An Effective Deep Autoencoder Approach for Online Smartphone-Based Human Activity Recognition

Bandar Almaslukh, Jalal AlMuhtadi, and Abdelmonim Artoli

Department of Computer Science, King Saud University, Riyadh, Saudi Arabia

Summary
Smartphones based human activity recognition (HAR) has a variety of applications such as healthcare, fitness tracking, etc. Nowadays, the signals generated by smartphone-embedded sensors such as accelerometer and gyroscope are used for HAR. However, achieving high recognition accuracy with low computation cost is required in smartphone based HAR. Therefore, we utilize one of the well-known deep learning approach named stacked autoencoder (SAE) to enhance the recognition accuracy and decrease recognition time. To evaluate the proposed method, we applied it on a public benchmark dataset; and compared it against available methods known of highest recognition accuracy on the same dataset. We have found that the new method increase the overall classification accuracy from 96.4% to 97.5% and as well the average recognition time of each testing sample is decreased from 0.2724ms to 0.0375ms.

Key words:
Deep Learning, Stacked Autoencoder, Human Activity Recognition, Machine Learning.

1. Introduction

Smartphone-based Human activity recognition (HAR) has aided new applications such as, healthcare, entertainment and safety. [1]. in the past, special wearable body motion sensors were used to recognize different human activities [1]. However, the availability of different sensors in smartphones such as accelerometer and gyroscope has enabled researchers to use these sensors for activity recognition as a result of their continuous enhancement in computational capabilities which made them cost-effective.

According to [2], the human activity recognition systems consist of four main steps as follow:

1) Sensing: here the sensors collect signal at a specific sampling rate.
2) Pre-processing: in this step, the collected signals is handled using different methods such as noise reduction and segmentation.
3) Feature Extraction: several data features are extracted from the segmented raw data.
4) Training or Classification: In training phase, that usually is conducted offline, the model is built and tuned with the optimal parameters. After constructing the optimized model, it become ready to use in classification phase. In this paper we mainly focus on training and classification phases.

In recent times, many training and classification methods for HAR have been applied on the smartphones. These methods were implemented using different machine learning approaches such as Naive Bayes [3], K-Nearest Neighbor (KNN) [3, 4 and 5], Decision Tree [6], Support Vector Machine (SVM) [7 and 8], Neural Network [9], Boosting algorithm [10] etc.

Deep Learning is one of the main classes of machine learning. Recently, deep learning methods have been gaining an intensive attention because they provide better performance in many fields such as image and speech recognition. However, in the past (before 2006) training supervised deep neural network with many layers (2 hidden layers or more) the weights are randomly initialized from a Gaussian distribution. After that, back-propagation optimization method is applied to the network in order to find the optimal parameters. Practically it was proved that this way of random initialization will lead to very slow optimization as well as it may stuck in poor local minima since the loss function is extremely uncontrolled when parameterized by millions of dependence variables. In response to these limitation Hinton et al. [11] in 2006 significantly reduce the learning problem by pre-train each layer of the network in unsupervised way to learn a discriminative representation of data before the classification task. In addition, in 2006 Hinton et al [2] proposed the first successful deep stacked autoencoder (SAE) to reduce the dimensionality of the data effectively. Therefore, stacked SAE is used in this paper but with a random weights initialization.

Currently, there are few works that utilized deep learning approaches for smartphone based HAR. For instance, works in [13, 14 and 15] utilized deep convolution neural network. SAE and denoising autoencoder (DAE) [16] were utilized in [17]. In addition, the study in [18] utilize Deep Belief Networks (DBN) [19]. The majority of these works apply the deep learning approach directly to the original signal (raw data). In contrast, our study apply SAE to the extracted feature from the original signal data.

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

We summarized the main contribution of this paper as follow:

- We propose a low cost SAE model for smartphone-based HAR.
- To get a high recognition accuracy, we optimize the parameters of the proposed method using the best practice techniques in the literature.
- We have conducted experimental analysis on public dataset and demonstrated that the proposed model is outperforms state-of-art studies on the same dataset.

The reminder of the paper is structured as follows: Section 2 we describe the dataset used in this paper. We explain the proposed method in section 3. Results and discussions are demonstrated in section 4. Finally, the conclusions and several promising venue for future works are stated in section 5.

