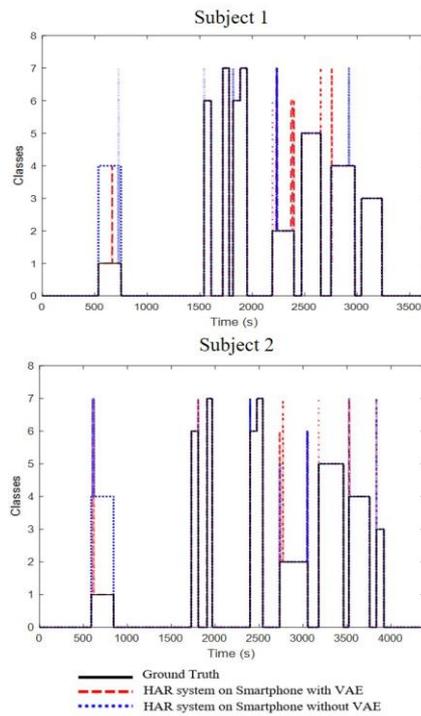


Fig. 6. HAR results via the continuous epochs evaluation on an Android smartphone: Label 0 corresponds to Null, Label 1 Standing, Label 2 Walking, Label 3 Running, Label 4 Cycling, Label 5 Nordic Walking, Label 6 Ascending Stairs, Label 7 Descending Stairs.

Table 5 shows a summary of the accuracy and inference time for all subjects on the Android smartphone. The accuracy per subject is the correctly recognized activities according to Eq. 11. The effect of VAE is reflected with an improvement in accuracy

for each subject. On average, our HAR system increases its accuracy by 8.97% with VAE.

Table 5. HAR results via the continuous epochs evaluation on an Android smartphone.

Subjects	Signal Duration [s]	Accuracy Without VAE [%]	Accuracy With VAE [%]	Inference Time On Smartphone [s]
1	3672	82.64	96.53	709
2	4386	79.65	98.02	847
3	2467	90.33	96.19	477
5	3654	81.54	87.52	707
6	3550	80.54	96.78	686
7	3073	95.30	97.07	594
8	4003	92.89	93.56	774

4. Discussion

This study introduces a HAR system that uses only one IMU on the dominant wrist to recognize ADLs. The proposed system uses VAE to denoise IMU time-series signals. The Android Deep RNN classifies the denoised epochs into activity labels. To make our HAR system practical, our Android Deep RNN was implemented in a standalone mode on a smartphone to perform HAR in real-time using continuous epochs.

Our experiments showed that VAE has a positive effect on our HAR system. VAE reconstructs the IMU signals as less-noisy signals by using the features in the IMU signals because the SNR improves around 9 dB. Overall, the F1-score improves by 6%, and accuracy improves by 5% using VAE.

For statistical significance of performances, a paired T-test, i.e., performance metrics without and with VAE, was performed over Tables 2 and 4. The T-test revealed the statistical significance (p -value ≤ 0.05) for precision ($p=0.046$), recall ($p=0.020$), and NPV ($p=0.40$). Only the specificity shows a p -value > 0.05 with $p=0.068$.

Similar to the discrete epoch evaluation results, in the continuous epoch evaluation results, there is an improvement in performance when VAE is used. This improvement is shown as a decrease in confusion between classes. The average accuracy of our HAR system increases from 86.13% without VAE to 95.09% with VAE. The inference time for a single epoch on Android smartphone is 0.2 seconds. Thus, our Android Deep RNN could be implemented in a real-time mode.

Table 6 summarizes the previous HAR works and their performances including ours. Some of the previous works did not use AE in their HAR systems, such as [41] and [42]. The highest F1-score among the previous works is 93.9%, using Random Forest and hand-extracted features [42]. The highest F1-score using the DL algorithms is 93.7%, using a CNN based classifier [41]. Our approach achieved F1-score of 95.11% using VAE for denoising and Android Deep RNN for HAR. There are only few works that used AE in their HAR systems. For instance, in the work of Mohamed and Tashev [14], AE was used to denoise the IMU signals, then Deep Convolution LSTM (DCLSTM) was used for HAR, achieving an increment of 1% in F1-score. In our case,

F1-score is increased by 6% with VAE. In the work of Almaslukh et al. [43], the HAR classifier was based on a variant of AE called Stacked Autoencoder (SAE). Our VAE differs from SAE in terms of network structures, SAE is a feedforward neural network without convolutional or LSTM layers. Also, SAE was used only for feature extraction and classification similar to the neural networks in [41] and [42], but not for denoising the IMU signals. With SAE, an accuracy of 97.5% was reported instead of F1-score, which is comparable to our accuracy. Note that these summarized works do not represent the fair comparison with the same datasets and models, but the usefulness of AE is clearly demonstrated.

