IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 19, 2020, accepted July 8, 2020, date of publication July 20, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010715

Deep Neural Networks for Human Activity Recognition With Wearable Sensors: Leave-One-Subject-Out Cross-Validation for Model Selection

DAVOUD GHOLAMIANGONABADI, NIKITA KISELOV, AND KATARINA GROLINGER, (Member, IEEE)
Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada

Corresponding author: Katarina Grolinger (kgroling@uwo.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2018-06222.

ABSTRACT Human Activity Recognition (HAR) has been attracting significant research attention because of the increasing availability of environmental and wearable sensors for collecting HAR data. In recent years, deep learning approaches have demonstrated a great success due to their ability to model complex systems. However, these models are often evaluated on the same subjects as those used to train the model; thus, the provided accuracy estimates do not pertain to new subjects. Occasionally, one or a few subjects are selected for the evaluation, but such estimates highly depend on the subjects selected for the evaluation. Consequently, this paper examines how well different machine learning architectures make generalizations based on a new subject(s) by using Leave-One-Subject-Out Cross-Validation (LOSOCV). Changing the subject used for the evaluation in each fold of the cross-validation, LOSOCV provides subject-independent estimate of the performance for new subjects. Six feed forward and convolutional neural network (CNN) architectures as well as four pre-processing scenarios have been considered. Results show that CNN architecture with two convolutions and one-dimensional filter accompanied by a sliding window and vector magnitude, generalizes better than other architectures. For the same CNN, the accuracy improves from 85.1% when evaluated with LOSOCV to 99.85% when evaluated with the traditional 10-fold cross-validation, demonstrating the importance of using LOSOCV for the evaluation.

INDEX TERMS Deep neural networks, human activity recognition, model selection, convolutional neural networks, feed forward neural networks, model evaluation, wearable sensor, leave-one-subject-out.

I. INTRODUCTION

Human activity recognition (HAR) aims to detect, identify, and interpret human activities employing signals received from the environment or from wearable sensors [1]. There is a wide area of HAR applications including health monitoring [2], ambient assisted living [3], and targeted advertising [4]. Intraclass variability, interclass similarity, and null-class dominance make HAR a difficult classification task [5]–[7]. Intraclass variability refers to variations of the same activity (e.g., walking) among different people or even for the same person in different recording sessions, while interclass similarity indicates similarity between different activities such as jogging and running. As large parts of the data are not labeled or do not contain relevant activities, null-class is dominant, which limits how usable the data is for modelling [7].

There are two main categories of HAR approaches based on the type of data used for recognition: vision-based and sensor-based. Vision-based approaches require installation of cameras; therefore, these systems are limited in terms of the size and condition of the monitored space and raises concerns around intrusiveness and privacy. On the other hand, advances in sensor technology have enabled HAR with wearable devices and decoupled activity monitoring from the environment. Many sensors can be applied for HAR including accelerometers, gyroscopes, magnetometer, and radio-frequency identification [8]. Because of their robustness,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanpeng Jin.

diversity, availability, and wide acceptance among the population, wearable sensors are one of the most common approaches for HAR applications [9].

To recognize human activities, many Machine Learning (ML) methods have been applied: for example, Hidden Markov Models (HMM) [10], Decision Trees [11], Support Vector Machines (SVM) [12], Conditional Random Fields [13], and K-Nearest Neighbor (K-NN) [14]. In recent years, Deep Neural Networks (DNNs) have been quite popular in machine learning and have had a significant impact on a variety of application domains [15], including object recognition [16], natural language processing [17], and energy management [18]. In HAR, DNNs, especially Convolutional Neural Networks, have demonstrated great success [1]. In deep learning architectures, multiple layers perform non-linear transformations, and input data are transformed into hierarchical representations, each one indicating different abstraction levels. In spite of recent DNNs success in HAR, model selection remains a challenge.

The model evaluation is essential for comparing results obtained by different studies as well as for selecting the best model. Studies typically use a single model or a subject-specific model. A single model approach develops one model by using data from all subjects (users) while subject-specific approach results in one model per subject. In both cases, the models are assessed by traditional hold-out or k-fold cross-validation. The drawback of these traditional evaluation methods is that data from the same person is present in the training as well as in the testing set. Consequently, the model may struggle to generalize to new (unseen) users as parameters and hyperparameters were learned on the same subjects as those used for the evaluation. Although, the need for personalized models has been recognized [1], it remains essential to evaluate generalization on new subjects.

In HAR systems, the sliding window technique has been widely used to increase accuracy [19]. However, most studies used the sliding window technique before splitting data into train and test. This results in parts of data being present in both train and test datasets, as illustrated in Figure 1. Consequently, the accuracy on the test set is not a true representation of the model's ability on unseen samples [19]. In addition to the need of splitting data into train and test sets before applying the sliding window technique, the impact of the sliding window on the generalization to unseen subjects needs to be explored.

Hence, this paper investigates the impact of the ML model architectures and the sliding window technique on the accuracy of HAR on previously unseen subjects. Two types of deep learning models, Feed Forward Neural Networks (FFNNs) and Convolutional Neural Networks (CNNs), are investigated in terms of how well they recognize activities for new subjects. Evaluation is performed using Leave-One-Subject-Out Cross Validation (LOSOCV), a modified k-fold cross-validation with each fold consisting of single subject data. Experiments show that the CNN architectures outperform FFNNs and that preprocessing including vector magnitude and sliding window improves the activity recognition accuracy. Moreover, the variability of the accuracy among subject-based folds of the cross-validation highlights the importance of using LOSOCV for the evaluation of HAR models.

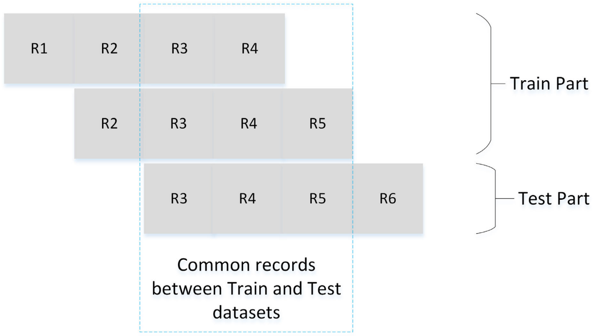

FIGURE 1. Sliding window approach before splitting the dataset into train and test.

The rest of the paper is organized as follows: Section II describes the background and Section III reviews related work. Section IV presents the methodology, Section V explains the experiments and discusses corresponding results. Finally, Section VI concludes the paper.

II. BACKGROUND

This section first provides an overview of Artificial Neural Network, in particular, Feed Forward Neural Networks and Convolutional Neural Networks in terms of structure and function.

A. FEED FORWARD NEURAL NETWORK

Artificial Neural Networks (ANNs) mimic the human brain to solve nonlinear problems. Similar to the human mind, ANNs learn to perform a task from examples without a need to be explicitly programmed.

The Feed Forward Neural Network (FFNN) is a type of ANN consisting of layers, namely, input, hidden, and output. In this network, information moves in one direction, from the input layer through the hidden layer(s) to the output layer. The input nodes receive the signals while the nodes in the output layer represent network outputs, in classification problems the outputs are different classes. During training, samples are passed forward through the network and the output of each hidden neuron j in the first hidden layer is calculated as follows:

$$y_j = f(\sum_{i=1}^{N}(h_i * w_{ij}) + b_j) \tag{1}$$

where h_i are neuron inputs, w_{ij} are the synaptic weights connecting the $i-th$ neuron in the input layer to the $j-th$ neuron in the hidden layer, b_j is a $j - th$ neuron bias, and N is the number of input neurons. Finally, f is the activation function, which is usually modeled as a *Relu* or *Sigmoid* function.

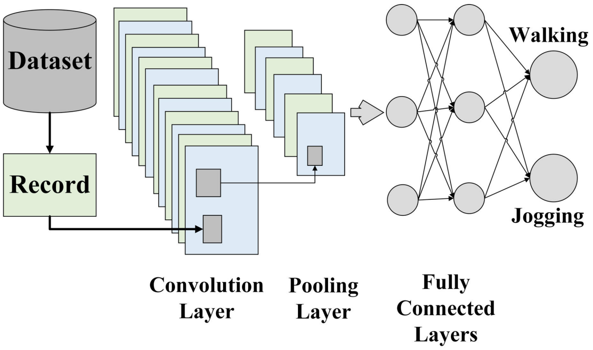

FIGURE 2. Convolutional neural networks architecture.

