International Journal of Innovative Computing, Information and Control
Volume **19**, Number **3**, June **2023**

ICIC International ©2023 ISSN 1349-4198
pp. **973–990**

HUMAN ACTIVITY RECOGNITION USING LONG SHORT-TERM MEMORY NETWORK

Kulwarun Warunsin[1], Kamphol Promjiraprawat[1]
and Orachat Chitsobhuk[2,*]

[1]Department of Computer Engineering
Ramkhamhaeng University
Bangkapi, Bangkok 10240, Thailand
kulwarun@ru.ac.th; kamphol.p@rumail.ru.ac.th

[2]School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand
*Corresponding author: orachat.ch@kmitl.ac.th

Received November 2022; revised February 2023

Abstract. *Human Activity Recognition (HAR) plays a significant role in the Ambient Assisted Living (AAL) system, which aims to provide sustainable healthcare for an aging population and those with special needs. HAR automatically categorizes people's activities while they wear wearable sensors. With an effective HAR system, we should be able to monitor the behavior of individuals as well as their activities and issue specific warnings as necessary. The goal of this paper is to propose a methodological framework for developing the HAR model based on an application of Long Short-Term Memory (LSTM) network. We investigated the model selection and parameters based on Cross Validation (CV) and learning rate optimization across two well-known public HAR datasets, MobiAct and WISDM. An analysis of the CV variance becomes a considerable impact on the generalization of the model's learning capability. The relationship between the CV variance and accuracy can be used to guide the selection of the fold number in k-fold CV. Our studies had shown the scientific evidence and technical guidance for solving the HAR problem with improvements not only in the proposed model's accuracy and AUC of more than 99% on average, but also in its generalization performance, which could be useful for future related studies.*

Keywords: Human Activity Recognition (HAR), Long Short-Term Memory (LSTM), Deep learning optimizer, Cross validation, Generalization performance

1. **Introduction.** Over the past few decades, an increase in the elderly population and people with special needs has been a primary challenge for sustainable healthcare development. Encouragement and support for patients to take care of themselves with a home care system contribute to circumventing a serious deficiency in nursing facilities, personnel, and budgetary resources. Consequently, modern home care services should be improved in order to satisfy their various requirements with respect to health conditions, safety, security, convenience, and independence. The Ambient Assisted Living (AAL) system has been developed with the goal of offering an autonomous, intelligent living environment to residential healthcare facilities. A wide variety of research objectives are being pursued, from sensor-based low-level data acquisition to high-level information integration and knowledge inference.

DOI: 10.24507/ijicic.19.03.973

With the rapid growth of Internet of Things (IoT) technology, the AAL integrates accessible sensors in smartphones or other wearable devices to expand the capacity of a monitoring network and data collection. The embedded sensors are capable of recording signals associated with daily motions and would be preferable due to their non-invasive nature, low cost, and ease of installation without expert assistance [1]. A range of sensors is generally required to reliably detect the various activities at the same time. To minimize the influence of the raw data, it is critical to preprocess this data in order to deliver valuable information to the application [2]. Appropriate data preprocessing can significantly improve detection performance [3]. The overlapping window facilitates the detection of movement transitions between two segments. In [4], the authors found that different activities have different repetitive periods and that the window size is a key parameter since a too-small window does not guarantee continuity of information and a too-large window can affect the recognition performance.

Feature extraction is another key performance and has always been a challenging task due to the fact that the characteristics of sensor signals for the same activity will be different, which may significantly degrade the recognition accuracy [5,6]. In principle, Human Activity Recognition (HAR) plays an important role in automating the AAL services. HAR performs the automatic categorization of the activities of individual people while wearing a variety of wearable sensors located throughout the body. It analyzes time series of sensor signals to extract features and categorizes them as sitting, standing, walking, jogging, falling, and going upstairs and downstairs. The HAR can be formulated as a multiple-class classification problem, which has been widely solved by statistical and machine learning approaches. Well-known traditional classification algorithms, namely, Decision Tree (DT) [7], Linear Discriminant Analysis (LDA) [7], K-Nearest Neighbors (KNN) [7], Hidden Markov Model [8], Support Vector Machine (SVM) [7,9], Random Forest (RF) [9], and so on, have been considerably studied and achieved high accuracy as classifiers for the HAR.

Due to the wide range of sensing devices, signal distortion and noise, and variations in the spatial and temporal dimensions, it is significant that the HAR is capable of comprehending such spatio-temporal characteristics of the sensor signals. In order to cope with such complex network structures, various optimization techniques should be experimented. During each epoch of training the DL network, a learning rate is perhaps the most significant hyperparameter that influences navigation in search space and the convergence speed of optimal network weights. Various learning rate optimizers have been proposed, such as Adam, RMSProp, and AdaGrad. A comparative study of their properties on the HAR model is required in order to not only select the most compatible one but also determine its best practice. Furthermore, there still exists a concern that such a complex DL network may pose a risk of overfitting and curtail generalization performance, which refers to the accuracy of the trained classifier on a new dataset that has never been trained before. k-fold cross-validation is widely recommended to evaluate the generalization performance and to select the optimal classifier due to the increased amount of independent test data in the evaluation process. Splitting entire available training data into k folds and then training repeatedly k times with a different test fold is a general concept to obtain an average of generalization accuracy. Determining the most appropriate k number is a significant challenge whose solution is strongly dependent on the particular problem which requires the experimental study on the particular dataset.

Since early 2000, there have been several researchers conducting experiments in the area of human activity recognition. In the early stage, most of the wearable device-based

HAR involves hand-crafted features such as statistical based features [10,11] from domain-specific knowledge, which may not be suitable for many complex time series activities with spatio-temporal characteristics of the sensor signals.

Deep learning based HAR has been introduced such as DNN [12], CNN [13,14], RNN [13], and LSTM [15]. Several datasets have been explored including UCI HAR [12,14] and WISDM dataset [12-16] (6 daily activities such as walking, jogging, stairs, sitting, and standing), Opportunity dataset [4,14] (13 low-level actions, 17 mid-level gesture classes, and 5 high-level activity classes such as wake up, groom, prepare breakfast, clean), PAMAP2 [14,16,17] dataset (18 daily activities such as walking, eating, car driving, vacuum cleaning, and playing soccer), MHEALTH [15,16] dataset (12 physical activities), sisFall [20,21] (19 ADLs with varied moments and 15 Falls such as fall forward, fall backward, and lateral fall occurring due to several ADLs) and MobiAct dataset [16-19] (4 different types of falls, 12 different ADLs).

Fall is another event with significant impact on physical and social independence [22-24]. Musci et al. [25] used the SisFall dataset to verify the proposed multi-layer LSTM model for Fall Detection System (FDS). The accuracy achieved 97.16% accuracy in both fall and non-fall activity. Wu et al. [26] examined CNN and LSTM structures on MobiAct dataset with 98.83% accuracy. Waheed et al. [21] proposed BiLSTM for FDS on SisFall and UP-Fall dataset and resulted in the accuracy of 97.21% and 97.41%, respectively.

In this study, we propose a methodological framework for developing the HAR model based on an application of Long Short-Term Memory (LSTM) network. The proposed methodology entails an adaptive learning rate strategy to improve the generalization performance of the obtained LSTM network. In addition, model selection and evaluation are performed using cross-validation, where the number of folds varies from 5 up to 10 depending on the recommendation from the prior related research. Our experimental results indicate that better generalization performance than traditional classification approaches can be further obtained across both well-known public HAR datasets, namely MobiAct and WISDM. Furthermore, in experimental designs, the best parameter configuration for cross validation and learning rate optimization is identified and analyzed for its characteristics and specific quality in relation to our hypothesis. Thus, this study is motivated to contribute scientific evidence and technical guidance for solving the HAR problem in order to be beneficial for forthcoming related studies.

