Wi-Fi CSI Based Human Sign Language Recognition using LSTM Network

Hasmath Farhana Thariq Ahmed
School of Computer Science and Engineering
Taylor's University
Selangor, Malaysia
0000-0002-9159-7804

Hafisoh Ahmad
School of Computer Science and Engineering
Taylor's University
Selangor, Malaysia
Hafisoh.Ahmad@taylors.edu.my

Swee King Phang
School of Computer Science and Engineering
Taylor's University
Selangor, Malaysia
sweeking.phang@taylors.edu.

Houda Harkat
Instituto de Telecomunicações
Instituto Superior Técniconame of organization
Lisboa, Portugal
Houda.HARKAT@usmba.ac.ma

Kulasekharan Narasingamurthi
Simulation CoE
Valeo India Pvt Ltd.,
Chennai, India
0000-0001-7919-7229

***Abstract*—Human sign language gesture recognition is an emerging application in the domain of Wi-Fi-based recognition. The recognition application utilizes the Channel State Information (CSI) of the Wi-Fi signal and captures the human gestures as signal amplitude and phase values. Most existing gesture recognition studies utilize only the amplitude values ignoring the phase information. Few works use both amplitude and phase information for recognition application. Besides, the existing studies adopt deep learning networks, especially Convolutional Neural Network (CNN), to improve recognition performance better. This motivates the present work to study the influence of using (i) amplitude values and (ii) amplitude and phase values together, using the Long Short-Term Memory (LSTM) network, as an alternate for CNN. Moreover, the proposed LSTM framework is fed with the CSI values without much pre-processing applied on it, except standardizing the data to make it more suitable for classification. This paper applies the proposed LSTM framework on a public sign language gesture dataset, SignFi with Adam and SGDM optimizer and analyses the performance with increasing hidden units. LSTM reported better recognition performance using Adam with 150 hidden units, and reported 99.8%, 99.5%, 99.4% and 78.0% for lab 276, home 276, lab+home 276 and lab 150 datasets, respectively.**

Keywords—Gesture Recognition, Wi-Fi, CSI, LSTM, Deep Learning.

I. INTRODUCTION

Recent advancements in IoT pave the way for technological innovations that enable smart living environment for humans. The smart living environment has its applications in the healthcare industry that primarily adopt wearable sensors [1-4] or wireless sensing technologies [5-8] that monitor the wellbeing of people in an ambient environment. Wireless sensing seems preferable to wearable-based sensing methods as it performs the sensing task in a non-intrusive manner [9]. Wireless sensing technologies widely adopt radar, Wi-Fi or RFID for healthcare applications [10]. Wi-Fi is readily available in all indoor environment and easy to deploy; the present work considers Wi-Fi protocol for human sign language recognition application.

Sign language recognition has several applications and considered a non-verbal mode of communication among people with hearing impairments. Wi-Fi-based sign-language recognition automates the sign gesture recognition task by adopting model-based or learning-based approaches [9]. Learning-based methods adopt either a shallow or a deep learning approach. This paper adopts a deep-learning approach via a Long Short-Term Memory (LSTM) framework for sign-language recognition utilizing Channel State Information (CSI) metric of Wi-Fi signal. Generally, the CSI values are acquired through physical experiments with a Wi-Fi transmitter and a receiver. The acquired CSI values are a matrix of complex numbers having both amplitude and phase values. This CSI data reveals different signal reflection characteristics due to human movements within the sensing environment.

The existing studies widely perform Wi-Fi based gesture recognition by applying more than one pre-processing techniques on the raw CSI values sequentially to denoise the acquired signal. The raw CSI values contain useful information, and when pre-processed, there is a high probability of losing significant signal characteristics [11-14]. Also, to eliminate the laborious task of manual feature extraction, most of the reported studies adopt a deep learning approach that performs auto feature extraction. Convolutional Neural Network (CNN) framework is one of the popular deep learning networks in Wi-Fi-based gesture recognition studies for achieving better recognition performance [15].

This paper implements LSTM framework for the sign gesture recognition task, with minimal pre-processing applied on the raw CSI values. The present work standardizes the raw CSI values to make it suitable for the deep learning network, instead of transforming the signals by applying filters. Further, compare the performance of the neural network with two data types: (i) data with amplitude values alone (A) and (ii) data with amplitude and phase values (A+P). The existing gesture recognition studies adopting LSTM consider amplitude value and eliminate phase values, as it may distort the performance as the phase values are sensitive to any change in the sensing environment [11]. The present work proposed to implement the LSTM architecture and validate the methodology on a public Wi-Fi CSI dataset SignFi [16].

