International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 4, August 2021, pp. 3551~3558
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i4.pp3551-3558

Time series activity classification using gated recurrent units

Yi-Fei Tan, Xiaoning Guo, Soon-Chang Poh
Faculty of Engineering, Multimedia University, Malaysia

Article Info

Article history:

Received Jun 2, 2020
Revised Dec 11, 2020
Accepted Dec 21, 2020

Keywords:

Activity classification
AReM dataset
Gated recurrent units
Recurrent neural network
Time series

ABSTRACT

The population of elderly is growing and is projected to outnumber the youth in the future. Many researches on elderly assisted living technology were carried out. One of the focus areas is activity monitoring of the elderly. AReM dataset is a time series activity recognition dataset for seven different types of activities, which are bending 1, bending 2, cycling, lying, sitting, standing and walking. In the original paper, the author used a many-to-many recurrent neural network for activity recognition. Here, we introduced a time series classification method where Gated Recurrent Units with many-to-one architecture were used for activity classification. The experimental results obtained showed an excellent accuracy of 97.14%.

Corresponding Author:

Tan Yi-Fei
Faculty of Engineering
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia
Email: yftan@mmu.edu.my

1. INTRODUCTION

The elderly population is on the rise globally. According to a report by the United Nations, it was projected that the elderly population will skyrocket and reach 2.1 billion by 2050 [1]. The population of elderly in Malaysia is expected to be about 20% of its total population by 2040 [2]. Due to these projections, many researches have focused on improving elderly assisted living to ensure the health and well-being of the elderly. One of the concerns is that there is no one to take care of elderly living in solitude. Hiring caregivers to take care of elderly is one of the solutions but it might be not affordable for some family. One of the research areas is on remote monitoring of elderly such as vital signs and activities using sensors which include activity recognition. There are currently two main types of activity recognition namely vision-based activity recognition and non-vision-based activity recognition.

For vision-based activity recognition, the data is video feed of people carrying out certain activity captured by camera [3-7]. The deep learning model that is commonly used in vision-based activity classification is a combination of convolutional neural network (CNN) and recurrent neural network (RNN) [5, 6]. CNN is a type of neural network that is used for image-related tasks such as image classification [8-11] and object detection [12-13]. CNN is also used to extract features from the speech signals. Alternatively, RNN is a type of neural network used for sequence-related tasks [14-18]. In a typical video, each of the frame is basically a sequence of fixed-size images. CNN encodes each frame of the video into a vector, which results in a sequence of vectors for the video. RNN consumes this sequence of vector and outputs a prediction. The computational cost and accuracy of this method increases with the increases of the frame rate of the video and the size of the frame. Vision-based method may raise privacy concerns as the elderly will be recorded at all times.

As for non-vision-based method, it involves using sensor data as input. There are different types of sensors that may be used here such as accelerometer [19], an accelerometer is a sensor which measures the acceleration of a moving object. It is miniscule and can fit into a smart phone. Anguita and colleagues used smart phone which has a triaxial accelerometer for data collection [19]. Triaxial accelerometer measures acceleration in three axes. When a person carries out an activity, the accelerometer measures acceleration in all three axes. Different activities can result in different patterns of acceleration. A machine learning algorithm called support vector machine (SVM) is used here for activity classification.

One-dimensional CNN can be used to extract important feature from time series sequences such as sound signals [20-22]. Lee and researchers [23] used one-dimensional CNN for time series to perform activity recognition on accelerometer data achieving a high accuracy of 92.71%. In [24], Xu and colleagues compared CNN with SVM on accelerometer data and found that accuracy achieved by CNN is higher than SVM. The proposed approach achieved high accuracy and has low computational cost.

Palumbo *et al.* [25] proposed a novel non-vision-based activity recognition method which involves using wireless beacons that implementing IEEE 802.15.4. In their research, three of the wireless beacons were attached to human subject's chest and both ankles. Wireless beacons are passive like RFID cards, without the need to use battery to power the beacons. The received signal strength (RSS) of these beacons were read by a scanner for data collection. The collected data is then stored and named as "activity recognition system based on multisensor data fusion," (AReM) dataset. The author's idea was to observe the alteration of RSS of three beacons when a person is carrying different activities. These RSS may provide sufficient distinction such that the activities can be classified using machine learning algorithm, namely many-to-many RNN method. Many-to-many RNN model takes in time series sequences of the RSS values and classify the values into categories of activities at each time step. This method managed to achieve an overall accuracy of 92.30%.