2. Data Sets Description

We have used Smartphones based Human Activity Recognition data set created in [20]. The data set built with a group of 30 volunteers within an age between 19 and 48 years. Each person was wearing a smartphone (Samsung Galaxy S II) on the waist; and performed six different activities (walking, upstairs, downstairs, sitting, standing and lying). The generated dataset was divided into two parts: 70% of the volunteers were selected for the training set while the other 30% were used for the test set. Specifically , the training set contains 7352 instances while the test set contains 2947 instances , all instances are manually labeled with one of the six activities.

The dataset provides a large number of extracted features (561 feature) extracted by prepossessing the raw signals generated from the accelerometer and the gyroscope sensors. First, at a sampling rate of 50Hz the triaxial linear acceleration and angular velocity signals using the smartphone accelerometer and gyroscope are collected. After that, the median filter is applied to the collected signals to remove the noise. Then, the acceleration signals are separated by using Butterworth low-pass filter into body acceleration and gravity. Finally, the time signals sampled in fixed-width sliding windows of 2.56 sec and 50% overlap.

3. Methods

3.1 The Overall Procedure

The workflow that we follow in this paper is shown in Fig. 1. First, we implement the Stacked Autoencoder (SAE) classifier using Matlab, with the default setting. After that, the SAE model parameters were adjusted continuously until obtaining the best mixture of parameters that provide the highest accuracy on the test set. Then, we store the best-tuned model for further use. Finally, in terms of recognition accuracy we compare our model with state-of-the-arts works that had used the same dataset. However, in the following sections one of the deep learning models SAE classifier is explained in details.

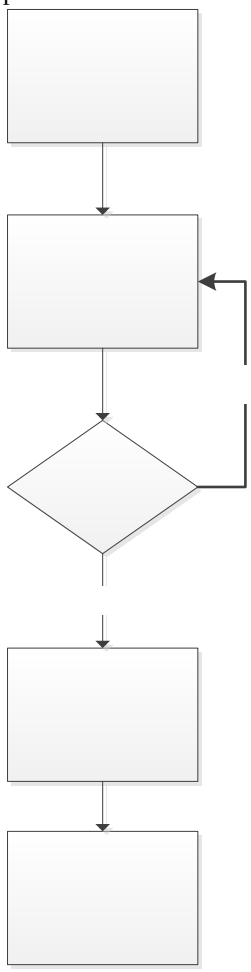

Fig. 1 The overall procedure of this paper.

3.2 Autoencoder (AE)

The basic unit of the SAE model is the AE. Architecturally, the AE (Fig 2) is a feed-forward neural network much similar to the multilayer perceptron (MLP). It has an input layer, one or more hidden layer(s) and an output layer. Exceptionally, for AE the number of neurons in the output and input layers are equal. In addition, it considered as an unsupervised learning since it learn to reconstruct its input instead of predicting the target value.

AE consists of two phases: the encoder phase (from input layer to hidden layer) and the decoder phase (from hidden layer to output layer). The encoder phase can be formulated as in Eq. (1), where W is the weight matrix and b_1 is bias

vector for the encoder phase. In Eq. (2), the decoder phase is formulated, where W^T is the weight matrix and b_2 is bias vector for the decoder phase. The f in Eq. (1) and (2) refers to one of the well-known nonlinear activation functions, in this work the sigmoid function is used. In the following, we describe x, h and $\tilde{x}$ in Fig. 2 as:

- x is the input layer.
- h is the latent representation (code) of the input layer x.
- $\tilde{x}$ is $p(x|h)$ which approximately should have the same shape of x.