To summarize, the following are the contributions of the present work:

1) The raw CSI values are standardized for the classification task, eliminating the application of rigorous pre-processing techniques.

2) The recognition performance is measured using amplitude values and amplitude + phase values.

3) Implemented a simple LSTM framework that performs sequence-to-label classification to assess the recognition performance.

4) Evaluated the performance of LSTM with two different optimizers SGDM and Adam, with an increasing number of hidden units.

II. Related Work

Presently, the gesture recognition paradigm shifts from conventional machine learning algorithms to deep learning algorithms due to improved recognition performance. The main reason for this migration is that deep learning automates the feature extraction or selection process, reduces manual effort, and saves time. CNN is the widely adopted deep learning algorithm in the domain of Wi-Fi CSI gesture recognition. CNN is popular as it can extract local features more effectively and perform quick network training of the samples supplied. However, CNN lack of memorizing the time-series information or sequence of data. In contrast, LSTM is a type of RNN that can store time-series information and achieve temporal order of data. Thus, the LSTM framework recently gains attention in performing Wi-Fi based gesture recognition with achieved performance.

The existing studies that adopt the LSTM framework for Wi-Fi CSI based gesture recognition mainly rely on amplitude values of CSI. The acquired CSI values consist of high-frequency noise along with the signal information. Therefore, most of the existing studies pre-process the raw signal to denoise or filter the noise present in the signal and feed the pre-processed data to the LSTM network. Activity recognition is done by exploiting only the amplitude values of CSI [17]. The raw amplitude values are pre-processed using Principal Component Analysis (PCA) and bandpass filter with STFT and converted the signal values into spectrogram image to fed as input to the Dense-LSTM network and achieved around 90% accuracy.

The amplitude values are pre-processed for complex-motion recognition using Hampel filter and lowpass filtering technique as denoising method followed by PCA based dimensionality reduction [18]. Later, the pre-processed signal values are segmented to extract statistical features and fed as input to the LSTM network and achieved 96% accuracy. A fall detection [19] system pre-process amplitude values using lowpass Butterworth filter and adopt PCA based feature extraction with STFT and achieved 80% accuracy using LSTM. DWT applied along with LSTM with a lowpass filter and PCA based pre-processing technique to recognize a person from a crowded environment and achieved 95% accuracy [20]. The raw CSI amplitude values are pre-processed with DWT [21] and extracted time-frequency features as input to LSTM and achieved 95% accuracy. Similar DWT based pre-processing is applied for gait recognition with amplitude values with varying users in the sensing environment and achieved 98% and 96% (for two people) and 92% and 91% (for a group of 8 people) [22]. LSTM adopted for complex gesture recognition in dynamic time and achieved 98.9% accuracy [23].

Fingerpass [24] adopt LSTM based finger gesture recognition study utilizing both amplitude and phase values of CSI for finger gesture recognition and user authentication, respectively. Some recent studies also adopt the LSTM framework with the spatial features of CNN and achieved better recognition performance [18,25-29]. For example, WiSDAR [26] extracted spatial features using CNN from the pre-processed amplitude values and performed classification with temporal features of LSTM with 96% recognition accuracy. Deepcount [28] apply a Butterworth filter with Weighted Moving Average (WMA) and PCA to pre-process the raw amplitude values and extract auto CNN features. Later, the CNN features are fed as input to the LSTM network for classification and achieved 90% in counting people in a crowded environment. Other reported work also adopt Bidirectional LSTM, an extension of the LSTM framework for gesture recognition using only amplitude values [12, 30].

III. Materials and Methods

A. Channel State Information

CSI is a complex matrix that consists of amplitude and phase values that reveal the signal propagation characteristics in the wireless medium between a transmitter and receiver. The Orthogonal Frequency Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO) scheme's implementations in wireless technologies enable extraction of CSI values from commercial COTS device. The transmitter is usually a commercial Wi-Fi device, and a laptop with NIC will act as a receiver. The reflected signal from the transmitter due to the human movements reveal different characteristics of gesture patterns and captured in the receiver end. In a wireless medium, the signal propagation between the transmitter (T) and the receiver (R) is captured in the form of Channel Frequency Response (CFR), represented as CSI as follows,

$$R = HT + N \tag{1}$$

H is the complex CSI matrix, and N is noise. However, the CSI values can be acquired by accessing the network's physical medium with the specialized hardware devices. The receiver laptop is widely equipped with Intel 5300 Network Interface Card (NIC) or Atheros NIC for data acquisition with corresponding CSI tool [31, 32]. It is worth noting that the value of CSI should be captured instantaneously due to unstable channel condition.