The rest of the paper is organized as follows. Section 2 presents details of our proposed methodology regarding DL model architecture, widespread public datasets (MobiAct and WISDM), parameter optimization, model selection, and evaluation. Section 3 provides descriptions of experimental designs and reports comparative experimental results with state-of-the-art methodologies on the aforementioned datasets. Additionally, the novel findings of this paper are discussed. Finally, in Section 4, conclusions are drawn and potential future research directions are suggested.

2. **Proposed Framework.**

A schematic of the proposed framework for solving the HAR problems is illustrated in Figure 1. First of all, the selected datasets were prepared with regard to being compatible with the proposed classification model, which is designed in order to improve activity recognition. After that, various optimizers were examined through the proposed LSTM network, and the best performer was obtained. Lastly, the optimal classifier trained from the previous phase was evaluated with the k-fold cross validation technique not only to determine generalization performance but also to provide sufficient results for making comparisons with previous related work.

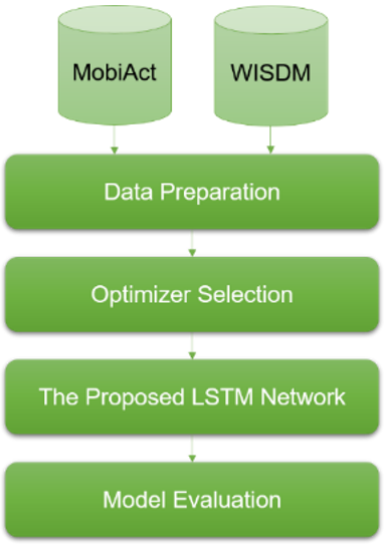

FIGURE 1. The proposed methodology

2.1. **Data preparation.** In this study, we have selected two publicly available datasets, namely MobiAct [27] and WISDM [31] for developing and evaluating our proposed DL framework. The reasons behind the selection of both datasets are that they have been carried out in manifold previous studies and provided similar time-series features in a variety of activity class labels. Additionally, consistent comparisons can be made with state-of-the-art and traditional approaches due to their popularities. Generally, both datasets were collected using the Android smartphone, which was carried by each subject while performing assigned activities. Table 1 summarizes important aspects of the datasets used in our experiments including details of activity classes and data acquisition sensors. Since WISDM has provided only triaxial accelerometer, we therefore decided to disregard gyroscope and orientation features of MobiAct from our modeling for the sake of a fair comparison and consistency. Although MobiAct includes more class labels especially for fall activities, both datasets share similar problem of imbalanced class distribution which requires particular metric to evaluate the classifier performance.

The sampling rate of the original MobiAct is somewhat high compared with WISDM, which leads to unnecessary redundancy in the learning process and waste of resources. We further reduced the sampling rate of MobiAct four times to obtain similar data sizes. Furthermore, a sliding window technique was implemented for time-series data segmentation. The previous studies have employed a wide range of windows. Walking, jogging, and going up or down the stairs have been identified using window sizes between 4 and 10 seconds [26-30]. In our study, a window length of 4-10 seconds with a 0.5-second interval was used for analysis during segmentation. Our analysis showed that a sliding window with a 5-second width produced the best outcomes. Consequently, the fixed width of the window size was 5 seconds with 90% overlapping for each of the adjacent windows in order to retain a continuum of activities. After that, the samples were randomly divided into two sets, with 70% selected for the training set and the rest (30%) for the test set.

2.2. **LSTM networks.** The size of input matrix variable is 100×3, featuring three sets of triaxial accelerometers and each of which comprises 100 raw signals. The proposed DL architecture connects two hidden layers of 64 LSTM cells. Each LSTM cell principally comprises three primary gates: input (i_t), forget (f_t) and output (o_t). These gates implement a gating mechanism from the sigmoid activation function (σ) to decide what

TABLE 1. Description of MobiAct and WISDM datasets

Dataset	MobiAct	WISDM
Subjects	57	36
Raw sensor features	9 Triaxial accelerometer Triaxial gyroscope Triaxial orientation	3 Triaxial accelerometer
Activity distribution	13 Standing: 35.52% Walking: 29.94% Jumping: 8.79% Jogging: 8.74% Upstairs: 4.57% Downstairs: 4.19% Car stepping out: 2.07% Car stepping in: 1.94% Sitting: 1.19% Falling: 3.05% Back sitting chair: 0.9% Sideward lying: 0.77% Front knees lying: 0.73% Forward lying: 0.65%	6 Walking: 38.6% Jogging: 31.2% Upstairs: 11.2% Downstairs: 9.1% Sitting: 5.5% Standing: 4.4%
Sample frequency	87 Hz	20 Hz
Raw data samples	12,159,878	1,098,207
Total sliding windows	60,675	109,762
Training (Testing)	42,472 (18,203)	76,833 (32,929)

information (The previous state hidden layer output, h_{t-1} and the current state input, $\mathbf{x}_t$) is needed and relevant to classifier performance, as given in (1)-(3). From (4), the cell state, c_t, enables a memory mechanism of temporal dependencies in the training data under the control of forget gate to prevent taking useless information into account. Besides memorizing long-term dependencies, cell state partially includes the hyperbolic tangent activation function (tanh) which maintains the values flowing in range from -1 to 1 through the iterative network training in order to avoid information fading. In addition, the cell state update is regulated by the input gate. Lastly, Equation (5) shows that the output gate further controls the hyperbolic activation flow from the cell state to the hidden layer output (h_t).

$$i_t = \sigma(\boldsymbol{\theta}_i[\mathbf{x}_t, h_{t-1}] + b_i) \tag{1}$$
$$f_t = \sigma(\boldsymbol{\theta}_f[\mathbf{x}_t, h_{t-1}] + b_f) \tag{2}$$
$$o_t = \sigma(\boldsymbol{\theta}_o[\mathbf{x}_t, h_{t-1}] + b_o) \tag{3}$$
$$c_t = c_{t-1} \odot f_t + [\tanh(\boldsymbol{\theta}_c[\mathbf{x}_t, h_{t-1}] + b_c)] \odot i_t \tag{4}$$
$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

where $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_o$, $\boldsymbol{\theta}_c$ are the LSTM network weights. b_i, b_f, b_o, b_c are bias terms. $\odot$ is element-wise multiplication.

With such a complex network structure, the obtained classifier is susceptible to overfitting due to high dimensionality of network parameters and thus the dropout strategy is applied to regularizing the hidden layer by randomly ignoring some LSTM cells in each

training epoch with respect to the dropout probability setting (The dropout probability was chosen at 7.5%, which in our framework generated the best results from 2.5% to 12.5%). The last layer is the dense layer with the softmax activation function which determines a class probability distribution for each sample. Figure 2 shows our proposed LSTM network.