$$h = f(Wx + b_1) \quad (1)$$
$$\tilde{x} = f(W^T h + b_2) \quad (2)$$

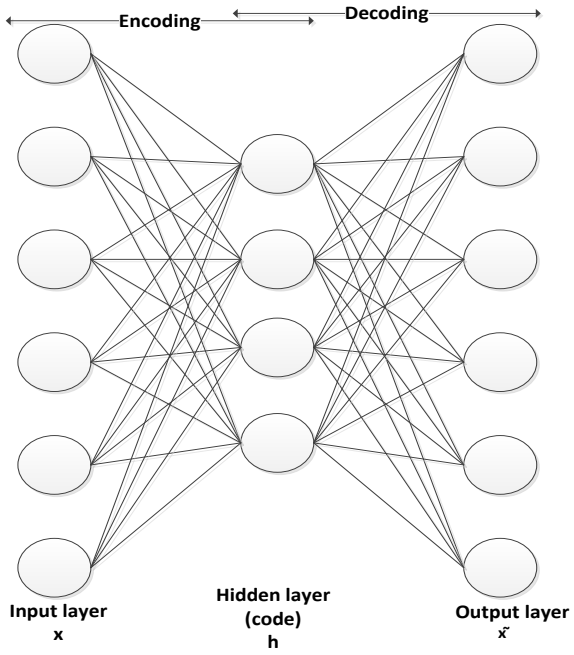

Fig. 2 The basic architecture of the AE.

3.3 Proposed SAE Classifier (AE)

The SAE architecture used in this work is shown in Fig. 3. It is formed by stacking two AE (AE1 and AE2) on top of each other then adding softmax layer on top of the AEs. However, there are two stages to train the SAE. The first stage is unsupervised pre-training where a greedy layer wise method is used to pre-training a deep network one layer at a time. At each layer the error in reconstructing of its input using unlabeled samples is minimized. Feeding the latent representation (code) of the AE1 as an input for the AE2, where the input of the AE1 is the original data features. After the AE1 and AE2 are trained, the second stage called supervised fine-tuning is started. It called supervised since we want to minimize prediction error using labeled samples. To minimize the prediction error we added a softmax layer on top of the network then train the whole network using the back-propagation as a regular MLP.

In details, the proposed SAE classifier operate as follows:

1) AE1 is trained on the input data (561 features) to learn the compressed code (80 features) called Code1.
2) AE2 is trained using the Code1 as an input to learn extremely compressed code (5 features) called Code2.
3) Construct the whole network that is called SAE by stacking AE1, AE2 and Softmax classifier; then the whole network is fine-tuned using back-propagation algorithm.

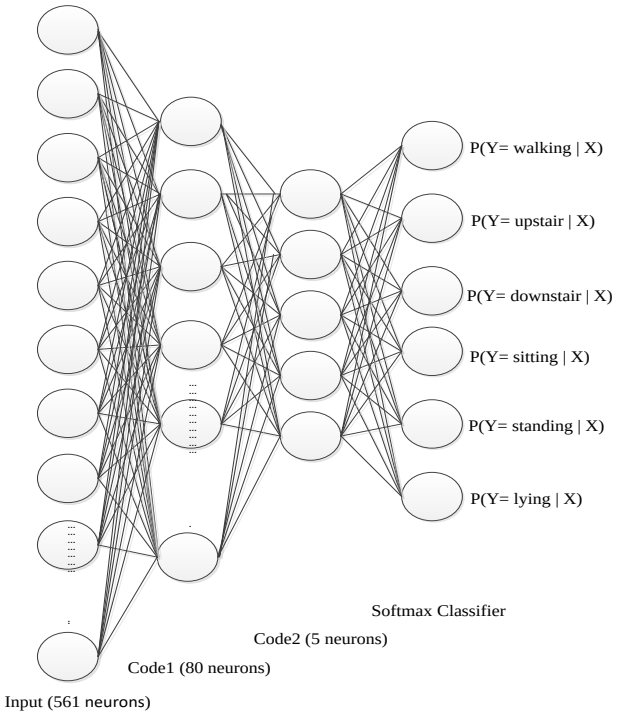

Fig. 3 The SAE classifier architecture use in this study.

In deep learning models such as SAE, it is non-trivial to optimize the model parameters that provide high recognition accuracy. However, in this study, we assess the accuracy of the model after each run to check if the model over-fits or under-fits; then we take the right action that could reduce the problem. There are many strategies to improve deep learning performance as mentioned in [21 and 22]. Actually, there is no rule of thumb to find the optimal parameters, but there are some of good practical strategies. For example, to choose the best network topology, first use one hidden layer with many neurons, if not work use more than one hidden layer (deep) with few neurons in each layer, or finally by combining the two techniques (more than one hidden layer with many neurons in each layer). However, the best model parameters used in this paper is provided in

Table.1, where in section 4.1 we show how some of the proposed SAE model parameters were optimized.