B. SignFi Dataset

The present work validated the proposed LSTM framework on a public American Sign Language (ASL) sign language dataset, SignFi [16]. The SignFi dataset contains CSI values acquired from two different environments: Home and Lab. In a single-user scenario, 8,280 samples were acquired (Lab + Home 276), for 276 ASL gestures. 5520 samples acquired from lab with 20 instances per gesture (Lab 276) and 2760 samples from home with 10 instances per gesture (Home 276). Besides, for 150 gestures, 7500 samples were acquired in lab environment, with 5 different users, with 10 instances per gesture (Lab 150). The SignFi dataset consists of raw CSI values in 4D matrix, that represent number of CSI values, number of subcarriers, number of receiving channel and gesture instance. Thus for an individual gesture instance the CSI data is of size 200×30×3, i.e., 200 CSI samples acquired from 30 subcarriers with 3 receiving channels. The gestures are labelled manually as an individual vector format that corresponds to the gesture instance. For more detailed information about the environmental setup and data acquisition of SignFi dataset, the interested readers are directed to refer [16].

The raw CSI values are extracted as a complex value consisting of amplitude and phase values for 30 subcarriers. For experimental evaluation, the amplitude and phase values are extracted separately, using MATLAB. The dataset matrix with only amplitude values (A) for 30 subcarriers is 200×30×3. With phase values appended to the amplitude, the resultant matrix contains both amplitude and phase values (A+P) for 30 subcarriers is 200×60×3.

Once the A and the A+P values are extracted, the present work standardize the raw CSI values using 'standard score' normalization technique [33]. The A and A+P values of the SignFi datasets are standardized for every column of data by subtracting the individual data with the mean value. Later, standard deviation is computed, and the dataset is standardized as follows:

$$D_{Std} = \left(D_{Raw} - \mu(D_{Raw})\right)/\sigma(D_{Raw}) \tag{2}$$

D_{Raw} is the input representing the raw CSI amplitude and phase values, and D_{Std} is the standardized dataset, respectively, with mean 'μ' and standard deviation 'σ'.

C. LSTM Framework

A simple LSTM framework is implemented in the present work that trains the network with sequence-to-label classification. The standardized input data is loaded as a sequence of cell array containing the total number of instances in a 3D matrix format. Thus, 3D matrix of size 200×30×3 and 200×60×3 with A and A+P values per instance respectively, are fed as input to the sequence input layer.

An LSTM layer is defined with hidden units followed with a flatten layer initialization that convert the input cell array in a sequence for the LSTM layer. The number of hidden units are varied as 50, 100 and 150 to understand its influence on gesture recognition performance. Later, a fully connected layer is defined with number of class labels, followed by a SoftMax and classification layer. The fully connected layer is assigned as 276 and 150 for the dataset with 276 and 150 class labels, respectively. The SoftMax layer normalizes the input sequence, and the classification layer computes the cross-entropy loss from the output size of the fully connected layer.

During training, the LSTM network efficiently capture the non-linear relationship between the gesture input data and class labels. However, there is a high chance of overfitting [21, 22]. Therefore, a 5-fold cross-validation is done to avoid overfitting by splitting the dataset into training, validation, and testing dataset in 60:20:20 proportion. The LSTM network specifies two different network training options with Adam, an adaptive moment estimation algorithm and Stochastic Gradient Descent with Momentum (SGDM) algorithm, to analyze the influence of optimizer on performance. For both optimizers, the threshold values are set to 1. The mini-batch size and epoch value set to 64 and 10, respectively, with learning rate 0.01 and the network implementation done using MATLAB.

IV. Performance Evaluation

A. Impact of Dataset (A and A+P)

LSTM network is trained with two different training options SGDM and Adam for SignFi dataset. The performance is evaluated with 50, 100 and 150 hidden units for each of optimizers. Figure 1 shows the LSTM network performance, in terms of *test* accuracy, for SGDM optimizer fed with both A and A+P datasets, for four different environments. Lab 276 environment with A data reported a gesture recognition accuracy of 84.6% with 50 hidden units, whereas the corresponding accuracy with A+P data is 78.7%. This can be attributed to the fact that the phase values are too sensitive to environmental changes and affect the recognition performance. With 100 hidden units, the recognition accuracy with A data increases to 89.1% and with 150 hidden units the accuracy further increased to 91.7%. Interestingly, no significant change in recognition accuracy observed with A+P data for both 100 and 150 hidden units for Lab 276 environment.

