CSITime: Privacy-Preserving Human Activity Recognition Using WiFi Channel State Information

Santosh Kumar Yadav[a,b,c], Siva Sai[d], Akshay Gundewar[d], Heena Rathore[e], Kamlesh Tiwari[d], Hari Mohan Pandey[f], Mohit Mathur[c]

[a] *Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, UP-201002, India*
[b] *Cyber Physical System, CSIR-Central Electronics Engineering Research Institute (CEERI), Pilani-333031, India*
[c] *DeepBlink LLC, 30 N Gould St Ste R, Sheridan, WY 82801, United States.*
[d] *Department of CSIS, Birla Institute of Technology and Science Pilani, Pilani Campus, Rajasthan-333031, India*
[e] *Department of Computer Science at the University of Texas, San Antonio, United States.*
[f] *Department of Computer Science, Edge Hill University, Lancashire, United Kingdom*

Abstract

Human activity recognition (HAR) is an important task in many applications such as smart homes, sports analysis, healthcare services, *etc.* Popular modalities for human activity recognition involving computer vision and inertial sensors are in the literature for solving HAR, however, they face serious limitations with respect to different illumination, background, clutter, obtrusiveness, and other factors. In recent years, WiFi channel state information (CSI) based activity recognition is gaining momentum due to its many advantages including easy deployability, and cost-effectiveness. This work proposes CSITime, a modified InceptionTime network architecture, a generic architecture for CSI-based human activity recognition. We perceive CSI activity recognition as a multi-variate time series problem. The methodology of CSITime is threefold. First, we pre-process CSI signals followed by data augmentation using two label-mixing strategies - mixup and cutmix to enhance the neural network's learning. Second, in the basic block of CSITime, features from multiple convolutional kernels are concatenated and passed through a self-attention layer followed by a fully connected layer with Mish activation. CSITime network consists of six such blocks followed by a global average pooling layer and a final fully connected layer for the final classification. Third, in the training of the neural network, instead of adopting general training procedures such as early stopping, we use one-cycle policy and cosine annealing to monitor the learning rate. The proposed model has been tested on publicly available benchmark datasets, *i.e.*, ARIL, StanWiFi, and SignFi datasets. The proposed CSITime has achieved accuracy of 98.20%, 98%, and 95.42% on ARIL, StanWiFi, and SignFi datasets, respectively, for WiFi-based activity recognition. This is

Email addresses: santosh.yadav@pilani.bits-pilani.ac.in (Santosh Kumar Yadav), f20170779@pilani.bits-pilani.ac.in (Siva Sai), f20180240@pilani.bits-pilani.ac.in (Akshay Gundewar), heena.rathore@ieee.org (Heena Rathore), kamlesh.tiwari@pilani.bits-pilani.ac.in (Kamlesh Tiwari), Pandeyh@edgehill.ac.uk (Hari Mohan Pandey), mohit2308@gmail.com (Mohit Mathur)

an improvement on state-of-the-art accuracies by 3.3%, 0.67%, and 0.82% on ARIL, StanWiFi, and SignFi datasets, respectively. In lab-5 users' scenario of the SignFi dataset, which has the training and testing data from different distributions, our model achieved accuracy was 2.17% higher than state-of-the-art, which shows the comparative robustness of our model.

Keywords: Human activity recognition, WiFi Channel State Information, Time series classification, Data augmentation

1. Introduction

Human activity recognition is a widely researched field owing to numerous applications such as healthcare services [1], context awareness [2], driver behavior analysis [3], abnormal human motion detection [1], sports analysis [4], smart-homes [5], and IoT applications [6]. For human activity recognition, computer vision and inertial sensors-based approaches are widely used. However, there are a few limitations associated with both of these modalities. Camera-based activity recognition poses many concerns including privacy, does not work well in poor lighting conditions, also the subject must be in the visible range of the camera, and cannot recognize actions in the presence of obstacles or walls. Wearable inertial sensors may be inconvenient, obtrusive, and expensive to maintain for the user [7]. Since the wearable inertial sensors require the user to carry additional tracking equipment, it makes them inconvenient.

WiFi-based methods are emerging solutions for human activity recognition resolving above mentioned limitations [8]. WiFi-based activity recognition has several advantages when compared to its counterparts such as easy deployability due to its easy availability, and low cost [8]. There are three broad categories of signals in WiFi that can be used for activity recognition - received signal strength indicator (RSSI), specialized radio hardware-based signals, and channel state information (CSI) [9]. Limited sensing capability and the low resolution of RSSI signals make it difficult to achieve fine-grained human activity recognition. Moreover, the specialized radio hardware is not a commercial off-the-shelf device and therefore is expensive to set up.