In this paper, we proposed gated recurrent units (GRUs) with a many-to-one architecture to classify the activities using the AReM dataset. The remaining parts of this paper is organized as follows: Section 2 describes the method, section 3 contains the experimental results and discussions. Finally, a concluding remark is presented in section 4.

2. RESEARCH METHOD

Figure 1 depicts the process of method which includes data collection, data processing, data partitioning, modelling and model selection. The following subsections describes each of the steps: data collection, data processing, data partitioning, modelling and model selection.

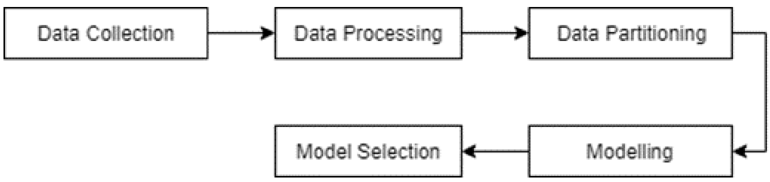

Figure 1. Flow of method

2.1. Data collection

The dataset used in this work is a publicly available activity recognition dataset called AReM dataset [25]. During data collection, three wireless beacons which implements IEEE 802.15.4 standard are attached to human subject's chest and both ankles. The human subjects were requested to carry out seven different activities which include cycling, lying down, sitting, standing and walking as well as two types of bending labelled as bending 1 and bending 2. The percentage of each activity type in the dataset is listed in Table 1. Bending 1 and Bending 2 has smaller percentages of 7.865% each, compared to the other activities of 16.85% each from the dataset.

Table 1. Percentage of each activity type

Activity Type	Percentage (%)
Bending 1	7.865
Bending 2	7.865
Cycling	16.85
Lying	16.85
Sitting	16.85
Standing	16.85
Walking	16.85

Wireless beacons emit electromagnetic wave which will attenuate over distance. The signal strength of the wireless beacon uses a measurement unit called received signal strength (RSS) indicator. Wireless scanners can scan the RSS of the wireless beacons. The wireless beacons' RSS values were sampled at a frequency of 20 Hz. For every 50 milliseconds, the scanner will obtain an RSS value for each of the three sensors. The author of the dataset computed the mean and variance for RSS value accumulated every 250 milliseconds for each beacon. Therefore, each data point is a vector which has six dimensions. It consists of six features which include the mean and variance for each of the three beacons calculated over five consecutive RSS values. The features are labelled as given in Table 2. The data were recorded sequentially in time. Table 3 shows a snippet of dataset for Bending 1. Every 250 milliseconds, there is a data point which consists of six values.

Table 2. Features of dataset

	Mean	Variance
Beacon 1	*avg_rss12*	*var_rss12*
Beacon 2	*avg_rss13*	*var_rss13*
Beacon 3	*avg_rss23*	*var_rss23*

Table 3. Snippet of bending 1 dataset

Time interval (ms)	*avg_rss12*	*var_rss12*	*avg_rss13*	*var_rss13*	*avg_rss23*	*var_rss23*
0- 250	43.67	0.47	24.75	0.43	30	0
250-500	43.33	0.47	25.33	0.47	30	0
500-750	42.75	0.83	25.25	0.83	30.5	0.5

For machine learning algorithm to classify data points of different activities, the patterns of each type of activity has to be distinct. Means and variances of RSS values are valid features for activity classification. Figure 2 shows a plot of the means and variances of RSS of Bending 1 activity. Figure 3 shows a plot of means and variances of RSS of sitting activity. Each feature behaves differently over time for Bending 1 and sitting. This enables the machine learning algorithm to learn to classify different types of activities.