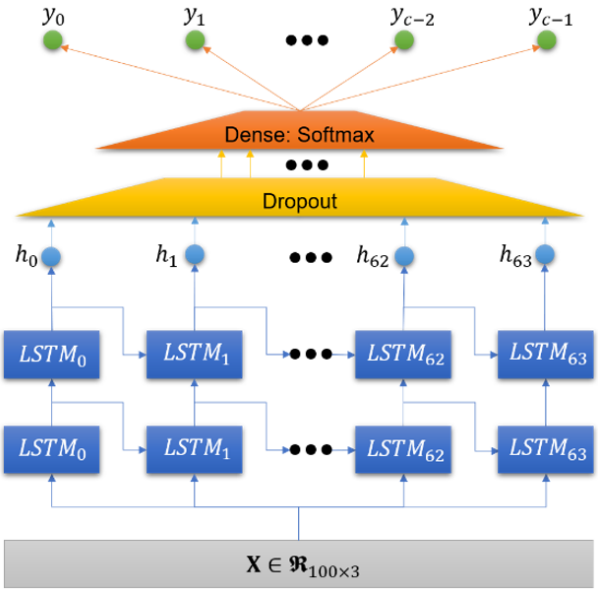

FIGURE 2. The proposed LSTM network structure

2.3. **Optimization algorithms.** The DL network learning basically relies on the concept of gradient descent with back-propagation technique to iteratively modify all network weight parameters and finally reach at the local optimum. An open-source DL library such as Keras has implemented various optimizers which mostly involve adaptation of a learning rate (η) and a gradient of loss function (∇_t). In this study, trails of available six optimizers including SGD, AdaGrad [32], RMSprop [33], AdaDelta, Adam and Adamax [34] were investigated and the best candidate was selected in order to maximize classification accuracy of our proposed LSTM network. Table 2 provides the formula for each selected optimizer and its corresponding parameters in Keras default setting. SGD is the simplest method of weight update rule which is a matrix product of the learning rate and the error gradient. AdaGrad places the focus on the learning rate component and monotonically decreases it with respect to the cumulative squared gradient (∇_t^2). As modifications of AdaGrad, RMSProp utilizes an exponential decay rate (β) for a moving average of the squared gradient to curtail excessive historical gradient. Furthermore, it does not even require the learning rate setting in AdaDelta optimizer which replaces the learning rate with the moving average of historical squared weight updates ($\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}$). Adam and its extension (AdaMax) introduce a bias-corrected momentum term into the learning rule. Adam incorporates both the first moment and the second moment which still inherits similar benefits from AdaGrad, RMSProp and AdaDelta. Additionally, gradient rescaling and step size annealing provide more stable parameter update without requirement of stationary objective for Adam rule. On the other hand, AdaMax is based on the replacement of the second moment with the infinity norm of the gradient which is normally more stable behavior.

In order to find the most appropriate optimization algorithm, performance measures in terms of accuracy and loss were examined. For both MobiAct and WISDM, all potential

TABLE 2. Mathematical formulation for optimization algorithms

Optimizer	Formula	Parameter setting
SGD	$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_t$	$\eta = 0.01$
AdaGrad	$\mathbf{d}_t = \mathbf{d}_{t-1} + \nabla_t^2$ $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\mathbf{d}_t + \varepsilon}} \nabla_t$	$\eta = 0.01$ $\varepsilon = 10^{-7}$
RMSProp	$\mathbf{d}_t = \beta \mathbf{d}_{t-1} + (1-\beta) \nabla_t^2$ $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\mathbf{d}_t + \varepsilon}} \nabla_t$	$\eta = 0.001$ $\beta = 0.9$ $\varepsilon = 10^{-6}$
AdaDelta	$\mathbf{d}_t = \beta \mathbf{d}_{t-1} + (1-\beta) \nabla_t^2$ $\mathbf{u}_t = \beta \mathbf{u}_{t-1} + (1-\beta)(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})^2$ $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\sqrt{\mathbf{u}_{t-1} + \varepsilon}}{\sqrt{\mathbf{d}_t + \varepsilon}} \nabla_t$	$\beta = 0.95$ $\varepsilon = 10^{-6}$
Adam	$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + \beta_1 \nabla_t$ $\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\beta_1^t}$ $\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + \beta_2 \nabla_t^2$ $\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_2^t}$ $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_t} + \varepsilon} \hat{\mathbf{m}}_t$	$\eta = 0.001$ $\beta_1 = 0.9$ $\beta_2 = 0.999$ $\varepsilon = 10^{-8}$
AdaMax	$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + \beta_1 \nabla_t$ $\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\beta_1^t}$ $\mathbf{L}_t = \max(\beta_2 \mathbf{L}_{t-1}, \lvert\nabla_t\rvert)$ $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\mathbf{L}_t} \hat{\mathbf{m}}_t$	$\eta = 0.002$ $\beta_1 = 0.9$ $\beta_2 = 0.999$

optimizers were tested on the training dataset (70% of the total data) and validated on the test dataset (the rest).

To provide further classification of all the formulas in the above table, the definitions of the variables involved are as follows: $\boldsymbol{\theta}_t$ is the weight vector at timestep t, η is the learning rate, ∇_t is the gradient of loss function at timestep t, ∇_t^2 is the squared gradient at timestep t, $|\nabla_t|$ is the $\mathbf{L}^2$ norm of gradient at timestep t, $\mathbf{d}_t$ is the decaying average of squared gradients at timestep t, $\mathbf{u}_t$ is the decaying average of squared weight update at timestep t, ε is the bias vector, β, β_1, β_2 are the exponential decay rates, β_1^t is the β_1 to power t, β_2^t is the β_2 to power t, $\mathbf{m}_t$ is the momentum vector at timestep t, $\hat{\mathbf{m}}_t$ is the bias-corrected momentum vector at timestep t, $\mathbf{L}_t$ is the exponentially weighted

infinity norm at timestep t, $\mathbf{v}_t$ is the second momentum vector at timestep t, and $\hat{\mathbf{v}}_t$ is the bias-corrected second momentum vector at timestep t.

2.4. **Evaluation analysis.** Both MobiAct and WISDM are imbalanced class problems. The accuracy metric is no longer sufficient to evaluate a classifier due to dominance of the majority class samples. The minority class is typically considered as the positive class. Thus, a comparison between true positive rate and false positive rate in a concept of the Receiver Operating Characteristic (ROC) curve has been acknowledged as more appropriate metric for imbalanced classification problem. Additionally, an Area Under of ROC Curve (AUC) represents the overall performance of a classifier over all possible thresholds.

In order to estimate generalization performance in terms of the above-mentioned metrics, the k-fold cross validation estimators were examined and compared to analyze their advantages and disadvantages in this study. The performance of different k-fold estimators can be compared using bias and variance, with the ideal estimator having the lowest bias and variance. The bias is the difference in performance between training and validation data, whereas the variance of an estimator relates to how far it deviates from its mean. With respect to evaluating the generalization performance, experimental results in validation phase are more important and valuable.

Another issue is determining how the k value affects cross validation results. According to previous studies [35,36], when it comes to the purpose of model selection, the k value is commonly recommended to set between 5 and 10 because statistical performance of the estimator does not improve significantly as k grows larger, and averaging across lower splits is computationally practical. Moreover, the estimator usually suffers from high variance when k is relatively small. Thus, we selected the k value between 5 and 10 for performance anlysis of the k-fold cross validation estimators.

3. **Experiment Results.** In order to solve the HAR problems using the DL classifier techniques, it is very difficult to identify its analytic solution. Thus, we strongly focus on providing numerical experiments on the topics of optimizer selection, generalization evaluation as well as comparison of our proposed DL classifier with state-of-the-art DL and conventional ML approaches. All these results were conducted in order of requirement of findings to improve the overall modeling performance. In this study, all experiments were implemented by the Google Colab cloud server with Keras DL framework. User Python code can be developed and executed through the Colab web-based platform, which is well-suited to data analysis and machine learning while also offering free access to computing resources such as GPUs. However, the Colab service must be able to control usage limitations and resource availability in order to provide a free-of-charge service. Hence, Colab service performance cannot be guaranteed due to fluctuations in service demands and limited resource consumption. For MobiAct and WISDM training of the classifier was completed within approximately 1.2-1.9 hours on average.

3.1. **Learning curve of optimizers.** Our experiments were conducted on two classification problems of time-series dataset using the proposed double layers of LSTM network with consideration of six optimizers.