The information of how wireless signals propagate from the transmitter antenna to the receiver antenna at a certain carrier frequency is captured by CSI. Activity recognition with CSI is carried out based on the following principle: when some objects or persons move between the transmitter antenna and receiver antenna, the amplitude and phase of CSI signals differ from the usual measurements when there is no such movement. Wang *et al.* [10] provides a theoretical framework explaining

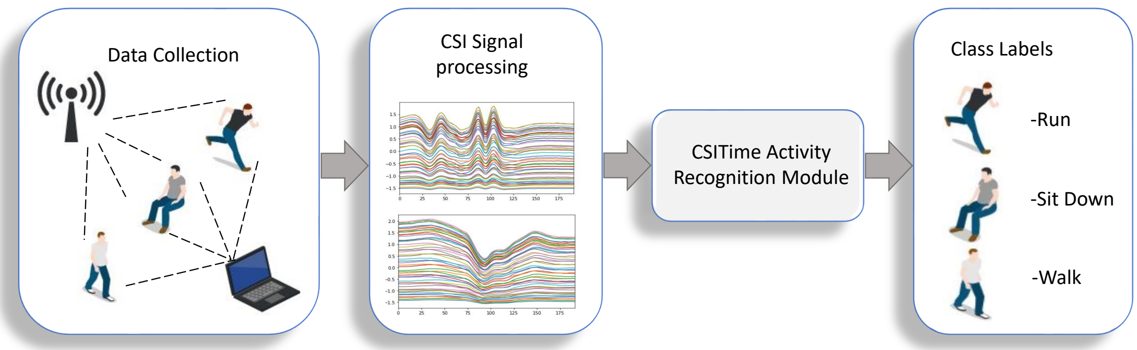


Figure 1: Human activity recognition with CSITime. Channel State Information (CSI) signal data is collected using a WiFi router and a laptop. This data is then preprocessed using different techniques. The preprocessed data is then passed through the CSITime Network. All the activities are hence recognized.

the relationship between human activities and CSI measurements. Researchers used WiFi CSI for a variety of applications such as hearing words through micro-movement detection [6], gesture recognition [11, 12], user identification and localization [13], driver activity recognition [14], handwriting recognition [15], pose estimation [16], and fall detection [1]. Interestingly, Zou *et al.* [17] fused WiFi CSI data with visual data for human activity recognition.

1.1. Challenges

Although there are many merits in using CSI for monitoring human activities, there are some challenges associated with it. The complex relationship between WiFi CSI measurements and human activities is difficult to model precisely with statistical models or classical machine learning algorithms. We propose CSITime for WiFi CSI based activity recognition that makes use of deep learning techniques for automated feature extraction and classification (as shown in Figure 1). The environmental setup for the collection of data plays a crucial role in the performance of the models. Yousefi *et al.* [18] presented that the performance of the classification system goes down drastically when the data is collected in an environment with larger interference.

1.2. Recent Advancements and Motivation

Past works built specific architectures for activity recognition using CSI signals [13, 18]. For example, SignFi [19] that involve more fine-grained actions (with fewer body movements) as compared to StanWiFi [18]. StanWiFi [18] have used LSTMs which are not very powerful when dealing with large number of classes as in SignFi [19], Memmesheimer *et al.* [20] has used deep-learning techniques by taking the multi-variate signal sequences as images and then classifying them using

EfficientNet [21], this had additional computation burden to transform signal sequences to images. We intend to build a generic CSI-based activity recognition system that can yield better results across different granularities of activities. Moreover, we try to correlate the problem of CSI-based activity recognition with time series classification. To the best of our knowledge, none of the previous works have explored the time series-based models for CSI-based activity recognition. Particularly, we see it as a multivariate time series classification problem.

In this work, we advanced the time series classification architecture *i.e.*, InceptionTime [22], for CSI-based activity recognition. InceptionTime [22] is an ensemble of five deep CNN models for time series classification. It was created by cascading multiple inception modules. An Inception module consists of three Inception blocks taken together. A basic Inception block consists of a bottle-neck layer followed by a concatenation of multiple 1D filters of varying lengths; parallel to this is a max-pooling layer followed by a bottle-neck layer. The inception module also has residual connections, which helps in achieving faster convergence. The bottle-neck layer helps in reducing the number of parameters drastically. The depth of a layer is decided based on the length of time-series data.

1.3. Contributions

The proposed CSITime model has three levels of abstractions in terms of its architecture. The lowest abstract layer consists of CSITime basic block. In a CSITime basic block, the CSI signals are passed through kernels of different sizes. The output features are concatenated and passed through a batch norm layer followed by a self-attention layer and a fully connected layer with the Mish activation function. The next higher level abstraction - the CSITime module, has six such basic blocks. The final level abstraction - the CSITime network has one CSITime module followed by a Global Average Pooling layer and a fully connected layer. Figure 1 presents the proposed pipeline and Figure 6 presents the basic block diagram of the proposed CSITime architecture.

The major contributions of the proposed work are as follows.

- This is the earliest work to utilize time series classification-based approach to address the WiFi CSI-based human activity recognition.
- The proposed model has been evaluated on three publicly available benchmark datasets, namely ARIL [13], StanWiFi [18], and SignFi [19], with differing granularities of human activity recognition. CSITime model lifts the state-of-the-art accuracy on the ARIL, StanWiFi, and SignFi datasets.