Home 276 environment reported lesser recognition accuracy with 50 hidden units, compared to corresponding values of Lab 276. With increasing number of hidden units, the recognition accuracy values were also observed to increase, for both A and A+P datasets. However, with A+P the recognition accuracy values drop than the corresponding A data. Similar trend in recognition performance observed in Lab+Home 276 and Lab 150 environments. Amongst all environments, lowest accuracy values are reported by Lab 150. This implies that with increasing number of users and with lesser number of instances per gesture affects the recognition performance to a greater extent.

With SGDM optimizer, across all environments, 150 hidden units with A data reported greater than 90% accuracy values except Lab 150. Highest recognition accuracy values of 91.7%, 94.0%, 93.3%, and 70.5% for Lab 276, Home 276, Lab+Home 276 and, Lab 150, respectively. Home 276 environment reported highest test recognition accuracy value of 94.0%.

Figure 2 shows the LSTM network performance, in terms of *test* accuracy, for Adam optimizer fed with both A and A+P datasets, for four different environments. Adam optimizer reported better recognition accuracy than SGDM, in all scenarios, across all environments and hidden units, for both A and A+P data. Similar to SGDM, Adam optimizer reported highest values of recognition accuracy for A data with 150 hidden units.

With Adam optimizer, across all environments, 150 hidden units with A data reported greater than 99% accuracy values except Lab 150. Highest recognition accuracy values are 99.8%, 99.5%, 99.4%, and 78.0% for Lab 276, Home 276, Lab+Home 276 and, Lab 150, respectively. Lab 276 environment reported highest test recognition accuracy value of 99.8%.

Both the optimizers with increasing hidden units reported improved *test* accuracy values with amplitude values, A. With A+P, the performance declines under all scenarios in all datasets of SignFi. LSTM tend to achieve better recognition accuracy with just amplitude values of the SignFi dataset with SGDM and Adam optimizers. Thus, further analysis of performance is done considering the amplitude values.

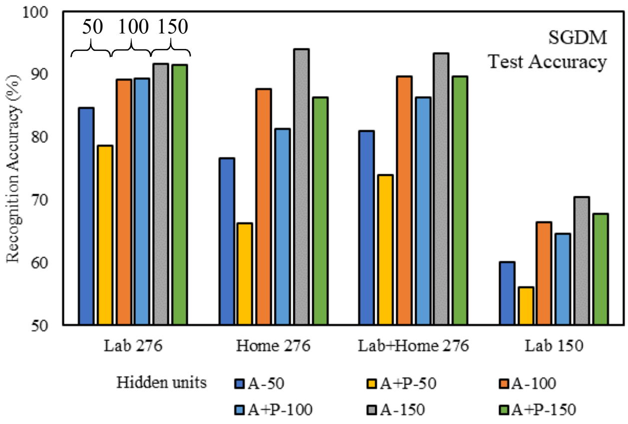

Fig. 1. LSTM performance using SGDM with A and A+P data

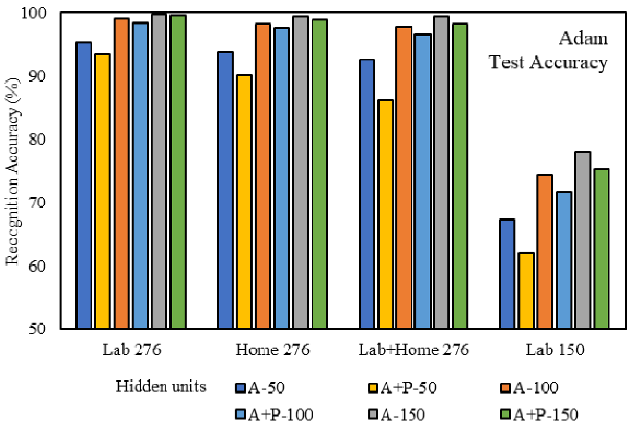

Fig. 2. LSTM performance using Adam with A and A+P data

B. *Impact of Optimizers*

Fig. 3 represents the LSTM network's test accuracy with amplitude values of SignFi datasets adopting Adam and SGDM optimizers for 50, 100 and 150 hidden units. For example, in the lab 276 dataset, with 50 hidden units, SGDM reported 84.6% *test* accuracy. The *test* accuracy value improved to 95.3% with Adam. Similarly, with 100 and 150 hidden units, the *test* accuracy values improved by 10% and 8.1%, respectively, with Adam than with SGDM. This increasing trend in *test* accuracy values was observed in home 276, lab+home 276 and lab 150 datasets with Adam optimizer compared to SGDM. It is to be noted that the *validation* accuracy values also show improvement with Adam than SGDM. The comparative analysis indicates that Adam reports better recognition performance than SGDM under all scenarios.