With the default parameter setting from Table 2, the beginning accuracy and loss obtained from the optimized classifier training could roughly estimate the overall optimizer performance. In general, all optimizers started better for WISDM dataset. Adam, RMSProp and AdaMax can achieve the highest accuracy ($> 80\%$) and the lowest loss (< 0.5) in the first epoch for both datasets, as shown in Figures 3-5. On the other hand, it is very difficult for AdaGrad and AdaDelta to construct the accurate classifier due to

starting from much worse position. After more than ten epochs, the accuracy and loss metrics in most epochs of training phase were better than those in validation phase and their convergences were reached at approximately before 50 epochs, except AdaGrad and AdaDelta.

A distint difference of performance between training and validation phases is evidence of overfitting or underfitting problems. Figure 3 has indicated that Adam performed impressively and consistent with similar learning curves. Nonetheless, for only a few epochs, they were at risk of overfitting which occurred more for MobiAct dataset.

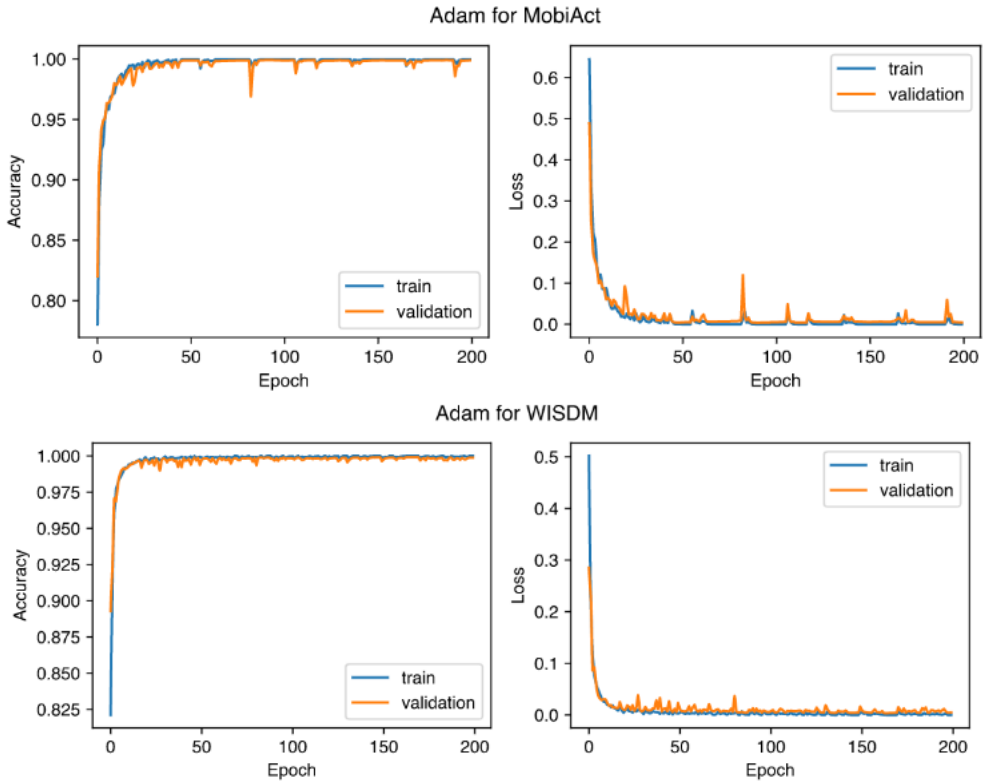

FIGURE 3. Learning curves of Adam in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

The optimizer performance is primarily influenced by the dataset used. Thus, numerical verification is an indispensable way to analyze impacts of data quality and quantity on classifier parameterization. RMSProp performed somewhat poorer than Adam regarding accuracy and loss on average. However, it was capable of handling both datasets with similar learning curves to each other as well as acceptable classifier accuracy, as shown in Figure 4. Likewise, Figure 5 also presents the robustness of AdaMax against data variation but unfortunately came with an indication of its slower convergence.

The performance of SGD was substantially different from the others with the most stringent fluctuation of learning curves, especially for early epochs. Furthermore, many epochs brought more concerns about overfitting in the case of WISDM dataset, and it seems difficult to reach at the convergence point, as shown in Figure 6.

AdaGrad and AdaDelta provided very consistent performance in training and validation phases for both datasets which implies the reliability of classification results. However, they were trapped in premature convergence leading to a local optimum in parameterization. As a consequence, Figure 7 and Figure 8 present the performance of AdaGrad and AdaDelta which was significantly lower than that of other optimizers.

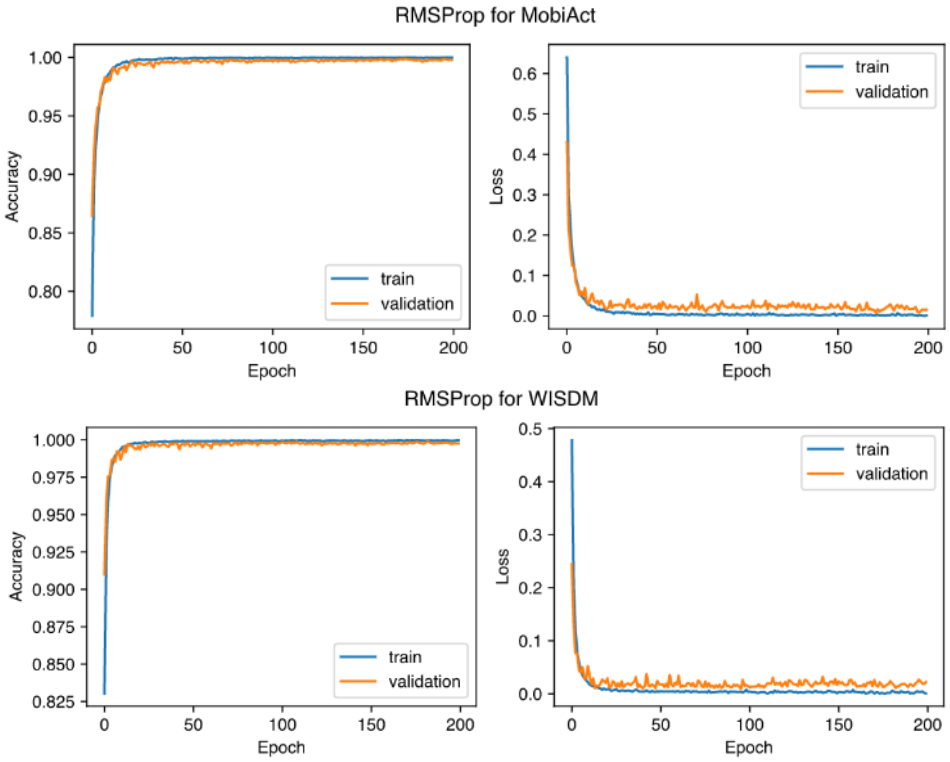

FIGURE 4. Learning curves of RMSProp in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