The outputs of the neurons in the next layer are calculated in the same way. At the output layer, the error is determined using the calculated neuron output and the expected/desired output, and the error is employed to update the weights using the backpropagation approach.

The performance of FFNN is affected by the network architecture and parameters including the number of layers, the number of neurons, and learning rates. Although approaches for determining the network architecture and parameters have being investigated [20], there are still no general rules, and the selection is usually based on the trial-and-error method [21], [22].

B. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) are a type of deep neural networks designed for data with a known grid-like topology. As the name indicates, these networks employ the convolution operation. CNNs have been used widely in image recognition [23] due to their ability capture the topology of images [24]. Then, due to its surceases with images, CNNs have been employed in other areas, such as HAR [25], and hand gesture recognition [26].

The CNN architecture consists of different layers such as input, convolution, pulling, fully connected and output. Figure 2 illustrates an example of a CNN architecture consisting of one input, convolution, pooling layer, two fully connected layers, and an output layer. The convolution layer obtains feature maps by means of element-wise multiplication of filters (kernels) and input data or output from the previous layer. After the convolution layer, the pooling layer works on each feature map to reduce the spacial size by down-sampling therefore reducing CNN computation. All the nodes in the fully connected layers are connected to all the nodes in the next layer, similarly to FFNN. Finally, at the output layer, activation functions are used to obtain outputs; for classification problems, *Softmax* is a common activation function [27]. After the calculation of error, the weights in the fully connected layers and learnable filters in the convolution layers are updated by applying a backpropagation approach and optimization algorithms such as the gradient descent.

III. RELATED WORK

This section first reviews human activity recognition works and then discusses the approaches for evaluating HAR models.

A. HAR APPROACHES

This section reviews recent works on a sensor-based HAR and focuses on the studies that used the MHEALTH dataset [28], [29] because this dataset allows us to compare results with literature.

Nguyen *et al.* [30] introduced a Feature-based and Attribute-based (FE-AT) learning approach to tackle the shortage of the labeled data in HAR datasets. They used a random oversampling approach with the goal to create a more balanced training dataset and attribute-based learning that would tackle the insufficient data problem. FE-AT variants based on three classifiers, SVM, K-NN, and Random Forest, were applied to three public datasets, MHEALTH, DailyAndSport, and RealDisp. As they are specifically interested in new activities, only a small number of samples from the target activity is used for the training. Experiments showed that FE-AT with Random Forest outperformed other approaches for new activity recognition.

Mehmood *et al.* [31] evaluated seven supervised learning algorithms in terms of HAR accuracy and classified activities included in the MHEALTH dataset into three groups, namely Ambulation, Transportation, and Exercise/fitness. Four activities, at least one from each group, were selected for evaluation: Waist Bends Forward, Standing Still, Cycling, and Jump Front and Back. Results showed that the Fuzzy Rule method with 99.79% accuracy outperformed Random Forest (99.7%), MultiLayer Perceptron Neural Network (98.96%), Decision Tree (98.58%), K-NN (95.95%), SVM (89.1%), and Naïve Bayes (53.18%) for these activities.

Chowdhury *et al.* [32] proposed a posterior-adapted class-based weighted fusion to integrate multiple accelerometers data for HAR. They first evaluated SVM, Random Forest, Binary Decision Tree, DNN, and Adaboost algorithms on PAMAP2 and MHEALTH dataset and selected SVM because of its high accuracy for further sensor fusion experiments. The proposed fusion approach with SVM outperformed the model-based and class-based weighted fusion approaches on both datasets, PAMAP2 and MHEALTH. Moreover, they investigated different accelerometer sensors configurations in terms of the number of sensors and body locations. With the proposed approach, the combinations of sensors Ankle+Wrist, Chest+Wrist, and Ankle+Chest+Wrist achieved higher accuracy than a single sensor on any location.

Zdravevski *et al.* [33] evaluated six different classifiers, namely K-NN, Logistic Regression, Naive Bayes, Random Forest, Extremely Randomized Trees, and SVM in terms of accuracy on five different datasets, DaLiAc, MHEALTH, FSP, SBHAR, and SBHARPT. In the first step, they performed feature extraction with a variety of techniques; for MHEALTH dataset, this resulted in 3232 features. Next, to reduce the number of features, they used feature importance, drift sensitivity, and diversified forward-backward feature selection. With the MHEALTH dataset, this resulted in 99.8% accuracy.

Subasi *et al.* [34] investigated a user-dependent human activity classification where an individual model is trained for each subject and evaluated on the same subject. Eight classifiers were evaluated: K-NN, ANN, SVM, C4.5, CART, Random Forest, and Rotation Forest on the MHEALTH dataset. The results show that SVM and Random Forest methods achieved the same accuracy (99.89%); however, this approach requires an individual model for each user.

Said *et al.* [35] proposed Deep Autoencoder with Low Rank Dictionary Learning (DALRD) to extract features from noisy sensor signals. Authors evaluated the proposed DALRD on two datasets and compared it to five other feature extraction techniques: Principle Component Analysis (PCA), Linear Discriminant Analysis, Robust PCA, Deep Autoencoder (DA), and Supervised Regularization-based Robust Subspace (SRRS). To examine the model robustness, they introduced different levels of random noise into the dataset. SRRS achieved the best accuracy, 98.1% on the MHEALTH dataset, with clean data, but DALRD performed better than the other approaches with noisy data.

Uddin and Hassan [36] presented a deep CNN for activity recognition from body sensors. Gaussian kernel-based PCA and Z-score normalization have been used in the preprocessing phase. For the MHEALTH dataset, the mean accuracy of the proposed CNN for all subjects was 93.9%, what was superior to Deep Belief Network (90.01%) and ANN (87.99%).

Ha *et al.* [37] also proposed a CNN for activity recognition with the MHEALTH dataset. To capture spacial and temporal dependencies among sensors, they used a 2D convolution kernel and a 2D pooling kernel. In their experiments, the proposed CNN with 2D kernels achieved better accuracy than CNN with a 1D kernel. CNN-pff [38] architecture is also based on CNN with a 2D kernel: it employs partial and full weight sharing to learn modality-specific as well as common (modality-independent) characteristics across modalities. In their experiments CNN-pff outperformed other models including HMM, SVM, Hidden conditional random fields, 1D CNN, and 2D CNN.

Finally, differences between our work and the reviewed studies can be categorized as follows:

- Nguyen *et al.* [30] and Chowdhury *et al.* [32] used LOSOCV (10 Fold) and the MHEALTH dataset. Nguyen *et al.* used oversampling method for making the dataset balanced and Chowdhury *et al.* considered only on eight different activities. Although we did apply LOSOCV (10 Fold) evaluation, we did not use oversampling and we are considering all 12 activities. Moreover, we used CNN and evaluated accuracy variability among different subjects.
- Mehmood *et al.* [31], Zdravevski *et al.* [33], Said *et al.* [35], and Ha *et al.*, [37] used hold-out validation and Ha *et al.* [38] applied a hybrid of leave-one-subject-out and hold-out validation methods. In contrast, our work uses the Leave-One-Subject-Out and Cross-Validation (10 Fold) approach.
- Subasi *et al.* [34] and Uddin and Hassan, [36] presented user-dependent (each user separately) models. On the other hand, our study considers a subject-independent model and evaluates it on the new users.

In contrast to the reviewed works, our study examines the impact of model selection and the sliding window technique on the model's ability to generalize on unseen subjects.

B. HAR EVALUATION

The recent research in the HAR field utilized different approaches to validate their models making it difficult to compare among studies even when they use the same dataset. We can classify these validation methods into four main categories.