C. *Impact of Hidden Units*

Fig. 4(a) and 4(b) represent the comparison of LSTM performance by varying the hidden unit's number to 50, 100 and 150 with SGDM and Adam, respectively. LSTM reported better recognition performance with an increasing number of hidden units for the SignFi dataset (with amplitude values) irrespective of the optimizer. For example, SGDM with increasing hidden units for the lab 276 dataset reported increased *test* accuracy value from 84.6% to 91.7% by increasing the hidden unit number from 50 to 150. Similarly, Adam reported an increase in test accuracy value from 95.3% to 99.8% by increasing the hidden units from 50 to 100 on the lab 276 dataset.

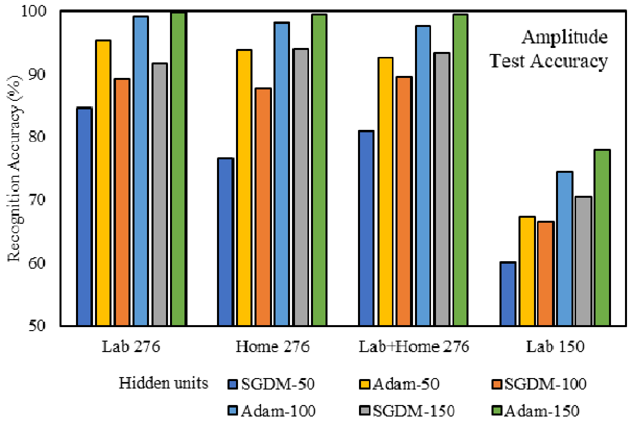

Fig. 3. Impact of SGDM and Adam on recognition performance

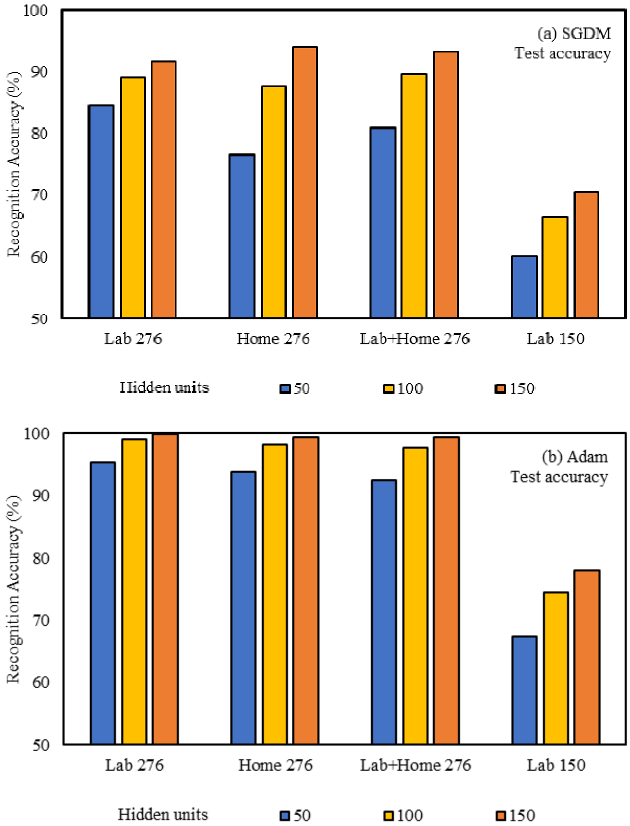

Fig. 4. Impact of hidden units on recognition performance

The other datasets such as home 276, lab+home 276 and lab 150 also reported an increasing trend in performance with increasing hidden units with both the optimizers.

D. *Impact of Environment*

The impact of environment on recognition performance are analyzed considering the highest recognition accuracy values reported with LSTM using amplitude values. LSTM with Adam optimizer and 150 hidden units reported better recognition performance of 99.8%, 99.5%, 99.4% and 78.0% for lab 276, home 276, lab+home 276 and lab 150 datasets, respectively. The reported accuracy values indicate that LSTM reports better performance in a single-user environment irrespective of the environmental impediments. However, the accuracy declines with more than 1 user, as it is challenging to achieve better accuracy with multiple users.