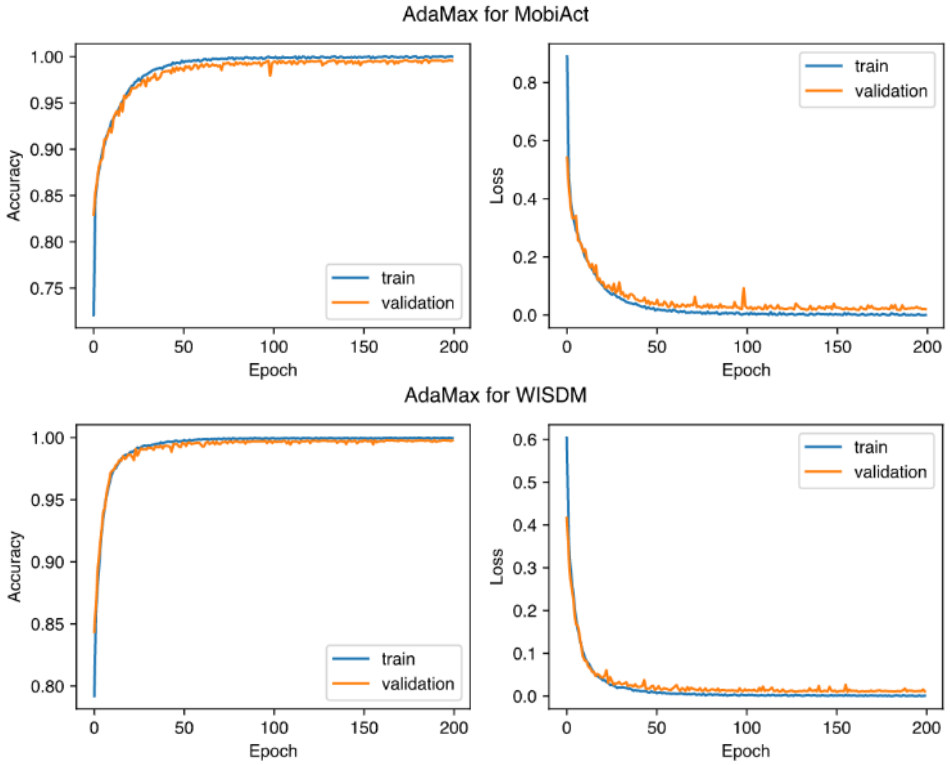

FIGURE 5. Learning curves of AdaMax in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

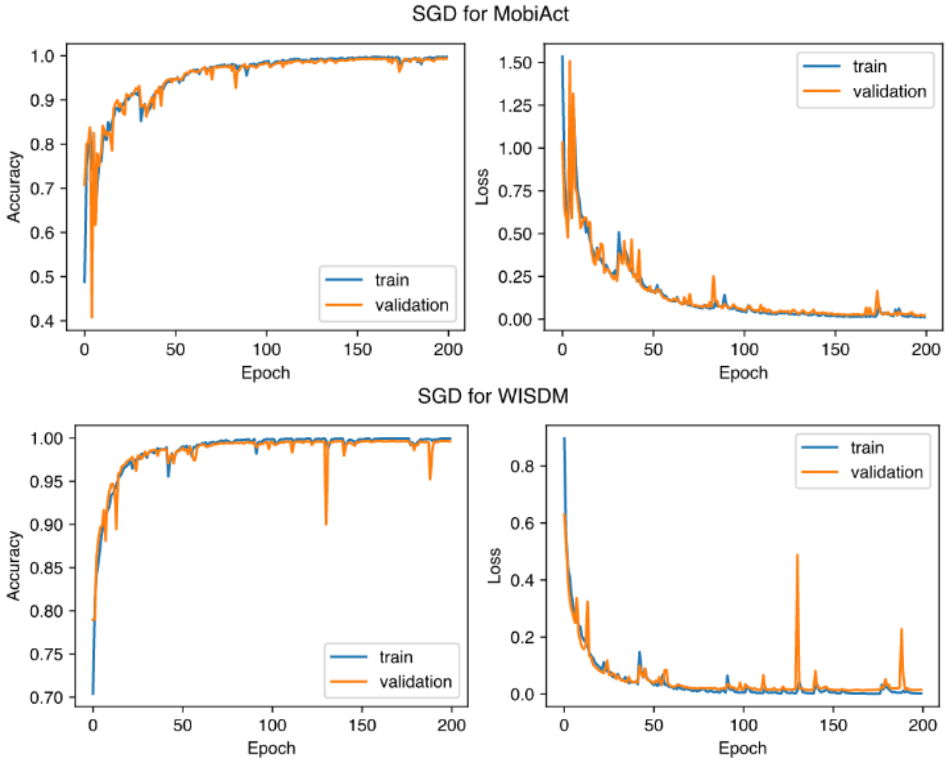

FIGURE 6. Learning curves of SGD in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

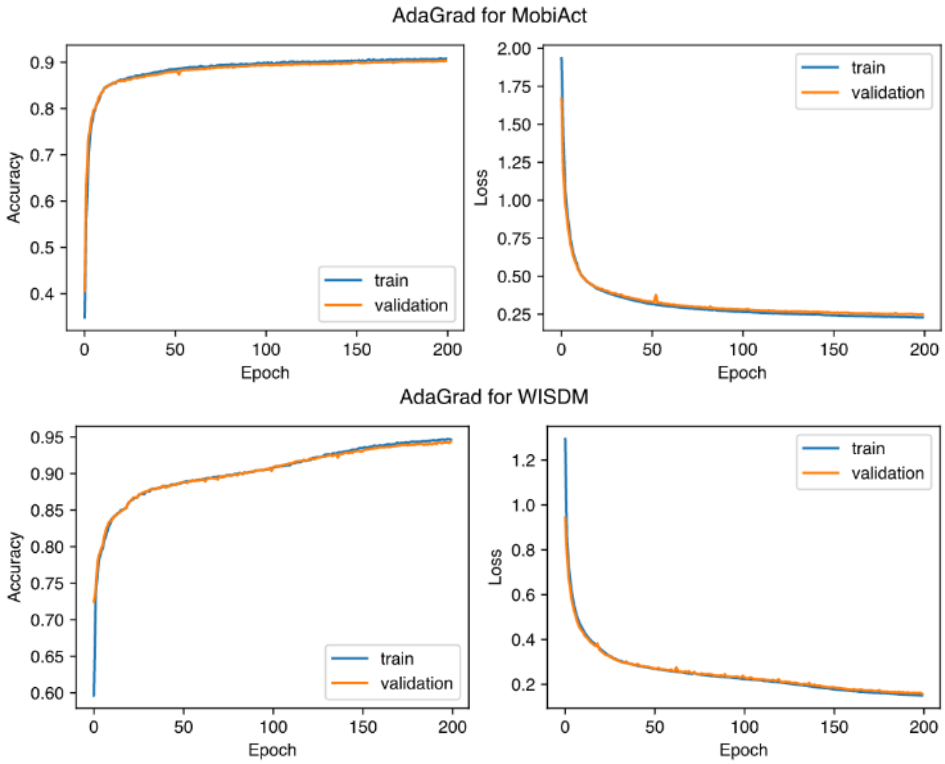

FIGURE 7. Learning curves of AdaGrad in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

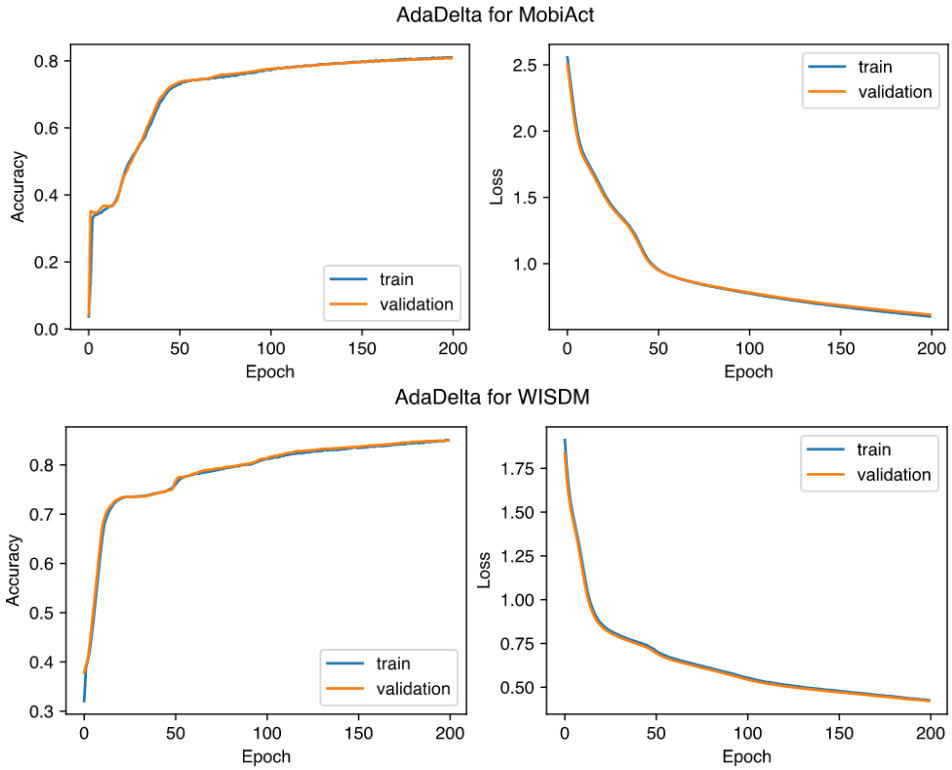

FIGURE 8. Learning curves of AdaDelta in training and validation phases for 200 epochs. From left to right, accuracy and loss across epoch sizes. From top to bottom, MobiAct and WISDM datasets.