1) HOLD OUT VALIDATION ([31], [33], [35], [37])

The dataset containing readings from all subjects is split randomly into train and test datasets. The main shortcoming of this approach is that the same person's data are in both, train and test; therefore, the results do not indicate how the model will perform on new users. Moreover, if the data are split again, the results of the model probably will change. Hold out validation can be carried out individually for each subject where it has an additional disadvantage of needed a separate model for each user.

2) K-FOLD CROSS-VALIDATION ([36])

The dataset (from one person or all people) is split into k parts; one part is reserved for evaluation and the remaining parts are used for training. The process is repeated k times, each time using a different part for evaluation. Although the results from this approach are more reliable than the results from the hold out approach, it does not evaluate accuracy for new subjects.

3) LEAVE-SUBJECT(S)-OUT HOLD OUT (LSOHO) ([38])

This is a variant of hold out validation, where one or more subjects are considered for the validation and other subjects for training the model. Although this approach evaluates on new subjects, the disadvantage is that the accuracy depends on the subject(s) selected for the evaluation.

4) LEAVE-SUBJECT(S)-OUT CROSS-VALIDATION (LSOCV) ([30], [32])

This is a variant of the k-fold cross-validation approach but with folds consisting of subjects. Similarly to LSOHO, LSOCV evaluates accuracy on new subjects, but LSOCV gives more realistic accuracy estimates, as it uses different subjects for evaluations in different folds. Moreover, LSOCV in the model selection phase should lead to more robust models. When each subject is one-fold, LSOCV becomes Leave-One-Subject-Out Cross-Validation (LOSOCV).

TABLE 1. Preprocessing methods for each scenario.

Scenario	Original Features	Magnitude Features	Sliding Window
1 (OR)	✓		
2 (MG)		✓	
3 (OR+W)	✓		✓
4 (MG+W)		✓	✓

Consequently, in this paper LOSOCV is used because it gives more realistic estimates of the model performance on new subjects.

IV. METHODOLOGY

This study explores the impact of machine learning model and data preprocessing on the system's ability to generalize on new subjects. The focus is on deep learning models as they have shown great success in recent years [1]. As the sliding window technique is commonly used for HAR due to its ability to capture temporal behaviours, impact of this technique as well as the effect of the sliding window size is examined. Additionally, vector magnitude preprocessing is considered as it reduces the number of features and, thus, simplifies the model.

Consequently, this section first discusses data preprocessing including normalization, vector magnitude, and sliding windows technique, and then describes Feed Froward Neural Networks and Convolutional Neural Networks for HAR. Finally, the evaluation methodology is described.

A. DATA PREPROCESSING

This section explains data preprocessing in preparation for neural network models. Two types of preprocessing are considered: sliding window and vector magnitude; thus, four different scenarios are designed, each one involving different preprocessing steps. Normalization is applied first for all scenarios, before any other processing. Table 1 shows the preprocessing steps used for each scenario:

- Scenario-OR: The original features are used directly without any further preprocessing.
- Scenario-MG: Features are created with the vector magnitude method.
- Scenario-OR+W: The sliding window technique is used directly on original features.
- Scenario-MG+W: Features are first created with the vector magnitude method and then, the sliding window technique is applied.

1) NORMALIZATION

Normalization is applied in order to bring all the features into a similar range and avoid dominance of large-scale features. There are different methods for normalization such as Min-Max and Z-score [39], [40]. In this paper, a Min-Max normalization, which is a common approach in HAR [41], [42], is used. The Min-Max normalization scales the numbers in a dataset to [0,1] range, which can significantly improve the accuracy of the subsequent machine learning models. The transformation function is presented as equation (2):

$$X^*_{new} = \left(\frac{X_{Old} - X_{min}}{X_{max} - X_{min}}\right) \tag{2}$$

where $X_{Old}, X_{max}, X_{min}$ are the original, maximum, and minimum values of the considered feature, respectively. X^*_{new} is the new normalized value of X_{Old}; it has values in range [0,1]. For normalization, the data is first split into train and test based on the subjects. Minimum and maximum values for the train part are extracted and used for normalizing both, train and test sets. This way, we ensure that the data from the test set is not used in the normalization process.

2) FEATURE CREATION WITH VECTOR MAGNITUDE METHOD

The data was gathered from three different wearable sensors, namely, acceleration, gyroscope, and magnetometer. These sensors were placed on the chest, left-ankle, and right-lower-arm and attached by using elastic straps. We calculate the output magnitude feature for each sensor. For instance, for acceleration from the chest sensor (A_c), we have:

With sensor-based activity recognition, data are gathered from sensors such as accelerometer, gyroscope, and magnetometer placed on different body parts including ankle, lower arm, and chest. These sensors provide three-dimensional readings corresponding to three axes:

$$[A_c = (A_x, A_y, A_z)] \tag{3}$$

The vector magnitude method takes advantage of this multi-dimensional aspect of sensor readings, and for each sensor creates a single feature representing vector magnitude:

$$Created\ Feature\ 1 : A_c = A_x^2 + A_y^2 + A_z^2 \tag{4}$$

Therefore, with vector magnitude method, equation (4), each sensor is represented with one feature reducing the number of features in 3:1 ratio.

3) SLIDING WINDOWS APPROACH

Data from HAR sensors are time series data, and, therefore, there is a dependency between the prior and current values. To capture these temporal dependencies, a well-designed feature generation mechanism is required; in HAR tasks, the sliding window technique illustrated in 3 is commonly applied for this purpose. In the figure, R1, R2, and so on are readings, each one consisting of several features obtained at the same time step. If the sliding window size is w, the first sample consists of first w readings. Next, the windows slides for s steps, and the next sample consists of readings $s + 1$ to $s + w + 1$. The Figure 3 illustrates scenario with $w = 10$ and $s = 1$.

B. DEEP LEARNING MODELS

This section describes different deep learning architectures for HAR used in this study. The two main categories of networks are considered: FFNN and CNN. For each category,

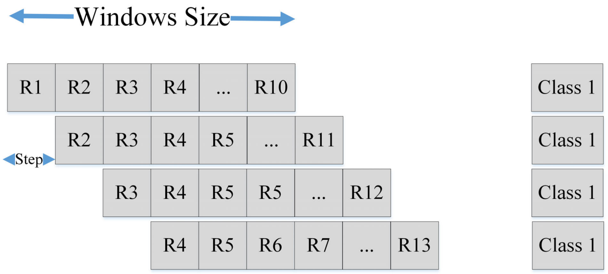

FIGURE 3. Sliding windows approach: Window size = 10 and step = 1.

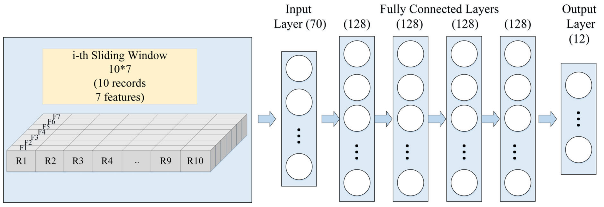

FIGURE 4. FFNN-4H for preprocessing scenario-MG+W and window w = 10.

different topologies are examined in order to determine the impact of architectures on activity recognition accuracy on new subjects.

1) FEED FORWARD NEURAL NETWORK (FFNN)

To compare different architectures, two FFNN variants are considered. Specifically, we are interested to find out how the network size impacts accuracy. In all architectures the number of inputs equals the number of features and the number of outputs corresponds to the number of classes. The two architectures are as follows:

- FFNN-4H: This FFNN architecture consists of 4 hidden layers, with 128 neurons in each hidden layer. Figure 4 shows the architecture with the sliding windows $w = 10$ and preprocessing Scenario-MG+W; thus, with 7 features, the input is 7*10.
- FFNN-6H: This FFNN architecture consists of 6 hidden layers. The first four hidden layers have 128 neurons each, and layers 5 and 6 have 64 and 32 neurons, respectively.

Each FFNN is used with each of the four preprocessing scenarios, and, for sliding window scenarios, different sliding window sizes have been evaluated.