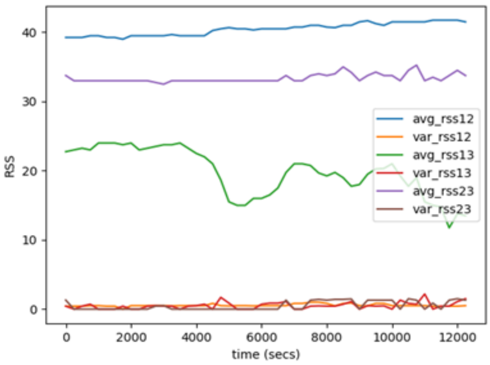

Figure 2. Plot of means and variances of RSS of bending 1 activity

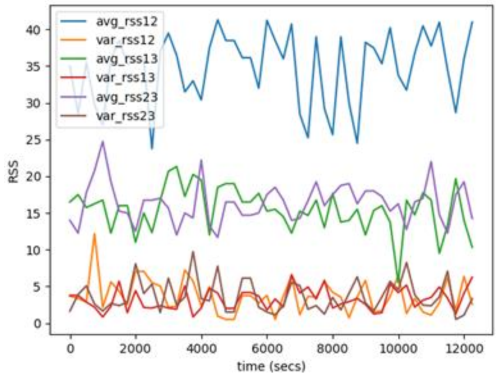

Figure 3. Plot of means and variances of RSS of sitting activity

2.2. Data processing

We proposed segmenting the sequence into smaller segments before passing them to RNN. We have shown that segmenting sequences before passing them to RNN resulted in excellent results in terms of accuracy on anomaly detection task [26]. In the data processing step, we segment the sequence of each activity type by using sliding window. The size of the sliding window is denoted by *winsize*. Sliding window with different window sizes which include 4, 8, 16 and step size of one is used to segment the sequence in dataset. Segmenting the longer sequence into multiple shorter sequences has two advantages. Firstly, it simplifies the complexity of the architecture of the model to be used because the problem becomes less complex. Instead having to take long and variable length sequence as input, the model takes in fixed size segment with shorter lengths. Secondly, segmenting increases number of data points. Because sliding window is used, the segmented data overlaps which means more features may be captured by the model. The window sizes 4, 8 and 16 correspond to 1 seconds, 2 seconds and 4 seconds time interval as given in Table 4.

Table 4. Window size and time interval

Window size	Time Interval (Second)
4	1
8	2
16	4

Figure 4 illustrates an example of segmenting a sequence of 5 data points into two segments using sliding window with window size of 4 and step size of one. In the segmentation process, the Segment 1 consists of data point 1 to data point 4 as window size is 4. Since the step size is set to one, the window slides one unit to the right and group four data points which include data point 2 to data point 5 as Segment 2. Since each data point is a six-dimensional vector, each segment is a matrix with the size of (*winsize* , 6).

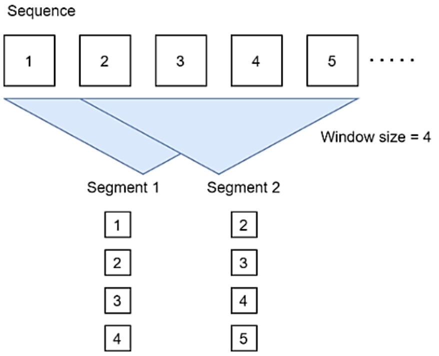

Figure 4. An illustration of data processing step with window size of 4 and step size of 1

2.3. Data partitioning

During data partitioning step, the dataset is partitioned into training and test set. The training set is dataset used to train the activity classifier. Whereas, the test set is an unseen pool of data used to evaluate the performance of the trained classifier. For this work, we divided the dataset into training set and test set with the ratio of 80% to 20%.

2.4. Modelling

Recurrent neural network (RNN) is one of the techniques that suitable to classify the time series and sequential data. There are four main architectures of RNN as illustrated in Figure 5 [27]. In Figure 5, blue blocks are outputs, red blocks are input and green blocks are the RNN units. RNN is suitable for classifying sequential data because of it uses information from previous time step in computation of current time step. In [25], the authors used a many-to-many RNN architecture illustrated in the rightmost of Figure 5 for activity classification. For many-to-many architecture, there is classification at every time step of the input sequence.