4. Results and Discussions

In this section, we demonstrate the experimental results of the proposed SAE model. Matlab R2016b installed on conventional computer with a 2.4GHZ CPU and 16GB memory is used to conduct the experiments. SAE classifier is trained using 7352 training instances (using dataset in section 2). Some of the classifier parameters are adjusted until we get the maximum possible accuracy using 2947 test instances. The impact of main parameters on the recognition accuracy of the proposed SAE is discussed in section 4.1.
It is important to mention that a competition aiming to develop novel classification approaches (for the dataset used in this study) was organized in European Symposium on Artificial Neural Networks (ESANN) in 2013. However, in [23] the dataset originator mentioned that the competition winner work has achieved the best overall accuracy, with 96.4 [7]. Therefore, (in section 4.2) we will compare our work with the competition winner in term of recognition accuracy and the computational cost. Moreover, we will compare the accuracy of our work with the state-of-art studies that utilize conventional machine learning or deep learning approaches on the same dataset.

4.1 Model Parameters Optimization

The recognition accuracy of SAE classifier at unsupervised pre-training and supervised fine-tuning stages is affected by many parameters. The main parameters of SAE classifier are the number of layers, the number of neurons in each layer, max epoch, learning rate batch size, etc. In this study, primarily we fine-tune three important parameters which are number of hidden layers, number of neurons in each layer and max epoch. According to Matlab documentation, other parameters are set to the default value. By adjusting the number of hidden layers (1, 2, 3 and 4 layers), the results show the best recognition accuracy when the number of hidden layers is two.
To choose the optimal number of neurons in each layer, we have obtained the highest possible accuracy using one hidden layer with 150 neuroses. Then, we added the second hidden layer with few neurons while reducing the number of neurons in first hidden layer. After that, we frequently adjusted the number of neurons in each layer until we get the highest classification accuracy with (80 neurons) in the first hidden layer and (5 neurons) in the second hidden layer. However, for the max epoch parameter, we initially used the default value in Matlab then we increase it by 20 until we get the best accuracy (for more details about the SAE parameter see Table 1.

Table 1: The main parameters of the proposed method

Network Model	***Parameters***
AE1	hiddenSize = 80 Encoder and Decoder Transfer Function = Logistic sigmoid function MaxEpochs=500 L2WeightRegularization= 0.004 Loss Function= Mean squared error function Training Algorithm= Conjugate gradient descent
AE2	hiddenSize = 5 Encoder and Decoder Transfer Function = Logistic sigmoid function MaxEpochs=400 L2WeightRegularization= 0.002 Loss Function= Mean squared error function Training Algorithm= Conjugate gradient descent
SAE	Number of layers = 4 (561-80-5-6 neurons in each layer) MaxEpochs=1300 L2WeightRegularization= 0.01 Training Algorithm= Conjugate gradient descent

4.2 Comparison with the Competition Winner

In this section, we compare the proposed method with the best recognition accuracy archived on this benchmark datasets [20]. According to [23], the competition winner used the One-Vs-One Multiclass linear SVM with majority voting and get the highest overall recognition accuracy with 96.4%. The competition winner and our method are compared in terms of overall recognition accuracy and average recognition time. However, we have found that the overall accuracy of the proposed method outperforms the competition winner method by 1.1%, as shown in Table 4. From the confusion matrix of the competition winner method (Table 3) and the proposed method (Table 2), it is obvious that the recognition accuracy of each corresponding activity is not the same. Significantly, the recognition accuracy of standing activity in the competition winner method is better than our method by 5.81%. In contrast, for sitting activity our method outperform the classification accuracy of the competition winner by 10.71%.