2) CONVOLUTIONAL NEURAL NETWORK (CNN)

As with FFNN, two CNN architectures have been considered. For both, the input is a matrix of dimension *Number Of Features* × *Window Size*. As with FFNN, the outputs correspond to the classes. The two CNN architectures are:

- CNN-1C: This CNN consists of one convolution layer with 64 feature maps, one max-pooling layer (32), and one fully connected layer with 32 neurons.
- CNN-2C: This CNN includes two convolution layers, each with 64 feature maps, two max-pooling layers (32, 32), and two fully connected layers with 64 and 32 neurons, respectively. As illustrated in Figure 5, the sequence of layers are: Convolution, Max-pooling, Convolution, Max-pooling, followed by the two fully connected layers.

Both, CNN-1C and CNN-2C, can be used with one dimensional kernels (1D); we refer to them as CNN-1C-1D and CNN-2C-1D. In these methods, the kernel moves in one direction.

Moreover, when siding window technique is used in the preprocessing step, the CNN input is two-dimensional, and, therefore, two dimensional kernels (2D) can be used. We refer to the two CNN with 2D kernel as CNN-1C-2D and CNN-2C-2D. In these methods, the kernel moves in both directions, up and down.

The four combinations can be summarized as:

- CNN-1C-1D: CNN-1C architecture with 1D kernels. This can be used for all preprocessing scenarios.
- CNN-1C-2D: CNN-1C architecture with 2D kernels. This can be used only for scenarios with the sliding window: OR+W and MG+W. It cannot be used without the sliding window (OR and MG) because in those scenarios input data has only one dimension.
- CNN-2C-1D: CNN-2C architecture with 1D kernels. This can be used for all scenarios.
- CNN-2C-2D: CNN-2C architecture with 2D kernels. As CNN-1C-2D, this can be used only for the sliding window scenarios OR+W and MG+W.

C. EVALUATION

As discussed in Section III, traditional approaches for evaluating machine learning models, hold out validation and k-fold cross validation, assess the model on new samples, but samples from the same subject are in both, training and test sets. As this study is concerned with accuracy of the HAR models on new subjects, Leave-One-Subject-Out Cross-Validation (LOSOCV) is used. In LOSOCV, one subject is reserved for the evaluation and the model is trained on remaining subjects. The process is repeated each time with a different subject reserved for the evaluation and results are averaged over all folds (subjects).

Similar to LOSOCV, Leave-Subject(s)-Out Hold Out (LSOHO), also evaluates models on new subjects, but LSOHO error estimates are affected by the selection of the subjects for the test set. As it will be illustrated in experiments, LSOHO largely varies across the subjects, which demonstrates the necessity of LOSOCV for the evaluation.

To calculate the performance metrics of LOSOCV, a confusion matrix is used. A confusion matrix consists of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP and TN determine the number of samples correctly identified as positive and negative, respectively. FP and FN refer to the number of samples incorrectly identified as positive and negative, respectively.

Accuracy evaluates the proportion of the samples correctly classified. It is a well-suited metric for the classification evaluation when the dataset is balanced or

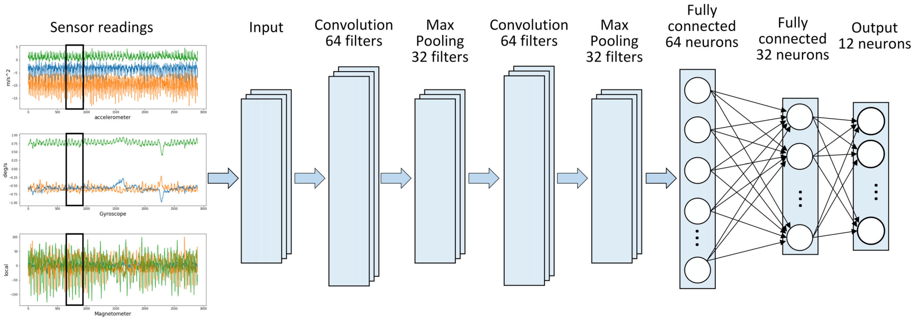

FIGURE 5. CNN-2C architecture used.

approximately balanced. Consequently, in addition to accuracy, this study uses precision, recall, and F1 score. These metrics are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

Precision quantifies the number of positive class predictions that actually belong to the positive class. Recall quantifies the number of positive class predictions made out of all positive examples in the dataset. Finally, F1 score is the harmonic mean of the precision and recall.

V. DATA AND RESULTS

This section first introduces the dataset and experiments and the presents the results followed by the discussion of the findings are.

A. DATA AND EXPERIMENTS

1) DATA

The experiments were carried out with the MHEALTH (Mobile Health) dataset. This dataset includes recordings of body motions and vital signs for ten individuals while preforming various activities. Recorded movement data is accompanied with twelve activity labels such as ‘Standing still,’ ‘Sitting and relaxing,’ ‘Lying down,’ ‘Walking,’ and so on. Accelerometer, gyroscope, and magnetometer sensors on subject’s chest, right wrist, and left ankle measured the movement experienced by various body parts, namely, acceleration, rate of turn, and magnetic field orientation. An additional sensor on the chest recorded ECG signals which can be used for heart monitoring, but these data are not used here for HAR as they do not directly relate to human motions.

At each reading time step, each of the three sensors, accelerometer, gyroscope and magnetometer, records three values corresponding to three axes. All three sensors are mounted on an ankle and an arm, and only an accelerometer is mounted on the chest; this makes a total of 21 readings for each time step.

TABLE 2. Number of input features for each scenario.

	Window	Number of Initial Features	Number of Input Features for ML
Scenario-OR		21	21
Scenario-MG		7	7
Scenario-OR+W	5	21	105
	10	21	210
	15	21	315
Scenario-MG+W	10	7	70
	20	7	140
	50	7	350

2) EXPERIMENTS

As discussed in Section IV, four scenarios (OR, MG, OR+W, MG+W) and six models (FFNN-4H, FFNN-6H, CNN-1C-1D, CNN-1C-2D, CNN-2C-1D, CNN-2C-2D) are considered. The number of input features for each of the considered scenarios is illustrated in Table 2. Columns ’Number of Initial Features’ and ’Number of Input Features for ML’ indicates the number of features before and after applying the sliding a window technique.

As can be seen from the table, with the two scenarios that include the sliding window technique, OR+W and MG+W, different window sizes are considered. For Scenario-OR+W, considered window sizes are 5, 10, and 15 and for Scenario-MG+W, window sizes are 10, 20, and 50. The sliding window sizes are larger for Scenario-MG+W than for OR+W because in MG+W there are only 7 features in contrast to 21 in OR+W. It is expected that fewer features will need larger windows to adequately capture movement patterns.

Considering different window sizes results in eight scenarios. The six scenarios with the sliding window (OR+W and MG+W with three different window sizes) are applied with each of the six DL models, while the non-window scenarios (OR and MG) are applied with only four DL models as they are not applicable for CNNs with 2D as discussed in subsection IV-B2. This results in the total of 6 × 6 + 2 × 4 = 44 experiments.

All experiments were implemented in Python. For Deep Neural Networks, the ’scikit-learn’ library was used [43].

TABLE 3. Accuracy for different scenarios and models.

Scenario	Window Size	FFNN		CNN			
		4H	6H	1C		2C	
				1D	2D	1D	2D
OR		62.4	59.6	**68.4**	—	62.1	—
MG		59.8	60.7	62	—	**63.4**	—
OR+W	5	72.8	70.3	78.8	**80.7**	78.7	78.9
	10	65.7	67	74.5	74.2	67.2	69.1
	15	67.3	68.5	72.6	75.2	69.4	69.9
MG+W	10	76	73.7	78.8	77.5	82.6	80.4
	20	75.2	74.9	79.1	79.8	78.3	77.3
	50	81.1	80.9	82.8	83.9	**85.1**	81.7

The experiments were executed on a computer with Ubuntu OS, AMD Ryzen 4.20 GHz processor, 128 GB DIMM RAM, and four NVIDIA GeForce RTX 2080 Ti 11GB graphics cards. Training the proposed DNNs is computationally expensive; hence, GPU acceleration was utilized. Nevertheless, once the model is trained, it does not need significant resources, and CPU processing is adequate.