Table 6. Comparison with others HAR studies.

Model	HAR Works	Sensors	AE	Database	F1-Score (%)
KNN	Arif and Kattan [42]	DWS ^d	x	PAMAP	70.80
NN	Arif and Kattan [42]	DWS	x	PAMAP	89.60
DNN	Hammerla et al. [41]	MS ^e	x	PAMAP	90.40
LSTM-F	Hammerla et al. [41]	MS	x	PAMAP	92.90
CNN	Hammerla et al. [41]	MS	x	PAMAP	93.70
Rotation Forest	Arif and Kattan [42]	DWS	x	PAMAP	93.90
DCLSTM	Mohammed and Tashev [14]	MS	✓	Opportunity	90.81
SAE	Almaslukh et al. [43]	SWS ^f	✓	Smartphone-based HAR	97.50*
ADRNN ^g	Proposed HAR System	DWS	✓	PAMAP	95.11

^d Dominant Wrist Sensor (DWS)

^e Multiple Sensor (MS)

^f Single Waist Sensor (SWS)

^g Android Deep RNN (ADRNN)

* Accuracy

5. Conclusion

The reduction of noise in HAR signals improves the SNR as well as the HAR performance. Our VAE successfully denoises motion signals from an IMU at wrist by reducing the noise about 9 dB in SNR. In turn, the HAR performance improves around 6% in all metrics. Furthermore, lifelogging seems feasible on Android smart device, since our Android Deep RNN runs on Android smartphone in standalone mode and the inference time on a smartphone is less than the sensing time. Then, real-time HAR is possible. Finally, our HAR system could be used to develop mobile Apps to monitor in real-time daily or sports activities of daily living.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2019R1A2C1003713). Edwin Valarezo Añazco gratefully acknowledges to “Escuela Superior Politécnica del Litoral (ESPOL)” and its excellence program “Walter Valdano Raffo”.

References

1. P. Kasnesis, C. Patrikakis and I. Venieris, "Changing the Game of Mobile Data Analysis with Deep Learning," *IT Professional*, no. 99. (June, 2017).
2. A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga, "Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey," *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pp. 1-10. (2010)
3. M. Kepski and B. Kwolek, "Embedded system for fall detection using body-worn accelerometer and depth sensor," *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015*, vol. 2, pp. 755-759. (2015)
4. L. Montanini, A. Del Campo, D. Perla, S. Spinsante and E. Gambi, "A Footwear-Based Methodology for Fall Detection," *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1233-1242. (2018)
5. B.-S. Lin, P.-C. Hsiao, S.-Y. Yang, C.-S. Su and I.-J. Lee, "Data Glove System Embedded With Inertial Measurement Units for Hand Function Evaluation in Stroke Patients," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 2204-2213. (2017)
6. D. J. Walker, P. S. Heslop, C. J. Plummer, T. Essex and S. Chandler, "A continuous patient activity monitor: Validation and relation to disability," *Physiological Measurement*, vol. 18, pp. 49-59. (1997)
7. J. Margarito, R. Helaoui, A. M. Bianchi, F. Sartor and A. G. Bonomi, "User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach," *IEEE Transactions on Biomedical Engineering*, vol. 63, pp. 788-796. (2016)
8. A. Anand, M. Sharma, R. Srivastava, L. Kaligounder and D. Prakash, "Wearable Motion Sensor Based Analysis of Swing Sports," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 261-267. (2017)
9. H. Ghasemzadeh and R. Jafari, "Coordination Analysis of Human Movements With Body Sensor Networks: A Signal Processing Model to Evaluate Baseball Swings," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 603-610. (2011)
10. Z. Zhang, D. Xu, Z. Zhou, J. Mai, Z. He and Q. Wang, "IMU-Based Underwater Sensing System for Swimming Stroke Classification and Motion Analysis," *IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 268-272. (2017)
11. D. Ravi, B. Lo and G. Z. Yang, "Real-time food intake classification and energy expenditure estimation on a mobile device," *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015*. (2015)
12. N. R. Singh, "Implementation of Safety Alert System for Elderly People Using Multi-Sensors," *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, pp. 282-286. (2017)
13. T. Elakkiya, "Wearable Safety Wristband Device For Elderly Health Monitoring With Fall Detect And Heart Attack Alarm," *Third International Conference On Science Technology Engineering and Management (ICONSTEM)*, pp. 1018-1022. (2017)
14. S. Mohammed and I. Tashev, "Unsupervised Deep Representation Learning to Remove Motion Artifacts in Free-mode Body Sensor Networks," *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2017*, pp. 183-188. (2017)
15. Y. H. Lai, F. Chen, S. S. Wang, X. Lu, Y. Tsao and C. H. Lee, "A Deep Denoising Autoencoder Approach to Improving the Intelligibility of Vocoded Speech in Cochlear Implant Simulation," *IEEE Transactions on Bio-medical Engineering*, vol. 64, pp. 1568-1578. (2017)