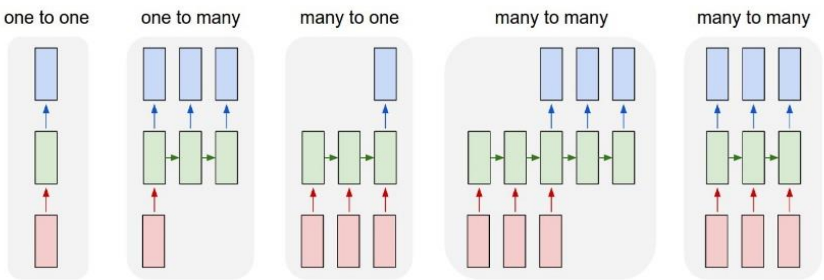

Figure 5. Different RNN architecture

In this paper, a variant of RNN called gated recurrent units (GRU) is used for activity classification [28]. The RNN architecture used is many-to-one architecture. For this architecture, the model predicts at the final time step. In this paper, many-to-one architecture was selected over many-to-many architecture. Firstly, many-to-one architecture consists of less computation when compared to many-to-many architecture because it only predicts or classifies at the final time step. Hence, the model is faster when predicting. The time step is denoted by t. As shown in Figure 6 [29], a GRU takes in the input vector for current time step denoted by x_t and the output vector from previous time step $t-1$ denoted by $h_t - 1$, and outputs an output vector for current time step t denoted by h_t. Using information from previous computation is a common feature of RNN as GRU uses $h_t - 1$ in computation for current time step. The transition functions in GRU are given as:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_\gamma \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

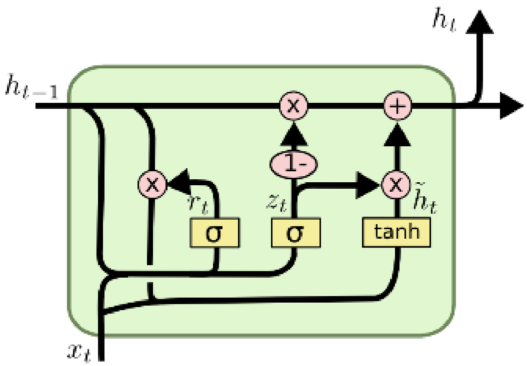

Figure 6. A gated recurrent unit

GRU consists of two gates namely update gate and reset gate. The update gate vector is denoted by z_t and the reset gate vector is denoted by r_t. z_t acts as a “gate” which decides how much of $h_t - 1$ is used in computation of output at current time step, h_t as given in the equations above. The W_Z, W_Y *and* W, are the parameters of the GRU. Whereas, σ is a non-linear function such as sigmoid function. In [30], Chung *et al.* compared the two popular variants of RNN, LSTM and GRU, it was found that GRU produces highly comparable or sometimes even better results as LSTM. Furthermore, GRU uses less training parameters, leads to less memory used and faster training and execution.

In this paper, we proposed using many-to-one architecture for GRU as shown in Figure 5. The model architecture used is many-to-one which means multiple inputs at multiple time steps and a single output at the last time step. The GRU model for data segmented using *winsize* = 4 is illustrated in Figure 7. At each time step, a six-dimensional data point of a data segment is consumed as an input. The number of time steps depends on the window size used to segment the sequence of data points. At the final time step, the model spits out a vector consists of seven dimensions, each dimension is for each of the seven class labels. The vector is then passed through the softmax layer which generates a seven-dimensional output vector. Each component of this vector is the conditional probability of the input being a type of activity given the input. The loss function used to train this model is categorical cross entropy.

In this work, the algorithm was implemented using Python programming language and Keras library. The optimization algorithm used for gradient descent is the Adam optimization algorithm [31]. The default parameters for Keras Adam optimizer was used. The models for varying window size were trained for 50 epochs. The number of hidden units of GRU is a hyperparameter. It’s a convention to use 2^n (where *n* is a positive integer) as choices of hyperparameters for neural networks. In this work, we tried several choices of number of hidden units which include 16, 32 and 64.

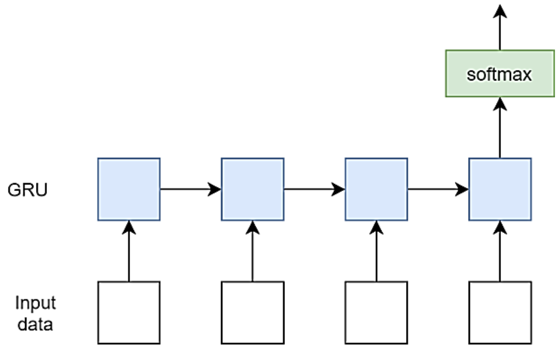

Figure 7. Many-to-one GRUs with window size of 4 for activity classification

2.5. Model selection

In this work, there are two hyperparameters namely window size and number of units of GRU. But the most suitable value for these hyperparameters are unknown. The standard way to decide these values is to sample a few choices, train a model for each of these choices and select the choice which lead to the best performing model. The list of choices of hyperparameters is provided by Table 5.