B. RESULTS

The results were assessed based on designed scenarios, methods, and subjects. Finally, we compare the results for two validation approaches, namely, 10-Fold Cross-Validation and LOSOCV.

1) ACCURACY FOR DL MODELS AND SCENARIOS

This subsection compares the results obtained by different models for each of the scenarios using LOSOCV. Table 3 shows the average accuracy for all cross-validation folds. The accuracy of the best model for each of the four main scenarios (OR, MG, OR+W, MG+W) is indicated with bold values in the table. Note that the numbers here are much lower than in many studies [33]; however, this is not caused by the model itself, but by the evaluation approach as it will be demonstrated later in this section.

For scenarios OR and MG, the highest accuracy is obtained with models CNN-1C-1D and CNN-2C-1D, respectively. For those two scenarios, the table does not present results for CNN-1C-2D and CNN-2C-2D because 2D convolution can only be used when window sliding technique is used, as discussed in Subsection IV-B2.

For Scenario-OR+W, the best result, 80.7% accuracy, has been obtained by the CNN-1C-2D model with sliding window size 5. From the table, it can be observed that as the window size increases from 5 to 10 and 15, the accuracy of each model decreases.

For Scenario-MG+W, the best model was CNN-2C-1D, with 85.1% accuracy. This value was the highest accuracy for all scenarios and all models; thus, CNN-2C-1D with vector magnitude and the sliding window size 50 was the best approach. In Scenario-MG+W, as the window size increases from 20 to 50, the accuracy improves for all models; however, the same pattern does not hold when window size increases from 10 to 20.

CNN-2C-2D results are superior to CNN-2C-1D results for the Scenario-OR+W; however, for Scenario-MG+W, the opposite pattern is observed. Moreover, in terms of CNN architecture comparison (1C vs 2C), for Scenario-OR+W, CNN-1C (1D or 2D) obtained better results than CNN-2C (1D or 2D). For Scenario-MG+W, CNN-2C outperformed CNN-1C sometimes, but not all window sizes and models.

As expected, adding the sliding window increases accuracy: comparing scenarios OR with OR+W and scenarios MG with MG+W, it can be observed that for all models adding the sliding window increases accuracy.

This subsection compared results based not only on accuracy but also on other metrics including precision, recall, and F1 Score. Regardless of the metric used, the best model for each scenario remained the same.

2) PERFORMANCE ANALYSIS FOR DIFFERENT SUBJECTS

Table 3 identifies the best model for each scenario, and Table 4 analyzes the performance of the best model on individual subjects. The first column includes the scenario and the best model for that scenario. Rows for subjects 1 to 10 represent the folds of the subject-based cross validation: for example, for subject 1 row, the model is trained using subjects 2-10 and evaluated on subject 1. It can be observed that the accuracy of the same model varies greatly among subjects illustrating the need to use LOSOCV as opposed to Leave-Subject(s)-Out Hold Out (LSOHO) in order to capture variability among subjects. Also, the standard deviation is different across models, which means that some models are more consistent than others. This shows that if a single generic model will be used for all users, the standard deviation should be considered when selecting the model.

In terms of accuracy, for Scenario-OR with CNN-1C-1D model, subjects 4, 3, and 9 have the highest and subjects 6, 8, and 1 have the lowest accuracy. For Scenario-OR+W with CNN-1C-2D model, subjects 3, 2, and 5 and 10 have the highest and subjects 6, 1, and 7 have the lowest accuracy. The patterns for scenarios MG and MG+W are similar; however, the sequence of subjects is different. For the Scenario-MG, the best results are for subjects 10, 3, and 9, and for Scenario-MG+W, the order is for 9, 3, and 10.

It can be observed that subject 3 appears between the best results for all models and scenarios. Subjects 9 and 10 are between the best results for three out of four scenarios. Subject 6 is between the worst performing for all models and scenarios, while subjects 8 and 2 are also often among the worst (three out of four and two out of four scenarios, respectively). The similar subjects appearing among the best/worst in terms of accuracy, irrelevant of the DL model, could be caused by the similarity/dissimilarity of the target (validation) subject to those present in the training set. For example, subject 3 being always among the best could be caused by its high similarity to other subject(s). In contrast, subject 6 may be

TABLE 4. Subject-level accuracy metrics for each scenario.

Scenario	Subject	Accuracy	Recall	Precision	F1 Score
Scenario-OR (Best: CNN-1C-1D)	1	57.3	73.7	73.3	72.3
	2	70	78.1	75.3	76.5
	3	84	89.1	87.2	88.1
	4	90.4	96	95.3	95
	5	73.4	85.1	85.6	85.2
	6	41.4	62.2	59.5	52.5
	7	60	70.7	69.9	70
	8	45.5	60.4	61.6	59.3
	9	83.3	89.7	88.1	88.5
	10	78.9	95.3	85.4	85
Average		**68.4**	**80**	**78.1**	**77.2**
Standard Deviation		**15.9**	**12.3**	**11.4**	**13**
Scenario-MG (Best: CNN-2C-1D)	1	60.6	75.7	75	74
	2	53.8	64.1	67.7	65.3
	3	80.2	92.1	86.4	88.4
	4	66.4	81.5	88.1	84.5
	5	63.7	76.1	76.4	75.1
	6	51.1	78	68.8	69.2
	7	61.3	75.3	72.2	69.9
	8	34.8	56.2	49.9	51.9
	9	78.8	91.5	85.5	87.7
	10	83.2	91	86.4	88.2
Average		**63.4**	**78.1**	**75.6**	**75.4**
Standard Deviation		**14.1**	**11.2**	**11.3**	**11.4**
Scenario-OR+W (Best: CNN-1C-2D)	1	70.1	78.8	82	79.4
	2	89.6	91.4	91	91.2
	3	90.6	91.5	91.2	91.3
	4	81.7	90	84.5	84.5
	5	85.7	88.9	89.9	89.3
	6	66.7	80.1	87.5	80.8
	7	74.8	82.9	78.6	79.3
	8	76.8	77.1	87.6	77.9
	9	85.2	89.8	88.1	88.4
	10	85.7	95.7	93.4	92.6
Average		**80.7**	**86.6**	**87.4**	**85.5**
Standard Deviation		**7.8**	**6**	**4.3**	**5.4**
Scenario-MG+W (Best: CNN-2C-1D)	1	88	91	89.2	89.9
	2	72.7	74.6	78.9	76.1
	3	96.3	99.8	96.7	98
	4	82.8	91.8	99.3	94.3
	5	88.2	86.8	89.8	87.8
	6	76.5	90.4	80.6	81.1
	7	89.7	86.4	91.2	88.1
	8	64	68.6	74.2	71
	9	97.3	98.8	98.5	98.6
	10	95.3	97.4	97.8	97.5
Average		**85.1**	**88.5**	**89.6**	**88.2**
Standard Deviation		**10.5**	**9.6**	**8.5**	**9.1**

more different than the others, therefore resulting in lower accuracy.

Comparing scenarios MG and MG+W, we can observe that for every single subject the Scenario-MG+W works better than Scenario-MG. The same pattern occurs for scenarios OR and OR+W for all the subjects except subject 4. This indicates that even on the subject level, adding sliding window in the preprocessing improves the accuracy.

In terms of recall, subject 10 accuracy appears as one of the best results and subjects 3 and 9 are between the best results in three out of four scenarios. Subject 8 exhibits the lowest accuracy for all scenarios.

In terms of precision, subjects 3, 4, 9, and 10 show higher accuracy than others while subjects 8 and 6 appear the most often among the lowest accuracy group. Finally, considering F1 score, subject 3 appears among the best performers for all scenarios and subject 8 consistently belongs to the low accuracy group.