16. D. T. Grozdic and S. T. Jovicic, "Whispered Speech Recognition Using Deep Denoising Autoencoder and Inverse Filtering," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, pp. 2313-2322. (2017)
17. J. Heymann, R. Haeb-Umbach, P. Golik and R. Schluter, "Unsupervised Adaptation of a Denoising Autoencoder by Bayesian Feature Enhancement for Reverberant ASR Under Mismatch Conditions," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 5053-5057. (2015)
18. P. Xiong, H. Wang, M. Liu, S. Zhou, Z. Hou and X. Liu, "ECG Signal Enhancement Based on Improved Denoising Auto-encoder," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 194-202. (2016)
19. N. D. Lane and P. Georgiev, "Can Deep Learning Revolutionize Mobile Sensing?," *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications - HotMobile '15*, pp. 117-122. (2015)
20. I. Roychowdhury, J. Saha and C. Chowdhury, "Detailed Activity Recognition with Smartphones," *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 1-4. (2018)
21. A. Vaughn, P. Biocco, Y. Liu and M. Anwar, "Activity Detection and Analysis Using Smartphone Sensors," *2018 IEEE International Conference on Information Reuse and Integration for Data Science*, pp. 102-107. (2018)
22. M. Jongprasithporn, N. Yodpijit and R. Srivilai, "A smartphone-based real-time simple activity recognition - IEEE Xplore Document," *3rd International Conference on Control, Automation and Robotics*, pp. 539-542. (2017)
23. J. J. Guiry, P. Ven, J. Nelson, L. Warmerdam and H. Riper, "Activity recognition with smartphone support," *Medical Engineering and Physics*, vol. 36, pp. 670-675. (2014)
24. E. Bulbul, A. Centin and I. A. Dogru, "Human Activity Recognition Using Smartphones," *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-6. (2018)
25. J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li and K. Shonali, "Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition," *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3995-4001. (2015)
26. E. Valarezo, P. Rivera, J. M. Park, G. Gi, T. Y. Kim and T. Kim, "Human Activity Recognition Using a Single Wrist IMU Sensor via Deep Learning Convolutional and Recurrent Neural Nets," *UNIKOM Journal of ICT, Design, Engineering and Technological Science*, vol. 1, pp. 1-5. (2017)
27. D. Tao, Y. Wen and R. Hong, "Multicolumn Bidirectional Long Short-Term Memory for Mobile Devices-Based Human Activity Recognition," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1124 - 1134. (2016)
28. F. J. Ordonez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors (Switzerland)*, vol. 16, no. 115. (July, 2016)
29. S. Basterrech and V. K. Ojha, "Temporal Learning Using Echo State Network for Human Activity Recognition," *2016 Third European Network Intelligence Conference (ENIC)*, pp. 217 - 223. (2016)
30. A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '12*, vol. 40, pp. 1-8. (June, 2012)
31. A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 108-109. (2012)
32. X. Zhang, H. Dou, T. Ju, J. Xu and S. Zhang, "Fusing Heterogeneous Features From Stacked Sparse Autoencoder for Histopathological Image Analysis," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*, vol. 20, no. 5, pp. 1377-1383. (2016)