Table 5. Hyperparameters

Hyperparameters	Choices
Window size	4, 8, 16
Number of units of GRU	16, 32, 64

To evaluate the model, an evaluation metric is needed. In this case, we used accuracy on the test set as the evaluation metric. Accuracy is the percentage of correct classification over all samples to be classified. We select the model with the highest accuracy as the best model.

3. RESULTS AND DISCUSSION

Table 6 listed the train and test accuracies for different models trained using varying number of units of GRU and data segmented by different window sizes. It can be observed that the accuracy improves as the numbers of units of GRU increase. This trend is consistent for different choice of window size. One explanation is due to increasing complexity. This is because the weights of the model increase as number of units increase. With more weights, GRU can approximate a more complex function. A more complex function can model the pattern of the means and variances of RSS of different activities with better accuracy.

Table 6. Hyperparameters

Number of units of GRU	*winsize* =4		*winsize* = 8		*winsize* = 16	
	Train accuracy	Test accuracy	Train accuracy	Test accuracy	Train accuracy	Test accuracy
8	81.79	80.80	85.43	84.90	89.05	89.05
16	85.71	85.00	90.18	88.70	93.98	94.71
32	88.53	86.37	93.14	91.44	96.49	97.14

It was also found that the larger the window size, the higher the accuracy. Based on observation, this trend occurs for all choices of number of units of GRU. This is due to window size corresponds to the time interval. As the window size increases, each data segment will consist of more data points. Thus, this enables the model to easily distinguish the sequence pattern of RSS for different activity type.

Overall, the best model used 32 hidden units of GRU and *winsize*=32 and has a test accuracy of 97.14% as shown in Table 6. In addition, the difference in train and test accuracies of the best model is only 0.65%. Small difference between test accuracy and train accuracy means that the model have not been overfitted to the data in the training set. This assures consistent performance on unseen pool of data in the test set. In [25], the authors reported an accuracy of 92.30% using a many-to-many architecture. This is not a fair comparison since they were trained and evaluated on different splits of dataset. However, it can be concluded that the many-to-one method has an accuracy which is at least comparable to that of many-to-many architecture. The advantage of many-to-one architecture and using segmentation is that there is less computation task compared to many-to-many architecture. Since the computation task is less, it can be trained and make prediction more quickly.

4. CONCLUSION

In this paper, an activity classification algorithm using many-to-one GRU network is introduced. An experiment was carried out using AReM dataset and the results demonstrated that the proposed many-to-one GRU model performs excellently in terms of test accuracy. The accuracy of the best model has an accuracy of 97.14%, which is at least comparable than the accuracy achieved in the original paper. The advantage that it has over many-to-many architecture described in the literature is that it has shorter inference time. Currently, the model is applied on a publicly available dataset. For future work, we would like to apply the model on other datasets and also create our own device and full system for activity recognition.