E. Time consumption for training and testing

Tables I and II report training and testing time of SGDM and Adam optimizers, respectively. The reported time values are obtained for the A values of SignFi datasets per instance. SGDM with increasing hidden unit report increase in training time and testing time, except for lab 276 dataset. For the lab 276 dataset, the training and testing time decreased slightly with 100 hidden units compared with 50 hidden units and increased with 150 hidden units. Similarly, Adam reports an increase in training and testing time with increasing hidden units. However, both optimizers spend more time on training than testing. Comparatively, the overall time taken by Adam for training and testing is higher than the SGDM with increasing hidden units.

TABLE I. TIME CONSUMPTION (IN MILLI SECONDS) - SGDM

Hidden units	**50**		**100**		**150**	
Dataset	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Lab 276	15.65	0.62	14.28	0.41	18.14	0.47
Home 276	11.05	0.42	13.90	0.48	17.21	0.48
Lab + Home 276	10.59	0.31	14.99	0.37	18.72	0.44
Lab 150	12.26	0.32	15.67	0.37	18.78	0.45

TABLE II. TIME CONSUMPTION (IN MILLI SECONDS)- ADAM

Hidden units	**50**		**100**		**150**	
Dataset	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Lab 276	13.82	0.39	14.98	0.39	21.12	0.52
Home 276	17.85	0.65	19.92	0.53	24.63	0.61
Lab + Home 276	12.84	0.35	15.90	0.35	20.81	0.43
Lab 150	18.62	0.47	20.00	0.67	21.15	0.43

The calculations are done using Intel® Core(TM) i5-1035G1 processor CPU + Intel® UHD Graphics GPU, 9GB memory @1.00GHz clock speed, 20GB RAM.

F. Comparison with Literature

Table III compare the accuracy values of present work achieved using LSTM with the reported work in literature, comparing the algorithm adopted, number of gestures considered. A combined Bi-RNN + IndRNN algorithm [12] for 6 gestures utilizing only A data, reported improvement in the recognition accuracy with original dataset than the filtered dataset. For A+P data, a combined approach [27], for 6 gestures, reported highest recognition accuracy of 95% compared to other approaches. In [28], adopting a CNN-LSTM approach, a higher recognition accuracy achieved than the baseline method.

TABLE III. RECOGNITION PERFORMANCE ON SIGNFI DATASET

Reference	**A / P / A+P**	**Algorithm**	**Number of Gestures**	**Recognition Accuracy**
[12]	A	Bi-RNN + IndRNN	6	Original - 93.4%, 93.5%, 94.9%, 87.8%, 95.1%, 96.7% Filtered - 88.0%, 93.4%, 93.9%, 95.3%, 91.2%, 97.3%
[27]	A+P	LSTM, De-noised LSTM, CNN + LSTM, Combined	6	Baseline LSTM - 75%; De-noised LSTM - 86%; CNN-LSTM - 84%; Combined model - 95%
[28]	A+P	Baseline, CNN + LSTM, HMM	8	Dataset-fixed, semi, open : 88.8%, 80.2%, 78.0% (Baseline method); 88.8%, 85.2%, 85.2% (CNN-LSTM); Average accuracy with door switch (samples): 87.5% (100), 92.5% (200) Amendment with HMM: Baseline method-87.0%, CNN-LSTM-90.0%
[34]	A+P	CNN, LSTM, ABLSTM	276	Home 276, Lab 276, Lab + Home 276 and Lab 150 : CNN - 99.6%, 99.8%, 99.7%, and 93.8% LSTM - 74.1%, 49.8%, 76.9%, and 56.9%; 63.5% (Self-test) ABLSTM - 95.4%, 96.2%, 94.9%, and 73.83%; 70% (Self-test)
Present work	A, A+P	LSTM	SignFi 276	Lab 276, Home 276, Lab + Home 276, and Lab 150 : A - 99.8%, 99.5%, 99.4%, and 78.0% A+P - 99.6%, 98.9%, 98.3%, and 75.3%

Further improvement in recognition accuracy observed with HMM amendment. The reported work [34] performed the recognition with A+P using CNN, LSTM and ABLSTM. However, we are interested in comparing our results with LSTM and ABLSTM neural network. In comparison to the performance reported in the literature, the present work reported better accuracy values using LSTM with A and A+P data. It is to be noted that the present work did not use any filtering technique except a simple normalization procedure, without losing the non-Gaussianity information in the input data. It is to be noted that the present work did not use any filtering technique except a simple normalization procedure, without losing the non-Gaussianity information in the input data.