Table 2: Confusion matrix for SAE classifier (proposed method)

Activity	Wa	Up	Do	Si	St	Ly	Accuracy %
Walking	**491**	8	3	0	0	0	97.8
Upstairs	2	**458**	6	3	0	0	97.7
Downstairs	3	5	**411**	0	0	0	98.1
Sitting	0	0	0	**455**	5	0	98.9
Standing	0	0	0	43	**527**	0	92.5
Lying	0	0	0	0	0	**537**	100
Precision %	99.0	97.2	97.9	90.6	99.1	100	**97.5**

Table 3: Confusion matrix of The competition winner method [7]

Activity	Wa	Up	Do	Si	St	Ly	Accuracy %
Walking	**493**	0	0	3	0	0	99.4
Upstairs	28	**430**	0	0	0	0	94.06
Downstairs	2	6	**412**	0	0	0	98.1
Sitting	0	2	0	**433**	56	0	88.19
Standing	0	0	0	9	**523**	0	98.31
Lying	0	0	0	0	0	**537**	100
Precision%	94.3	98.2	99.3	98	90.3	100	**96.4**

Table 4: Performance Comparison with the competition winner method

Method	Overall Accuracy %	Average Recognition Time (ms)
The competition winner [7]	96.4	0.2724
Our work	97.5	0.0375

On the other hand the training time of the proposed method is very small (less than 9 minutes) using conventional computer with a 2.4GHZ CPU and 16GB memory. Since the training phase is done offline on conventional computer, it will not be compared with other methods. However, recently for online HAR the classification phase is run using limited-resources smartphones, so it is very important to compare average classification time with other methods. As shown in Table 4, the average recognition time of our method is decreased from 0.2724ms to 0.0375ms compared to the competition winner method.

Finally, the overall recognition accuracy of the proposed method is compared with some of the related works as shown in Table 5. Making the comparison meaningful, we compared the proposed method with state-of-art studies that use the same dataset in this paper. From Table 5. , it is clear that our method outperforms all the methods in term of overall recognition accuracy. Even though some of the related works utilize deep learning approachs as we do in this paper, the proposed method still outperforms those studies.

Table 5: Comparison of our method with some state-of-art studies.

Machine Learning Approach	***Method***	***Overall Accuracy %***
Deep Learning	Deep Convolutional Neural Network [14]	95.75
	Stacked Autoencoder [17]	92.16
	Denoising Autoencoder [17]	90.50
Shallow Learning	Two-Stage Continuous Hidden Markov Models [24]	91.76
	Confidence-based boosting algorithm Conf-AdaBoost.M1 [10]	94.33
	A sparse kernelized matrix learning vector quantization model [25]	96.23
	OVO Multiclass linear SVM with majority voting [7]	96.4
	Our method	**97.5**

5. Conclusions and future works

In this paper, we proposed a smartphone-based HAR system that utilizes one of the well-known deep learning approaches called stacked autoencoder (SAE). Experimentally, the proposed method enhances the recognition accuracy by 1.1%. In addition, it reduce the average recognition time of each test sample from 0.2724ms to 0.0375ms compared to the competition winner method. However, making the comparison more realistic, we have compared the proposed method with other state-of-art methods that use the same dataset in this paper. It is shown that the proposed method outperforms all of these work in term of overall recognition accuracy. Since it is not difficult to collect unlabeled data, in the future work we will use more data in pre-training phase in hope of enhancing the accuracy of the proposed method. In this paper, some of the model parameters are not optimized so we will optimize these parameters in future work. However, the accuracy might enhance using different deep learning approaches such as Deep Convolutional Neural Network.

Acknowledgments

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

[1] O. D. Lara, and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Communications Surveys and Tutorials, vol.15, no.3, pp.1192-1209, 2013.

[2] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” ACM Computing Surveys (CSUR), vol.46, no.3, 2014.

[3] M. Kose, O. D. Incel, and C. Ersoy, “Online human activity recognition on smart phones,” in Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data, vol.16, no.2012, pp.11-15, 2012.

[4] S. Das, L. Green, B. Perez, M. Murphy, and A. Perring, “Detecting user activities using the accelerometer on android smartphones,” TRUST REU the Team for Research in Ubiquitous Secure Technology, no.29, 2010.

[5] S. Thiemjarus, A. Henpraserttae, and S. Marukatat, “A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone,” in Body Sensor Networks (BSN), 2013 IEEE International Conference on, pp.1-6. IEEE, 2013.

[6] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, “Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach,” in Wearable Computers (ISWC), 2012 16th International Symposium on, pp.17-24. IEEE, 2012.