REFERENCES

[1] World Health Organization, “Health situation and trend assessment,” 2018. [Online]. Available: http://www.searo.who.int/entity/health_situation_trends/data/chi/elderly-population/en/
[2] M. U. Mahidin, “Selected demographic indicators,” Department of Statistics Malaysia, 2018.
[3] A. Bux, P. Angelov and Z. Habib, “Vision based human activity recognition: a review,” *Advances in Computational Intelligence System*s, vol. 513, pp. 341-371, 2016.
[4] I. Rodríguez-Moreno, J. M. Martinez-Otzeta, B. Sierra, I. Rodriguez Rodriguez and E. Jauregi Iztueta, “Video Activity Recognition: State-of-the-Art,” *Sensors*, vol. 19, pp. 3160-3184, 2019.
[5] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," *IEEE Access*, vol. 6, pp. 1155-1166, 2018.
[6] X. Wang, L. Gao, P. Wang, X. Sun and X. Liu, "Two-Stream 3-D convNet Fusion for Action Recognition in Videos with Arbitrary Size and Length," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 634-644, 2018.
[7] K. Rangasamy, M. A. As’Ari, N. A. Rahmad, N. F. Ghazali, and S. Ismail, “Deep learning in sport video analysis: a review,” *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 18, no. 4, pp. 1926-1933, 2020.
[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Las Vegas, NV, USA, 2016, pp. 770-778.
[9] F. Sultana, A. Sufian and P. Dutta, "Advancements in Image Classification using Convolutional Neural Network," *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN),* Kolkata, India, 2018, pp. 122-129.
[10] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, 2017, pp. 721-724.
[11] N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET),* Hammamet, Tunisia, 2018, pp. 397-402.
[12] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779-788.
[13] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448.
[14] D. Arifoglu, and A. Bouchachia, “Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks,” *Procedia Computer* Science, vol. 110, pp. 86-93, 2017.
[15] W. Anani and J. Samarabandu, "Comparison of Recurrent Neural Network Algorithms for Intrusion Detection Based on Predicting Packet Sequences," *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, Quebec, QC, Canada, 2018, pp. 1-4.
[16] M. Rhanoui, S. Yousfi, M. Mikram, and H. Merizak, “Forecasting Financial Budget Time Series: ARIMA Random Walk vs LSTM Neural Network,” *IAES International Journal of Artificial Intelligence (IJ-AI),* vol. 8, no. 4, pp. 317-327, 2019.
[17] M. A. M. Azizi, M. N. M. Noh, I. Pasya, A. I. M. Yassi and M. S. A. M. Ali, “Pedestrian Detection using Doppler Radar and LSTM Neural Network,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 3, pp. 394-401, 2020.
[18] M. Nasser Al-mhiqani, R. Ahmad, Z. Zainal Abidin, W. Yassin, A. Hassan and A. Natasha Mohammad, "New insider threat detection method based on recurrent neural networks," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS),* vol. 17, no. 3, pp. 1474-1479, 2020.
[19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine,” *International Workshop on Ambient Assisted Living- IWAAL 2012*, vol. 7657, pp. 216–223, 2012.
[20] D. Palaz, M. Magimai-Doss and R. Collobert, "Convolutional Neural Networks-based continuous speech recognition using raw speech signal," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* South Brisbane, QLD, Australia, 2015, pp. 4295-4299.

[21] J. Huang, J. Li and Y. Gong, "An analysis of convolutional neural networks for speech recognition," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia, 2015, pp. 4989-4993.

[22] Y. Wang, H. Zhou, Z. Wang, J. Wang and H. Wang, "CNN-Based End-To-End Language Identification," *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2019, pp. 2475-2479.

[23] Song-Mi Lee, Sang Min Yoon and Heeryon Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," *2017 IEEE International Conference on Big Data and Smart Computing (BigComp),* Jeju, 2017, pp. 131-134.

[24] W. Xu, Y. Pang, Y. Yang and Y. Liu, "Human Activity Recognition Based on Convolutional Neural Network," *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018, pp. 165-170.

[25] F. Palumbo, C. Gallicchio, R. Pucci and A. Micheli, "Human activity recognition using multisensor data fusion based on Reservoir Computing," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, pp. 87-107, 2016.

[26] S. Poh, Y. Tan, X. Guo, S. Cheong, C. Ooi and W. Tan, "LSTM and HMM Comparison for Home Activity Anomaly Detection," *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2019, pp. 1564-1568.

[27] Andrej, K., "The Unreasonable Effectiveness of Recurrent Neural Networks," Karpathy.github.io., 2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[28] K. Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio, "On the properties of neural "machine translation: Encoder-Decoder approaches," *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103-111, 2014.

[29] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539-1548,

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Neural Information Processing Systems (NIPS) Workshop on Deep Learning, Montréal*, Canada, 2014.

[31] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on learning Representations*, 2015.

BIOGRAPHIES OF AUTHORS

Yi-Fei Tan obtained her B.Sc. (Hons), M.Sc. and PhD from University of Malaya (UM). She is currently a senior lecturer at the Faculty of Engineering in Multimedia University (MMU), Cyberjaya, Malaysia. Her research interests include machine learning, image processing, big data analytics and queueing theory.

Guo Xiaoning obtained her BEng (Hons) in Mechatronics Engineering from Lancaster University, UK and PhD in Electrical and Electronics Engineering from Nottingham University, UK. She is current a lecturer and researcher in the Faculty of Engineering at Multimedia University (MMU), Malaysia. Her research interests include machine learning, computer vision and predictive analytics.

Soon-Chang Poh obtained his B. Eng. (Hons.) Electronics and Master of Engineering Science from Multimedia University, Malaysia. He is currently a research officer at the Faculty of Engineering in Multimedia University (MMU), Cyberjaya, Malaysia. His research interests include deep learning and anomaly detection.