V. CONCLUSIONS

The present work achieved better recognition performance exploiting the amplitude values of the SignFi dataset with Adam optimizer. Besides, with an increasing number of hidden units, the recognition performance shows improvement for all datasets. LSTM with Adam optimizer and 150 hidden units reported better recognition performance of 99.8%, 99.5%, 99.4% and 78.0% for lab 276, home 276, lab+home 276 and lab 150 datasets, respectively. Improving the recognition performance in a multiple user environment is extended as future work. The recognition performance can be improved with proper data preparation and minimal pre-processing on the raw CSI values. Moreover, the selection of proper optimizer and the hyperparameters for training the neural network also greatly impacts the performance despite environmental factors.

ACKNOWLEDGMENT

This work was supported by Taylor's University, Malaysia through its TAYLOR'S PhD SCHOLARSHIP Programme.

REFERENCES

[1] A. WG, "Smart Living Environment for Ageing Well," 2015.

[2] S. Alani, S. N. Mahmood, S. Z. Attaallah, H. S. Mhmood, Z. A. Khudhur, and A. A. Dhannoon, "IoT based implemented comparison analysis of two well-known network platforms for smart home automation," *International Journal of Electrical & Computer Engineering (2088-8708),* vol. 11, no. 1, 2021.

[3] S. H. M. Siddiqa, K. Apurva, D. Nandan, and S. Kumar, "Documentation on Smart Home Monitoring Using Internet of Things," in *ICCCE 2020*: Springer, 2021, pp. 1115-1124.

[4] M. O. Agyeman and A. Al-Mahmood, "Design and implementation of a wearable device for motivating patients with upper and/or lower limb disability via gaming and home rehabilitation," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019, pp. 247-252: IEEE.

[5] X. Wang, C. Yang, and S. Mao, "On CSI-Based Vital Sign Monitoring Using Commodity Wi-Fi," *ACM Transactions on Computing for Healthcare,* vol. 1, no. 3, pp. 1-27, 2020.

[6] Y. Zeng, D. Wu, R. Gao, T. Gu, and D. Zhang, "FullBreathe: Full human respiration detection exploiting complementarity of CSI phase and amplitude of Wi-Fi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 2, no. 3, pp. 1-19, 2018.

[7] F. F. S. Ariffin, L. M. Kamarudin, N. Ghazali, H. Nishizaki, A. Zakaria, and S. M. M. bin Syed Zakaria, "Inhalation and Exhalation Detection for Sleep and Awake Activities Using Non-Contact Ultra-Wideband (UWB) Radar Signal," in *Journal of Physics: Conference Series*, 2021, vol. 1755, no. 1, p. 012038: IOP Publishing.

[8] S. W. Shah and S. S. Kanhere, "Smart user identification using cardiopulmonary activity," *Pervasive and Mobile Computing,* vol. 58, p. 101024, 2019.

[9] H. F. T. Ahmed, H. Ahmad, and C. Aravind, "Device free human gesture recognition using Wi-Fi CSI: A survey," *Engineering Applications of Artificial Intelligence,* vol. 87, no. 2020, pp. 1-19, 2020, Art. no. 103281.

[10] S. A. Shah, H. Abbas, M. A. Imran, and Q. H. Abbasi, "RF Sensing for Healthcare Applications," *Backscattering and RF Sensing for Future Wireless Communication,* pp. 157-177, 2021.

[11] Y. Ma *et al.*, "Location-and Person-Independent Activity Recognition with Wi-Fi, Deep Neural Networks, and Reinforcement Learning," *ACM Transactions on Internet of Things,* vol. 2, no. 1, pp. 1-25, 2021.

[12] Y. Cao *et al.*, "Contactless body movement recognition during sleep via Wi-Fi signals," *IEEE Internet of Things Journal,* vol. 7, no. 3, pp. 2028-2037, 2019.

[13] H. F. T. Ahmed, H. Ahmad, S. K. Phang, C. Aravind, H. Harkat, and K. Narasingamurthi, "Higher Order Feature Extraction and Selection for Robust Human Gesture Recognition using CSI of COTS Wi-Fi Devices," *Sensors,* vol. 19, no. 13, p. 2959, 2019.