33. H. Li and S. Misra, "Prediction of Subsurface NMR T2 Distributions in a Shale Petroleum System Using Variational Autoencoder-Based Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, pp. 2395-2397. (2017)
34. D. Park, Y. Hoshi and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1544-1551. (2017)
35. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735-1780. (1997)
36. M. Milenkoski, k. Trivodaliev, S. Kalajdziski, M. Jovanov and B. R. Stojkoska, "Real Time Human Activity Recognition on Smartphones using LSTM Networks," *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1126-1131. (2018)
37. Deeplearning4j Development Team, "Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation 2.0," <http://deeplearning4j.org>. (2017)
38. A. Khan, N. Hammerla, S. Mellor and T. Plotz, "Optimising sampling rates for accelerometer-based human activity recognition," *Pattern Recognition Letters*, vol. 73, pp. 33-40. (2016)
39. V. T. Hees, L. Gorzelniak, E. C. Dean Leon, M. Eder, M. Pias, S. Taherian, U. Ekelund, F. Renstrom, P. W. Franks, A. Horsch and S. Brage, "Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity," *PLoS ONE*, vol. 8, no. 4. (April, 2013)
40. C. Beleites, R. Salzer, and V. Sergo, "Validation of soft classification models using partial class memberships: An extended concept of sensitivity & co. applied to grading of astrocytoma tissues," *Chemometrics and Intelligent Laboratory Systems*, vol. 122, pp. 12 – 22. (2013)
41. N. Hammerla, S. Halloran and T. Plotz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables," *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 1533-1540. (2016)
42. M. Arif and A. Kattan, "Physical Activities Monitoring Using Wearable Acceleration Sensors Attached to the Body," *PLoS ONE*, vol. 10, no. 7. (2015)
43. B. Almaslukh, J. AlMuhtadi and A. Artoli, "An Effective Deep Autoencoder Approach for Online Smartphone-based Human Activity Recognition," *International Journal of Computer Science and Network Security*, vol. 17, no. 4, pp. 160-165. (2017)

Edwin Valarezo Añazco received his B.E. degree in Electronics and Telecommunication Engineering from the Escuela Superior Politécnica del Litoral (ESPOL), Ecuador. He is currently working toward his Ph.D. degree in the Department of Biomedical Engineering at the Bio-Imaging & Brain Engineering Laboratory at the Kyung Hee University, South Korea. His research interest includes Deep Learning, Human Activity Recognition (HAR), real-time applications for HAR, hand gesture recognition, computer vision, reinforcement learning, and autonomous control of anthropomorphic robotic hands.

Patricio Rivera Lopez received his B.E. degree in Electronics, Automation and Control Engineering from the University of the Armed-Forces-ESPE, Ecuador. He is currently working toward his Ph.D. degree in the Department of Biomedical Engineering at Kyung Hee University, Republic of Korea. His research interest includes artificial intelligence, signal processing, and machine learning.

Hyemin Park received her B.S. and M.S. degrees in Biomedical Engineering from Kyung Hee University, South Korea. Her research interests include machine learning such, deep learning, image processing, computer vision, Human Activity Recognition (HAR), and hand gesture recognition.

Nahyeon Park received her B.S. degree in Biomedical Engineering from Kyung Hee University, South Korea. She is currently working toward her M.S. degree in the Department of Biomedical Engineering at Kyung Hee University, Republic of Korea. Her research interests include machine learning such as deep learning, image processing, computer vision, and reinforcement learning with robotic arms.

Tae-Seong Kim received the B.S. degree in Biomedical Engineering from the University of Southern California (USC) in 1991, M.S. degrees in Biomedical and Electrical Engineering from USC in 1993 and 1998 respectively, and Ph.D. in Biomedical Engineering from USC in 1999. After his postdoctoral work in Cognitive Sciences at the University of California at Irvine in 2000, he joined the Alfred E. Mann Institute for Biomedical Engineering and Dept. of Biomedical Engineering at USC as Research Scientist and Research Assistant Professor. In 2004, he moved to Kyung Hee University in Korea where he is currently Professor in the Department of Biomedical Engineering. His research interests have spanned various areas of biomedical imaging, bioelectromagnetism, neural engineering, assistive biomedical lifecare technologies. Dr. Kim has been developing advanced signal and image processing methods, pattern classification, machine learning methods, novel medical imaging modalities, and rehabilitation technologies. Dr. Kim has published more than 300 papers and seven international book chapters. He holds ten international and domestic patents and has received nine best paper awards.

Received: September 20, 2019; Accepted: May 10, 2020