As shown in Table 3, Adam was the best optimizer in terms of minimizing loss function and classification accuracy, during both training and testing phases for MobiAct dataset. Likewise, Table 4 also confirms that the different dataset as WISDM does not have any negative impact on the performance of Adam. For both datasets, AdaDelta was the poorest performer due to without taking the learning rate into account. Nonetheless, with a somewhat difference among most of other competitors except AdaGrad and AdaDelta, they performed well with an accuracy of more than 99% for the best epoch of each optimizer. Most of the time a loss function in classifier training is consistent with classification accuracy. Nonetheless, Table 3 and Table 4 show some conflicts between loss and accuracy of optimizers (WISDM: RMSProp and AdaMax, and MobiAct: RMSProp, AdaMax and SGD). This occurs because, unlike the accuracy that counts the number (Discrete value) of correctly classified samples, the loss function is determined by a difference.

TABLE 3. Optimizer performances for MobiAct dataset

Optimizer	**Training**		**The best epoch**	**Testing**	
	Loss	**Accuracy (%)**		**Loss**	**Accuracy (%)**
Adam	0.0000	100.00	94	0.0033	99.92
RMSProp	0.0015	99.98	172	0.0097	99.87
AdaMax	0.0049	99.99	134	0.0194	99.56
SGD	0.0186	99.51	199	0.0226	99.35
AdaGrad	0.2287	90.75	198	0.2511	90.35
AdaDelta	0.6011	81.00	200	0.6101	81.28

TABLE 4. Optimizer performances for WISDM dataset

Optimizer	Training		The best epoch	Testing	
	Loss	Accuracy (%)		Loss	Accuracy (%)
Adam	0.0001	100.00	164	0.0056	99.88
RMSProp	0.0037	99.95	96	0.0139	99.80
AdaMax	0.0011	99.97	151	0.0102	99.76
SGD	0.0021	99.97	160	0.0123	99.67
AdaGrad	0.1505	94.64	200	0.1523	94.57
AdaDelta	0.4536	84.16	174	0.4443	84.54

Since most of optimizers have performed very close to each other and provided their own benefits, performance in unseen data (Testing dataset) is appropriate to be taken into account as the final criterion for decision making. With respect to the above-mentioned evidences, we selected Adam to help seek the optimal hyperparameter configuration for the rest of our experiments.

3.2. **Variations of *k*-folds cross validations.** For all different k-fold cross validations in MobiAct and WISDM, both metrics (accuracy and AUC) and their variances in training phase are higher than those in validation phase, as shown in Figure 9 and Figure 10. Between accuracy and AUC evaluations, there were no significant differences. The variance in AUC, on the other hand, was significantly lower. Thus, for both of these datasets, the AUC measure is strongly recommended for evaluating the classifier, and these strong evidences imply that the proposed LSTM network ensures outstanding generalization performance, with an AUC of more than 0.99 on average for all cases.

Nonetheless, ranking k-fold estimators based on their performance produces significant conflicts between the two datasets, making it difficult to distinguish the best estimator from the others in general. The 5-fold was the worst estimator over WISDM in terms of high bias and high variance. However, in the case of MobiAct, it proved to be the most effective. The 9-fold, in contrast, outperformed on MobiAct but suffered with WISDM.

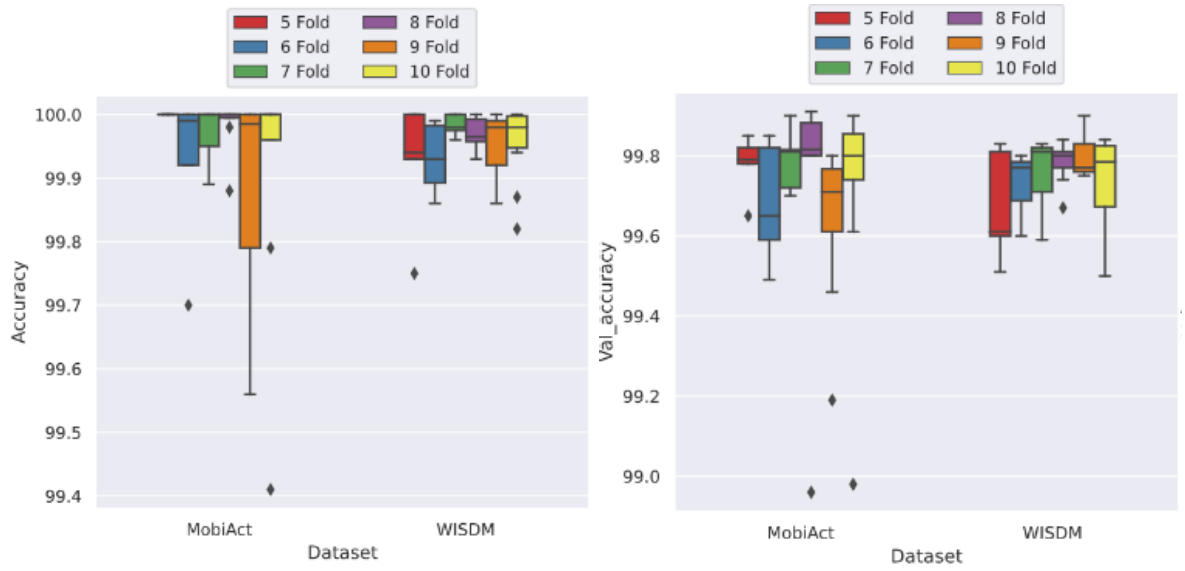

FIGURE 9. (color online) Accuracy comparison during the training (left) and validation (right) phases for MobiAct and WISDM datasets

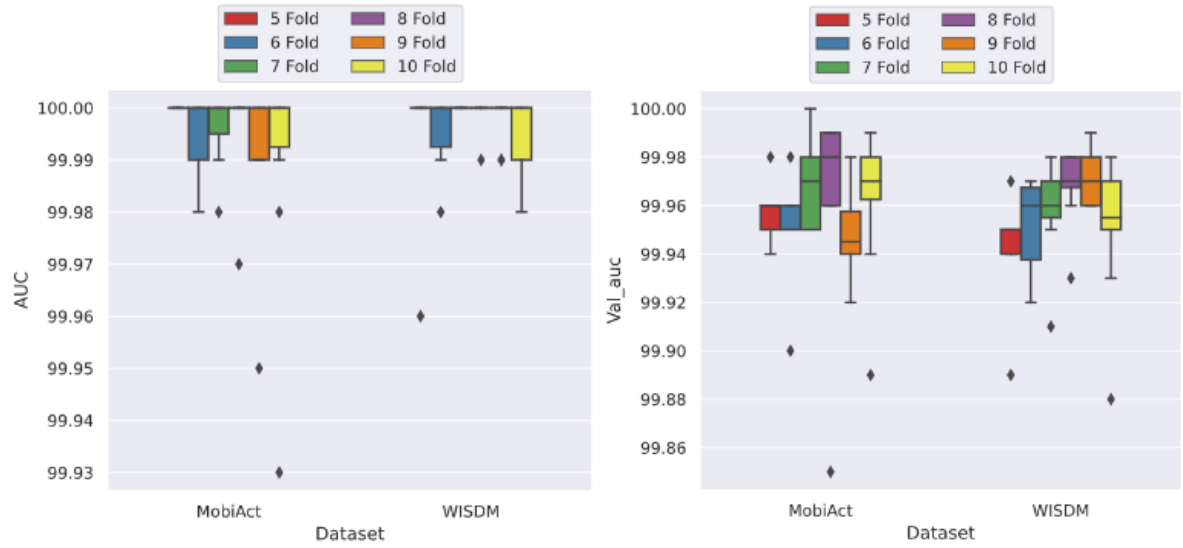

FIGURE 10. (color online) AUC comparison during the training (left) and validation (right) phases for MobiAct and WISDM datasets

In addition, the 7-fold was recommended as a robust estimator for both datasets since it had low bias and low variance on average.