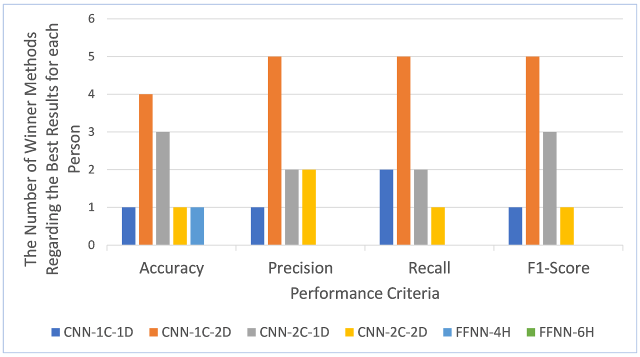

FIGURE 6. Subject-level best method: The number of subjects for which the model is the best.

For all performance metrics, accuracy, precision, recall, and F1 score, subject 3 is among the group with high accuracy for all scenarios. Subjects 6 and 8 appear often in low accuracy group for all metrics. As already mentioned when discussing accuracy, metrics variability among subjects is the result of different levels of similarity.

From table 4, it can be observed that the best model for each subject is not the same; for example, for subject 1, the best model is CNN-2C-1D, and for subject 2 it is CNN-1C-2D. Figure 6 shows the number of subjects for which the model is the best. Considering the accuracy metric, CNN-1C-2D was the best model for 4 subjects and CNN-2C-1D for 3 subjects. FFNN-6H was not the best model for any subjects; therefore, the figure does not show the bar for this model. Although the overall best model was CNN-2C-1D as shown in Table 4, CNN-1C-2D was better for more subjects as illustrated in Figure 6.

3) THE EFFECT OF THE WINDOW SIZE ON ACCURACY

The window size impacts the accuracy of the DL model as can be seen from Table 4. Here, we further investigate the impact of window size. Figure 7 shows the average accuracy for each model for Scenario-OR+W. It can be observed that as the window size increases from 5 to 10, the accuracy for all methods except FFNN-6H decreases. CNN-1C-2D experiences significant decline from 80.7% to 67% while FFNN-6H accuracy increases from 70.3% to 74.5%. The opposite pattern happens when the window size is increased from 10 to 15: for all methods except FFNN-6H, the accuracy increases slightly. Nevertheless, the methods' accuracy with sliding window 15 is still lower than with sliding window 5 for all methods but FFNN-6H.

While Figure 7 shows the accuracy for Scenario-OR+W, Figure 8 does the same for Scenario-MG+W. As the window size increases from 10 to 20, the accuracy for FFNN-4H, CNN-2C-1D, and CNN-2C-2D decreases and for FNN-6H, CNN-1C-1D, and CNN-1C-2D increases. As the window further increases to 50, the accuracy for all models increases. It is interesting to note that with the increase of the window size, the differences in accuracy among models reduces. For example, at window size (10), the best and worst accuracy

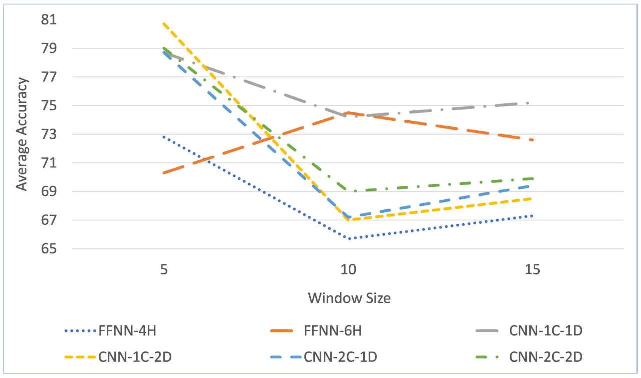

FIGURE 7. The impact of the window size on accuracy: Scenario-OR+W.

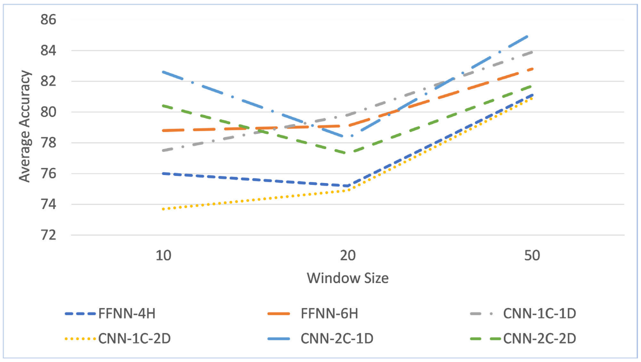

FIGURE 8. The impact of the window size on accuracy: Scenario-MG+W.

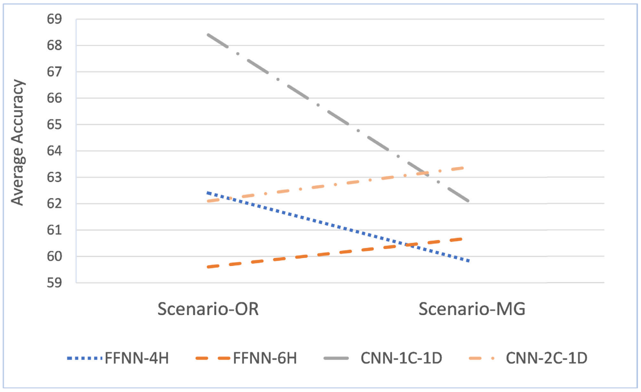

FIGURE 9. Comparison of scenarios OR and MG.

are 82.6% (CNN-2C-1D) and 73.7% (CNN-1C-2D), respectively. But for window size (50), the best and worst accuracy are 85.1% (CNN-2C-1D) and 80.9% (CNN-1C-2D), respectively.

4) PREPROCESSING IMPACT ON ACCURACY

Here preprocessing is investigated with respect to how it affects accuracy. First, accuracy of scenarios OR and MG is compared in Figure 9. It can be observed that as features are reduced from 21 (Scenario-OR) to 7 (Scenario-MG), the accuracy of more complex models, FFNN-6H and CNN-2C-1D, increases while the accuracy of simpler models, FFNN-4H and CNN-1C-1D, decreases.

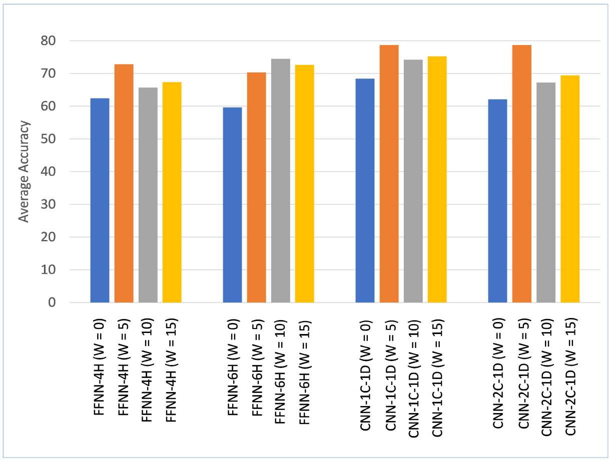

FIGURE 10. Comparison of scenarios OR and OR+W.

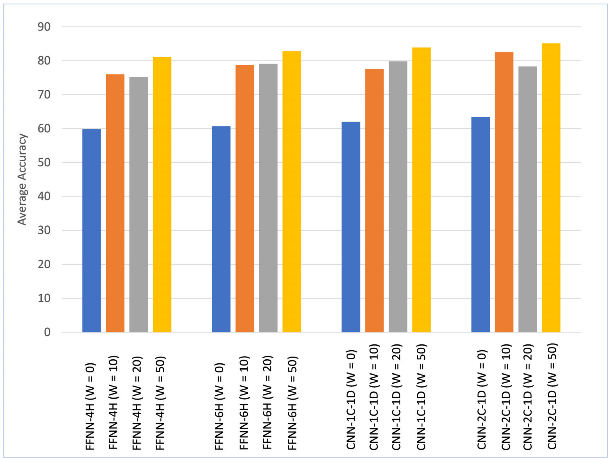

FIGURE 11. Comparison of scenarios MG and MG+W.

Next, Figure 10 shows the comparison between scenarios OR and OR+W based on the accuracy. Because 2D convolution is not applicable for Scenario-OR, the figure does not include 2D models: CNN-1C-2D and CNN-2C-2D. It can be seen that for all models the accuracy increases when the sliding window technique is used. The difference between the accuracy of Scenario-OR and Scenario-OR+W with window size 5 is more than ten percent, which illustrates that even small window size has a significant impact on the accuracy.