[7] B. Romera-Paredes, M. S. Aung, and N. Bianchi-Berthouze, “A one-vs-one classifier ensemble with majority voting for activity recognition,” in ESANN 2013 proceedings, 21st European Symposium on Artificial Neural Networks,

Computational Intelligence and Machine Learning, pp.443-448, 2013.
[8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Luis Reyes-Ortiz, "Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic," J. UCS vol.19, no.9, pp.1295-1314, 2013.
[9] A. M. Khan, M. H. Siddiqi, and S. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones,"Sensors vol.13, no.10, pp.13099-13122, 2013.
[10] A. Reiss, G. Hendeby, and D. Stricker, "A competitive approach for human activity recognition on smartphones," in European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), 24-26 April, Bruges, Belgium, pp.455-460. ESANN, 2013.
[11] G. E. Hinton, S. Osindero, and Y. The, "A fast learning algorithm for deep belief nets," Neural computation vol.18, no.7, pp.1527-1554, 2006.
[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science vol.313, no.5786, pp.504-507, 2006.
[13] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, pp.197-205. IEEE, 2014.
[14] C. A. Ronao, and S. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in International Conference on Neural Information Processing, pp.46-53, Springer International Publishing, 2015.
[15] D. Ravi, C. Wong, B. Lo, and G. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on, pp.71-76. IEEE, 2016.
[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research vol.11, no. Dec (2010), pp.3371-3408, 2010.
[17] Y. Li, D. Shi, B. Ding, and D. Liu, "Unsupervised feature learning for human activity recognition using smartphone sensors," in Mining Intelligence and Knowledge Exploration, pp.99-107, Springer International Publishing, 2014.
[18] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol.22, no.1, pp.1729. 2011.
[19] H. E. Geoffrey, "Deep belief networks," Scholarpedia, vol.4, no.5, 2009.
[20] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," in ESANN. 2013.
[21] B. D. Ripley, "Pattern recognition and neural networks," Cambridge university press, 2007.
[22] M. J. A. Berry, and G. Linoff, "Data Mining Techniques: for Marking," Sales, and Customer Support, New York: John Wiley&Sons Inc, 1997.
[23] J. L. R. Ortiz, "Smartphone-based human activity recognition," Springer, 2015.
[24] C. A. Ronao, and S. Cho, "Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models," in Natural Computation (ICNC), 2014 10th International Conference on, pp.681-686. IEEE, 2014.
[25] M. Kästner, M. Strickert, T. Villmann, and S. Mittweida, "A sparse kernelized matrix learning vector quantization model for human activity recognition," in ESANN. 2013.

Bandar Almaslukh is a PhD candidate in Computer Science at KSU University, Saudi Arabia. His on-going research of interest in human activity recognition using deep learning approaches. He holds a master degree in computer science from KSU University, Saudi Arabia. He is working as Lecturer at computer science department, Prince Sattam bin Abdul Aziz University, Saudi Arabia. He has taught many courses such as programming language and data structure. He worked as Database Developer in Arriyadh Development Authority in Saudi Arabia for many years.

Jalal F. Al-Muhtadi is the Director of the Center of Excellence in Information Assurance (CoEIA) at King Saud University. He is also an Assistant Professor at the department of Computer Science at King Saud University. Areas of expertise include cybersecurity, information assurance, privacy, and Internet of Things. He received his PhD and MS degrees in Computer Science from the University of Illinois at Urbana-Champaign, USA. He has over 50 scientific publications in the areas of cybersecurity and the Internet of Things.

A.M. Artoli is a leading computational scientist in the fields of biocomputing, lattice Boltzmann method and non-newtonian fluid flow. His diverse research interest includes blood flow at microscale, density matrix renormalization, and complex system dynamics. Artoli has published many articles in the above disciplines which are heavily cited. His Ph.D. thesis on "mesoscopic computational haemodynamics" has been cited more than 100 times. Artoli have worked at the Informatics Institute, University van Amsterdam and In IST, Portugal and in Sudan as a dean of the Graduate College, Alneelain University. He has received the Sudanese Ministry of High Education Award as the best scientist. He is also a member of a number of Scientific societies and an organizer of a conference sereies, editor in chief and regular reviewer of a number of journals. Artoli have supervised 10s of M.Sc. students and a few Ph.D. students in Europe, Sudan and Saudi Arabia.