We further provide a different perspective of estimator performance assessment on the given test set. Table 5 shows evidence of the high generalization performance of the proposed classification model, which is significantly robust across a variety of k-fold cross validation ($k = 5$ up to $k = 10$). The 10-fold cross validation yielded the best accuracy and AUC tested over both datasets. Nonetheless, these results pointed out the issue of inconsistent performance metrics due to an imbalanced class problem. For this case of WISDM dataset, AUC measure was very difficult to determine the best estimator.

TABLE 5. Performances of k-fold cross validations on testing datasets

***k*-fold cross validations**	**MobiAct**		**WISDM**	
	Accuracy (%)	**AUC**	**Accuracy (%)**	**AUC**
$k = 5$	99.80	0.9996	99.76	0.9996
$k = 6$	99.84	0.9993	99.74	0.9996
$k = 7$	99.80	0.9998	99.78	0.9996
$k = 8$	99.71	0.9997	99.78	0.9996
$k = 9$	99.05	0.9969	99.80	0.9996
$k = 10$	99.86	0.9998	99.84	0.9997

3.3. **Classification performances.** Fair performance comparisons with previous accomplished research were conducted with respect to the same datasets, evaluation schemes, and a class of algorithms in order to ensure the performance of our proposed methodology to deal with the HAR problem. The superiority of the proposed LSTM network over alternative non-deep-learning-based approaches such as SVM, KNN, and RF was examined in terms of classification accuracy, as shown in Table 6. Especially for the MobiAct dataset, the generalization performance using 5-fold Cross-Validation (5CV) can be significantly improved by approximately 16% on average, compared with the three above-mentioned algorithms. RF and KNN were proved to be accurate classifiers on the WISDM dataset. Nonetheless, the proposed classifier provided a marginal improvement of up to 3.8% when evaluated with 10-fold cross-validation (10CV).

TABLE 6. A comparison of the proposed method with previous works

Author	Model	Accuracy (%)	F1-score	Evaluation scheme
		MobiAct		
Proposed	**LSTM**	**99.94**	**0.9953**	Only test set
[18]	GAN	99.47	0.9941	
[37]	CNN-LSTM	98.98	N/A	
Proposed	**LSTM**	**99.88**	**0.9934**	5CV
[28]	EnsemConvNet	95.40	N/A	
[29]	SVM	87.70	0.7165	
[29]	RF	85.92	0.7077	
[29]	KNN	84.17	0.7025	
		WISDM		
Proposed	**LSTM**	**99.87**	**0.9978**	Only test set
[30]	CondConv	99.60	N/A	
[38]	RNN-LSTM	95.78	0.9573	
Proposed	**LSTM**	**99.85**	**0.9977**	5CV
[28]	EnsemConvNet	97.20	N/A	
[15]	ConvLSTM	96.00	0.9437	
Proposed	**LSTM**	**99.90**	**0.9985**	10CV
[39]	SVM	71.09	0.6940	
[39]	RF	98.09	0.9810	
[39]	KNN	97.09	0.6940	

We also made comparisons between applications of LSTM and other available options in DL algorithms. The selected LSTM model can compete with RNN, CNN, and even LSTM itself with alternative configuration designs. Our network structure based on LSTM is quite identical to that in [12], but more accurately classified. This may be because our network contains only one dropout layer with a lower rate. Enabling too much of the dropout effect may lead to an underfitting problem and a learning capability limitation. Advanced DL techniques such as convolution, ensemble, generative adversarial or hybrid lead to methodological advancement in this field. Moreover, the proposed method is proved to be an impressive alternative with having very small error less than 0.15% for all comparisons.

Since accuracy measurement has been concerned with handling the imbalanced class problem, our study also provides empirical proof with respect to F1-score in order to take bias for particular imbalanced classes into account. The results reveal that no conflict between the obtained accuracy and F1-score was found, and therefore performance of the proposed framework dominated previous related works regarding both DL-based and non-DL-based approaches at very high F1-score.

4. **Conclusions.** This article presents the proposed framework based on the LSTM network structure for classifying multiple human activities. We investigated efficient optimizers and discovered that the Adam optimizer delivered the most promising performance to build an accurate and robust classifier for both MobiAct and WISDM. In order to evaluate the generalization performance and make a fair comparison with previous studies, the cross-validation technique with a variety of k values was examined, and there was no obvious indication to determine the appropriate number of folds for both datasets. The

impact of this parameter is somewhat sensitive to variations in classification problems. Our empirical results show that the proposed LSTM network can achieve an accuracy and AUC of more than 99% on average and perform competitively when compared to previous related works. For our future research directions, we plan to validate the proposed methodology with a dataset that includes more variety in activity classes and physical condition features. With the goal of algorithm development, the LSTM concept has shown a promising potential to be improved. The development strategy will focus on seeking even better generalization performance, and the LSTM limitation will be further analyzed and provided with the development hint.

Acknowledgment. This work was supported by the Thailand Research Fund (TRF) and Office of the Higher Education Commission (OHEC) (MRG6280232).

REFERENCES

[1] Q. Ni, A. B. G. Hernando and I. P. de la Cruz, The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development, *Sensors*, vol.15, no.5, pp.11312-11362, 2015.

[2] E. Ramanujam, T. Perumal and S. Padmavathi, Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review, *IEEE Sensors J.*, vol.21, no.12, pp.13029-13040, 2021.

[3] S. B. Kotsiantis, D. Kanellopoulos and P. Pintelas, Data preprocessing for supervised learning, *International Journal of Computer Science*, vol.1, no.2, pp.111-117, 2006.

[4] S. Mahmud, M. T. H. Tonmoy, K. K. Bhaumik, A. K. M. M. Rahman, M. A. Amin, M. Shoyaib et al., Human activity recognition from wearable sensor data using self-attention, *arXiv Preprint*, arXiv: 2003.09018, 2020.

[5] G. Yuan, Z. Wang, F. Meng, Q. Yan and S. Xia, An overview of human activity recognition based on smartphone, *Sensor Rev.*, vol.39, no.2, pp.288-306, 2019.

[6] Z. Chen, Q. Zhu, C. S. Yeng and L. Zhang, Robust human activity recognition using smartphone sensors via CT-PCA and online SVM, *IEEE Trans. Ind. Informat.*, vol.13, no.6, pp.3070-3080, 2017.

[7] M. Webber and R. F. Rojas, Human activity recognition with accelerometer and gyroscope: A data fusion approach, *IEEE Sensors J.*, vol.21, no.15, pp.16979-16989, 2021.