[14] H. F. T. Ahmed, H. Ahmad, S. K. Phang, C. Aravind, H. Harkat, and K. Narasingamurthi, "Sign language gesture recognition with bispectrum features using SVM," *AIP Conference Proceedings,* vol. 2233, no. 1, p. 030001, 2020.

[15] T. Nakamura, M. Bouazizi, K. Yamamoto, and T. Ohtsuki, "Wi-Fi-CSI-based Fall Detection by Spectrogram Analysis with CNN," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 2020, pp. 1-6: IEEE.

[16] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using Wi-Fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 2, no. 1, pp. 1-21, 2018, Art. no. 23.

[17] J. Zhang *et al.*, "Data augmentation and dense-lstm for human activity recognition using Wi-Fi signal," *IEEE Internet of Things Journal,* 2020.

[18] H. Lee, C. R. Ahn, and N. Choi, "Exploiting Multiple Receivers for CSI-Based Activity Classification Using A Hybrid CNN-LSTM Model," in *Proceedings of the 1st ACM International Workshop on Device-Free Human Sensing*, 2019, pp. 18-21.

[19] S. Dayal, H. Narui, and P. Deligiannis, "Human fall detection in indoor environments using channel state information of Wi-Fi signals," ed: Stanford University, Published May, 2015.

[20] F. Wang, W. Gong, J. Liu, and K. Wu, "Channel Selective Activity Recognition with Wi-Fi: A Deep Learning Approach Exploring Wideband Information," *IEEE Transactions on Network Science and Engineering,* 2018.

[21] J. Ding and Y. Wang, "Wi-Fi csi-based human activity recognition using deep recurrent neural network," *IEEE Access,* vol. 7, pp. 174257-174269, 2019.

[22] J. Ding, Y. Wang, and X. Fu, "Wihi: Wi-Fi Based Human Identity Identification Using Deep Learning," *IEEE Access,* vol. 8, pp. 129246-129262, 2020.

[23] Z. Tang, Q. Liu, M. Wu, W. Chen, and J. Huang, "Wi-Fi CSI gesture recognition based on parallel LSTM-FCN deep space-time neural network," *China Communications,* vol. 18, no. 3, pp. 205-215, 2021.

[24] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong, and M. Li, "Fingerpass: Finger gesture-based continuous user authentication for smart homes using commodity Wi-Fi," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 201-210.

[25] M. T. Hoang *et al.*, "A CNN-LSTM Quantifier for Single Access Point CSI Indoor Localization," *arXiv preprint arXiv:2005.06394,* 2020.

[26] F. Wang, W. Gong, and J. Liu, "On spatial diversity in WiFi-based human activity recognition: A deep learning-based approach," *IEEE Internet of Things Journal,* vol. 6, no. 2, pp. 2035-2047, 2018.

[27] D. A. Khan, S. Razak, B. Raj, and R. Singh, "Human Behaviour Recognition Using Wi-Fi Channel State

Information," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7625-7629: IEEE.

[28] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, "DeepCount: Crowd Counting with Wi-Fi via Deep Learning," *arXiv preprint arXiv:1903.05316,* vol. abs/1903.05316, 2019.

[29] J. Zhang, F. Wu, W. Hu, Q. Zhang, W. Xu, and J. Cheng, "WiEnhance: Towards Data Augmentation in Human Activity Recognition Using Wi-Fi Signal," in *MSN*, 2019, pp. 309-314.

[30] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wi-Fi CSI based passive human activity recognition using attention based BLSTM," *IEEE Transactions on Mobile Computing,* vol. 18, no. 11, pp. 2714-2724, 2018.

[31] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Linux 802.11 n CSI tool," *ACM SIGCOMM Computer Communication Review,* vol. 41, no. 1, p. 53, 2010.

[32] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity Wi-Fi," *IEEE Transactions on Mobile Computing,* vol. 18, no. 6, pp. 1342-1355, 2018.

[33] H. F. T. Ahmed, H. Ahmad, K. Narasingamurthi, H. Harkat, and S. K. Phang, "DF-WiSLR: Device-Free Wi-Fi-based Sign Language Recognition," *Pervasive and Mobile Computing,* vol. 69, no. 101289, 2020.

[34] M. R. Bastwesy, N. M. ElShennawy, and M. T. F. Saidahmed, "Deep Learning Sign Language Recognition System Based on Wi-Fi CSI," *International Journal of Intelligent Systems & Applications,* vol. 12, no. 6, 2020.

978-1-6654-4494-1/21/$31.00 ©2021 IEEE