Scenarios MG and MG+W are compared in Figure 11. As for Scenario-MG, 2D convolution is not applicable, 2D models are not included in this figure. Again, the accuracy increases when the window sliding technique is used. For all window sizes, the accuracy of MG+W is more than ten percent higher than the accuracy of MG, and for window size 50, the MG+W accuracy is 20% higher than the MG accuracy.

5) COMPARISON OF K-FOLD CROSS-VALIDATION AND LOSOCV

As already noted, evaluation with LOSOCV exhibits lower accuracy than the traditional k-fold cross-validation (CV)

TABLE 5. Confusion matrix for 10-fold cross-validation and LOSOCV.

		Predicted class											
		1	2	3	4	5	6	7	8	9	10	11	12	
10-Fold Cross-Validation	Actual class	1	30230	0	0	0	0	0	0	0	0	0	0	0
		2	0	29879	351	0	0	0	0	0	0	0	0	0
		3	0	0	30230	0	0	0	0	0	0	0	0	0
		4	0	0	0	30229	1	0	0	0	0	0	0	0
		5	0	0	0	3	30226	0	0	0	0	1	0	0
		6	0	0	0	0	0	27771	2	52	0	0	0	0
		7	0	0	1	0	0	1	28949	0	0	0	0	0
		8	0	0	0	2	3	30	0	28810	2	0	0	0
		9	0	0	0	1	7	0	0	0	30222	0	0	0
		10	0	0	0	0	0	0	0	0	0	30192	32	6
		11	0	0	0	0	0	0	0	0	0	11	30218	1
		12	0	0	0	0	2	0	0	0	0	2	2	9846
LOSOCV	Actual class	1	20559	60	6905	0	0	0	2706	0	0	0	0	0
		2	3023	18139	9068	0	0	0	0	0	0	0	0	0
		3	6156	6091	17983	0	0	0	0	0	0	0	0	0
		4	0	0	0	30037	97	0	0	0	19	62	0	15
		5	0	13	15	749	29124	29	0	121	170	2	0	7
		6	77	124	0	0	36	25178	368	2016	26	0	0	0
		7	94	1	0	0	0	313	28516	27	0	0	0	0
		8	31	0	0	2	125	2525	41	25945	177	1	0	0
		9	1	0	1	87	1126	11	0	615	28324	28	0	37
		10	0	0	0	0	11	0	0	0	0	26851	3364	4
		11	0	0	0	0	0	0	0	0	0	2985	27244	1
		12	0	0	0	2	0	0	0	0	0	483	9	9358

where data points from the same subject can be in both training and test sets. However, LOSOCV gives the estimate of error for the new subject while k-fold CV gives the estimate for the subjects present in the training set.

To examine the impact of the evaluation on the performance metrics, we compare the k-fold CV and LOSOCV using the same model and the same preprocessing.

For LOSOCV, the overall best model as shown in Table 3 was CNN-2C-1D with the Scenario-MG+W, window size = 50; thus, this model is used for comparison with a k-fold CV. Specifically, a 10-fold CV is considered. For the 10-fold CV, after data preprocessing including vector magnitude and the sliding window technique, the data are split randomly into 10 parts. As the split is random, same subject data may appear in training and test sets. One part is reserved for testing while remaining parts are used to train the model. The process is repeated for each fold: the performance metrics for each fold are shown in Table 6. The average accuracy with a 10-fold CV was 99.85%; however, the accuracy of the same model with LOSOCV was 85.1% (Table 3). This demonstrates the necessity of using LOSOCV when the objective is to estimate accuracy of the model for new subjects.

To further compare the traditional cross-validation (10-fold CV) with LOSOCV, Table 5 shows the aggregation of 10 confusion matrices from 10 folds for the two approaches. The accuracy corresponding to this table for a 10-fold CV is 99.85% (the average accuracy in table 6), and for LOSOCV, it is 85.1% (Table 3). Consequently, misclassification for 10-fold CV is very low, for most classes zero or close to zero while for LOSOCV, the number of misclassified samples for some activities (classes) is significantly higher. This higher LOSOCV misclassification is caused by differences among subjects in the train and test datasets. Still, for some pairs of classes, LOSOCV misclassification is zero, demonstrating that distinction between those classes generalizes well for new subjects. As the overall accuracy with LOSOCV is lower than with traditional 10-fold CV, there is a need to improve performance for new subjects and/or develop HAR personalization.

TABLE 6. Performance metrics for each fold of the 10-fold cross-validation.

Fold	Accuracy	Recall	Precision	F1 Score
1	99.95	99.97	99.98	99.98
2	98.93	99.06	99.98	99.49
3	99.96	99.97	99.99	99.98
4	99.96	99.97	99.99	99.97
5	99.93	99.95	99.98	99.97
6	99.97	99.98	99.98	99.98
7	99.99	99.99	99.99	99.99
8	99.84	99.9	99.94	99.92
9	99.99	99.99	99.99	99.99
10	99.96	99.99	99.98	99.98
Average	99.85	99.88	99.98	99.93

With the traditional k-fold CV, the CNN-2C-1D model with Scenario-MG+W, window size = 50, demonstrated performance metrics (accuracy, precision, recall, F1 Score, and confusion matrix) comparable to those reported in literature [36] ; however, these metrics greatly differ from LOSOCV which estimates the performance of the model for new subjects. On the other hand, LOSOCV estimates the performance of the model for new subjects and, therefore, should be used for real-world applications as it is not possible to include each potential user data in the training set.

C. DISCUSSION

The main objective of this work is to evaluate the impact of model selection and preprocessing on the ability of the ML model to classify activities for new users. Consequently, LOSOCV was used for the evaluation. Comparison of the results obtained with LOSOCV (Table 3) and the 10-fold cross-validation (Table 6) for the same model shows that the two lead to very different estimates. The accuracy for the CNN-2C-1D model with Scenario-MG+W ($w = 50$) was 99.85% when evaluated with traditional 10-fold cross-validation and only 85.1% with LOSOCV evaluation. As LOSOCV ensures that different subjects are used for training and testing, LOSOCV estimates are closer to what can be expected for new users. Such significant differences also indicate the need to develop a new model capable of achieving higher accuracy for new users. A possible way of achieving this is by personalizing the model and exploring similarities among users [1].

As expected, using the sliding window technique increased the accuracy of each model, as illustrated in Figures 10 and 11. However, increasing the window size does not necessary lead to increase in accuracy. As shown in Figure 10, a larger window size may result in accuracy decrease.

When the number of features is reduced, such as in the case of vector magnitude shown in Figure 9 where the number of features is reduced, a more complex model is needed in order to capture the patterns. It can be observed that reducing features from 21 (Scenario-OR) to 7 (Scenario-MG), the accuracy of the more complex models (FFNN-6H and CNN-2C) increases while the accuracy of more simple models (FFNN-4H and CNN-1C) reduces.

Overall, the experiments demonstrated the importance of using LOSOCV for estimating the performance of an ML model for new users and the risks of accuracy overestimates with traditional k-fold cross-validation. CNN with two convolutional layers and 1D filters archived the highest accuracy. Preprocessing with vector magnitude and sliding window can improve the performance (Table 3), but selecting the window size remains a challenge as it is dependent on the model (Figures 10 and 11). As CNNs are sensitive to hyperparameter choice, further hyperparameter tuning has the potential to improve accuracy.

VI. CONCLUSION

Human activity recognition is becoming a big trend in some industries, but it is a challenging research area. Deep learning and pre-processing methods have been successfully used in recognizing patterns.

This paper presented four different scenarios designed to improve accuracy for human activity recognition. Results show that LOSOCV is a rigid criterion for evaluation models in comparison to Cross-Validation or Hold-Out approaches. Moreover, the sliding window technique can improve performance criteria; however, finding the best window size is a crucial issue. Using only the vector magnitude method cannot improve the performance, but using a hybrid of vector magnitude and sliding window approaches can improve results considerably. In the MHEATH dataset, Scenario-MG+W (w = 50) via CNN-2C-1D, we could reach 85.1% accuracy with LOSOCV. On the other hand, the accuracy for the same scenario and method with the 10 Fold Cross-Validation was 99.85%, which means that it is necessary to work on the design of architectures of methods and tune them based on LOSOCV.