[8] F. J. Ordonez, G. Englebienne, P. de Toledo, T. van Kasteren, A. Sanchis and B. Kröse, In-home activity recognition: Bayesian inference for hidden Markov models, *IEEE Pervasive Comput.*, vol.13, no.3, pp.67-75, 2014.

[9] S. Balli, E. A. Sağbaş and M. Peker, Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm, *Meas. Control*, vol.52, no.1, pp.37-45, 2019.

[10] D. Figo, P. C. Diniz, D. R. Ferreira and J. M. P. Cardoso, Preprocessing techniques for context recognition from accelerometer data, *Pers. Ubiquitous Comput.*, vol.14, no.7, pp.645-662, 2010.

[11] T. Plötz, N. Y. Hammerla and P. Olivier, Feature learning for activity recognition in ubiquitous computing, *Proc. of the 22nd Int. Joint Conf. Artif. Intell.*, pp.1729-1734, 2011.

[12] G. Memiş and M. Sert, Detection of basic human physical activities with indoor-outdoor information using sigma-based features and deep learning, *IEEE Sensors J.*, vol.19, no.17, pp.7565-7574, 2019.

[13] L. Shi, H. Xu, W. Ji, B. Zhang, X. Sun and J. Li, Real-time human activity recognition system based on capsule and LoRa, *IEEE Sensors J.*, vol.21, no.1, pp.667-677, 2021.

[14] Q. Teng, K. Wang, L. Zhang and J. He, The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition, *IEEE Sensors J.*, vol.20, no.13, pp.7265-7274, 2020.

[15] S. P. Singh, A. Lay-Ekuakille, D. Gangwar, M. K. Sharma and S. Gupta, Deep ConvLSTM with self-attention for human activity decoding using wearables, *arXiv Preprint*, arXiv: 2005.00698, 2020.

[16] R. Mondal, D. Mukherjee, P. K. Singh, V. Bhateja and R. Sarkar, A new framework for smartphone sensor based human activity recognition using graph neural network, *IEEE Sensors J.*, vol.21, no.10, pp.11461-11468, 2021.

[17] Y. Tang, Q. Teng, L. Zhang, F. Min and J. He, Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors, *arXiv Preprint*, arXiv: 2005.03948, 2020.

[18] Y.-H. Nho, S. Ryu and D.-S. Kwon, UI-GAN: Generative adversarial network-based anomaly detection using user initial information for wearable devices, *IEEE Sensors J.*, vol.21, no.8, pp.9949-9958, 2021.

[19] K. Warunsin and T. Phairoh, Wristband fall detection system using deep learning, *The 7th International Conference on Computer and Communication Systems (ICCCS)*, Wuhan, China, pp.223-227, 2022.

[20] J. He, Z. Zhang, X. Wang and S. Yang, A low power fall sensing technology based on FD-CNN, *IEEE Sensors J.*, vol.19, no.13, pp.5110-5118, 2019.

[21] M. Waheed, H. Afzal and K. Mehmood, NT-FDS-A noise tolerant fall detection system using deep learning on wearable devices, *Sensors*, pp.1-26, 2021.

[22] G. Khanna, M. Y. Cheng, P. Varadharajan, S. Bagchi, M. P. Correia and P. J. Verissimo, Automated rule-based diagnosis through a distributed monitor system, *IEEE Trans. Dependable and Secure Computing*, vol.4, no.4, pp.266-279, 2007.

[23] G. Tandon and P. Chan, Weighting versus pruning in rule validation for detecting network and host anomalies, *Proc. of ACM SIGKDD*, pp.697-706, 2007.

[24] M. Steinder and A. Sethi, End-to-end service failure diagnosis using belief networks, *Proc. of IEEE/IFIP Netw. Operations Manage. Symp.*, pp.375-390, 2002.

[25] M. Musci, D. De Martini, N. Blago, T. Facchinetti and M. Piastra, Online fall detection using recurrent neural networks, *arXiv Preprint*, arXiv: 1804.04976, 2018.

[26] X. Wu, L. Cheng, C.-H. Chu and J. Kim, Using deep learning and smartphone for automatic detection of fall and daily activities, *Proc. of Int. Conf. Smart Health*, pp.61-74, 2019.

[27] G. Vavoulas et al., The MobiAct dataset: Recognition of activities of daily living using smartphones, *Proc. of Int'l. Conf. Info. and Commun. Technologies for Aging Well and e-Health*, pp.143-151, 2016.

[28] D. Mukherjee, R. Mondal, P. K. Singh, R. Sarkar and D. Bhattacharjee, EnsemConvNet: A deep learning approach for human activity recognition using smartphone sensors for healthcare applications, *Multimedia Tools Appl.*, vol.79, pp.31663-31690, 2020.

[29] Z. Zheng, J. Du, L. Sun, M. Huo and Y. Chen, TASG: An augmented classification method for impersonal HAR, *Mobile Inf. Syst.*, vol.2018, pp.1-10, 2018.

[30] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu and J. He, Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices, *arXiv Preprint*, arXiv: 2006.03259, 2020.

[31] J. R. Kwapisz, G. M. Weiss and S. A. Moore, Activity recognition using cell phone accelerometers, *ACM SIGKDD Explorations Newslett.*, vol.12, no.2, pp.74-82, 2011.

[32] J. Duchi, E. Hazan and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research*, vol.12, no.7, 2011.

[33] G. Hinton, N. Srivastava and K. Swersky, *Neural Networks for Machine Learning – Lecture 6A Overview of Mini-Batch Gradient Descent*, 2012.

[34] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv Preprint*, arXiv: 1412.6980, 2014.

[35] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu and J. Hu, Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation, *Computational Materials Science*, vol.171, 2020.

[36] S. Arlot and A. Celisse, A survey of cross-validation procedures for model selection, *Statistics Surveys*, vol.4, pp.40-79, 2010.

[37] J. Xu, Z. He and Y. Zhang, CNN-LSTM combined network for IoT enabled fall detection applications, *J. Phys. Conf. Ser.*, vol.1267, 2019.

[38] P. Agarwal and M. Alam, A lightweight deep learning model for human activity recognition on edge devices, *Procedia Comput. Sci.*, vol.167, pp.2364-2373, 2020.

[39] K. H. Walse, R. V. Dharaskar and V. M. Thakare, Performance evaluation of classifiers on WISDM dataset for human activity recognition, *Proc. of the 2nd Int. Conf. Inf. Commun. Technol. Competitive Strategies (ICTCS)*, pp.1-7, 2016.

Author Biography

Kulwarun Warunsin received the B.Sc. degree in Physics from Silpakorn University, the M.S. degree in Computer Engineering and the Ph.D. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang. She is currently a lecturer at the Department of Computer Engineering, Faculty of Engineering, Ramkhamhaeng University, Thailand. Her research interests include artificial intelligence, big data analysis, the Internet of Things (IoT), and machine learning in industry applications.

Kamphol Promjiraprawat received the B.Sc. degree in Applied Mathematics from King Mongkut's University of Technology North Bangkok and the M.E. degree in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang and the Ph.D. (Engineering) degree from Sirindhorn International Institute of Technology (SIIT), Thammasat University. He is working as an Assistant Professor with the Department of Computer Engineering, Faculty of Engineering, Ramkhamhaeng University. His current research interests include artificial intelligence, machine learning, pattern recognition, big data analysis, cryptography, bioinformatics, optimization, automated planning and scheduling.

Orachat Chitsobhuk received the B.E. degree in Electronics Engineering from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1992, the M.S. degree in Computer Engineering from Arizona State University, AZ, in 1997, and the Ph.D. degree in Electrical Engineering from University of Texas, Arlington, US, in 2001. She is currently an associate professor and a lecturer at King Mongkut's Institute of Technology Ladkrabang, Thailand. Her research interests include image and scene analysis, machine learning and pattern recognition, and hardware design for image processing applications.