Training CNNs is computationally expensive and applying LOSOCV makes the training even more time consuming as it requires repetition of the process with different subjects in the test set. Nevertheless, LOSOCV provides more realistic estimates of the HAR accuracy for new users. The vector magnitude approach also has a disadvantage of eliminating the sign of the signal.

Future work will evaluate the presented approaches with different data sets and explore improving accuracy of HAR for new users through personalization.

REFERENCES

[1] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, “On the personalization of classification models for human activity recognition,” *IEEE Access*, vol. 8, pp. 32066–32079, 2020.

[2] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, “Mobile health monitoring system based on activity recognition using accelerometer,” *Simul. Model. Pract. Theory*, vol. 18, no. 4, pp. 446–455, Apr. 2010.

[3] H. Storf, T. Kleinberger, M. Becker, M. Schmitt, F. Bomarius, and S. Prueckner, “An event-driven approach to activity recognition in ambient assisted living,” in *Proc. Eur. Conf. Ambient Intell.*, 2009, pp. 123–132.

[4] K. Partridge and B. Begole, “Activity-based advertising,” in *Proc. Pervas. Advertising*, 2011, pp. 83–101.

[5] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, “The opportunity challenge: A benchmark database for on-body sensor-based activity recognition,” *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, Nov. 2013.

[6] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, Jan. 2014.

[7] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[8] G. A. Oguntala, R. A. Abd-Alhameed, N. T. Ali, Y.-F. Hu, J. M. Noras, N. N. Eya, I. Elfergani, and J. Rodriguez, “SmartWall: Novel RFID-enabled ambient human activity recognition using machine learning for unobtrusive health monitoring,” *IEEE Access*, vol. 7, pp. 68022–68033, 2019.

[9] C. Harito, L. Utari, B. R. Putra, B. Yuliarto, S. Purwanto, S. Z. J. Zaidi, D. V. Bavykin, F. Marken, and F. C. Walsh, “Review—The development of wearable polymer-based sensors: Perspectives,” *J. Electrochemical Soc.*, vol. 167, no. 3, Feb. 2020, Art. no. 037566.

[10] Y.-S. Lee and S.-B. Cho, “Activity recognition using hierarchical hidden Markov models on a smartphone with 3D accelerometer,” in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2011, pp. 460–467.

[11] T. Phan, “Improving activity recognition via automatic decision tree pruning,” in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. Adjunct Publication*, 2014, pp. 827–832.

[12] 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,” in *Proc. Int. Workshop Ambient Assist. Living*, 2012, pp. 216–223.

[13] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proc. 6th Int. Joint Conf. Auto. Agents Multiagent Syst.*, 2007, pp. 1–8.

[14] M. A. Ayu, S. A. Ismail, A. F. A. Matin, and T. Mantoro, "A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognition," *Procedia Eng.*, vol. 41, pp. 224–229, Jan. 2012.

[15] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016, *arXiv:1604.08880*. [Online]. Available: http://arxiv.org/abs/1604.08880

[16] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de-Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors J.*, vol. 19, no. 16, pp. 6872–6882, Aug. 2019.

[17] E. Batbaatar, M. Li, and K. H. Ryu, "Semantic-emotion neural network for emotion recognition from text," *IEEE Access*, vol. 7, pp. 111866–111878, 2019.

[18] L. Sehovac and K. Grolinger, "Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention," *IEEE Access*, vol. 8, pp. 36411–36426, 2020.

[19] A. Jordao, A. C. Nazare, Jr., J. Sena, and W. Robson Schwartz, "Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art," 2018, *arXiv:1806.05226*. [Online]. Available: http://arxiv.org/abs/1806.05226

[20] H. Faris, S. Mirjalili, and I. Aljarah, "Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2901–2920, Oct. 2019.

[21] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Hillsdale, NJ, USA: Psychology press, 2013.

[22] S. Haykin, *Kalman Filtering and Neural Networks*, vol. 47. Hoboken, NJ, USA: Wiley, 2004.

[23] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10. Cambridge, MA, USA: MIT Press, 1995.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[25] M.-O. Mario, "Human activity recognition based on single sensor square HV acceleration images and convolutional neural networks," *IEEE Sensors J.*, vol. 19, no. 4, pp. 1487–1498, Feb. 2019.

[26] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A hand gesture recognition sensor using reflected impulses," *IEEE Sensors J.*, vol. 17, no. 10, pp. 2975–2976, May 2017.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[28] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: A novel framework for agile development of mobile health applications," in *Proc. Int. Workshop Ambient Assist. Living*, 2014, pp. 91–98.

[29] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *Biomed. Eng. OnLine*, vol. 14, no. 2, p. S6, 2015.

[30] L. T. Nguyen, M. Zeng, P. Tague, and J. Zhang, "Recognizing new activities with limited training data," in *Proc. ACM Int. Symp. Wearable Comput.*, 2015, pp. 67–74.

[31] A. Mehmood, A. Raza, A. Nadeem, and U. Saeed, "Study of multi-classification of advanced daily life activities on shimmer sensor dataset," *Int. J. Commun. Netw. Inf. Secur.*, vol. 8, no. 2, p. 86, 2016.

[32] A. K. Chowdhury, D. Tjondronegoro, V. Chandran, and S. G. Trost, "Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometer data," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 3, pp. 678–685, May 2018.

[33] E. Zdravevski, P. Lameski, V. Trajkovik, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo, and N. Garcia, "Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering," *IEEE Access*, vol. 5, pp. 5262–5280, 2017.

[34] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb, "IoT based mobile healthcare system for human activity recognition," in *Proc. 15th Learn. Technol. Conf. (L&T)*, Feb. 2018, pp. 29–34.

[35] A. B. Said, A. Mohamed, T. Elfouly, K. Abualsaud, and K. Harras, "Deep learning and low rank dictionary model for mHealth data classification," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf.*, Jun. 2018, pp. 358–363.

[36] M. Z. Uddin and M. M. Hassan, "Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network," *IEEE Sensors J.*, vol. 19, no. 19, pp. 8413–8419, Oct. 2019.

[37] S. Ha, J.-M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 3017–3022.

[38] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 381–388.

[39] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," *IEEE Access*, vol. 5, pp. 3095–3110, 2017.

[40] R. Xi, M. Li, M. Hou, M. Fu, H. Qu, D. Liu, and C. R. Haruna, "Deep dilation on multimodality time series for human activity recognition," *IEEE Access*, vol. 6, pp. 53381–53396, 2018.

[41] J. Suto, S. Oniga, and P. P. Sitar, "Comparison of wrapper and filter feature selection algorithms on human activity recognition," in *Proc. 6th Int. Conf. Comput. Commun. Control (ICCCC)*, May 2016, pp. 124–129.

[42] Y.-L. Hsu, S.-C. Yang, H.-C. Chang, and H.-C. Lai, "Human daily and sport activity recognition using a wearable inertial sensor network," *IEEE Access*, vol. 6, pp. 31715–31728, 2018.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

DAVOUD GHOLAMIANGONABADI received the B.Sc. degree in applied math from Ferdowsi University and the M.Sc. degree in industrial engineering-system and productivity management from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. He is currently pursuing the second master's degree in software engineering. His current research interests include machine learning, deep learning, neural networks, and data analytics.

NIKITA KISELOV is currently pursuing the bachelor's degree in computer science (the IoT specialization) with Lviv Polytechnic National University, Ukraine. He completed MITACS Summer internship program at Western University, Canada, where he worked on human activity recognition. His current research interests include deep learning, the IoT, computer vision, and neural networks.

KATARINA GROLINGER (Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of Zagreb, Croatia, and the M.Eng. and Ph.D. degrees in software engineering from Western University, Canada. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Western University. She has been involved in the software engineering area in academia and industry, for over 20 years. Her current research interests include machine learning, sensor data analytics, data management, and the IoT.