- CSITime eliminates the need for additional encoding of time series data [20] and it also performs efficiently with more activity classes [19].
- The proposed model is generic and can be easily extended to other activity recognition datasets. Our model requires minimal preprocessing techniques.
- We incorporate recent advancements like Mish activation function, Ranger optimizer, learning rate strategies like Cosine annealing, and data augmentation strategies like Cutmix and Mixup for better performance. We have taken inspirations from neural network developments to take in Self-attention a part of the model as well.

1.4. Paper Organization

The rest of the paper is organized as follows. Section 2 discusses work related to human activity recognition with CSI data. Section 3 describes the proposed methodology. Later, in Section 4, we describe our experiments and results. Finally, we conclude the work in Section 5.

2. Related Works

This section provides a brief literature review of the existing works related to CSI-based WiFi activity recognition. Guo *et al.* proposed a WiFi-activity recognition (WiAR) dataset and a multimodal model - HuAc [23]. The WiAR dataset was recorded in three environments *i.e.*, an empty room, a meeting room (with a desk), and an office, for 16 different activities. They used crowd-sourced skeleton joints (using Kinect sensors) data and CSI data as two different modalities. The feature vectors obtained from posture analysis of skeleton data and the feature vectors of CSI data were merged and fed to the HuAc framework. Four classifiers namely k-nearest neighbor (kNN), Random Forest, Decision Tree, and support vector machines (SVM) were used for the classification. Among all the classifiers, SVM performed the best with an average accuracy of greater than 93% on WiAR. However, this model fails to recognize actions when there are multiple people in the environment. The Kinect sensors lose the sensing ability when there are barriers or occlusions. Ma *et al.* [8] provided a good survey of CSI-based applications.

Wang *et al.* [13] used CSI data for joint activity recognition and indoor localization. The authors collected a dataset of 1440 CSI samples for six different activities like walking, sitting, standing, *etc.* They proposed a multi-task 1D convolutional neural network (CNN) with basic architecture

Table 1: Summary of the literature survey.

Reference	Hardware	Learning method	Datasets	Classes	Public?
Yousefi *et al*[18], 2017	Intel 5300 NIC	LSTM	StanWifi (Self-Collected)	6	Yes
Chowdhury *et al*[9], 2017	Consumer WiFi	FFT features & SVM	Self-Collected	6	No
Ma *et al*[19], 2018	Consumer WiFi	CNN	SignFi (Self-Collected)	276	Yes
Guo *et al*[23], 2018	Consumer WiFi & Kinect	SVM	WiAR (Self-Collected)	16	Yes
Chen *et al*[24], 2018	Intel 5300 NIC	ABLSTM	Self-Collected & StanWiFi[18]	6	No
Wang *et al*[13], 2019	USRP	1-D ResNet	ARIL (Self-Collected)	6	Yes
Yan *et al*[25], 2019	Consumer WiFi	AACA & ELM	Self-Collected	7	No
Damodaran *et al*[26], 2020	Intel 5300 NIC	SVM and LSTM	Self-Collected	5	No
Memmesheime *et al*[20], 2020	Not Applicable (NA)	EfficientNet	ARIL[13],	6	NA
Memmesheime *et al*[20], 2020	Not Applicable (NA)	EfficientNet	UTD-MHAD[27]	6	NA
Memmesheime *et al*[20], 2020	Not Applicable (NA)	EfficientNet	NTU RGB+D[28]	6	NA

based on ResNet [29] and used a simple aggregate loss function. The proposed architecture achieved an average accuracy of 88.13% for activity recognition and 95.68% for indoor localization on their dataset. The CSI data for indoor localization was collected from sixteen locations in a small controlled lab environment. Considering that there are only six classes, an average accuracy of 88.13% for activity recognition is less, which We seek to improve with our proposed architecture.

Damodaran *et al.* [26] performed human activity recognition using WiFi CSI with five different classes, *i.e.*, sit, stand, run, walk, and empty. They have performed the experiments in an indoor environment using two sets of data, Line of Sight scenario (LOS) and Non-Line of Sight (N-LOS) scenario. They have compared the performance of two different algorithms - one using sophisticated preprocessing and feature extraction techniques based on wavelet analysis with a support vector machine (SVM) as a classifier and the other using raw data directly with a long short-term memory (LSTM) network. Although LSTM based algorithm used less preprocessing, its performance was similar to SVM based algorithm.

Yousefi *et al.* [18] proposed a CSI behavior recognition dataset named as StanWiFi dataset. This dataset consist of seven[1] activities like run, lay down, sit down, stand up, *etc.* For the classification of activities, they used a random forest algorithm (with 100 trees), a hidden Markov model (HMM), and an LSTM network. The proposed classification algorithms achieved an accuracy of 64.6%, 73.3%, and 90.5%, respectively. This clearly shows that deep learning networks, like LSTM, perform better than classical machine learning algorithms for WiFi-based activity recognition. Further, the authors stated the need for higher WiFi transmission rates for activity recognition. The performance

[1]in the paper, authors use only six activities

of the classification algorithms was degraded when they sampled the data at 50 Hz, instead of 1kHz that was used in collecting the main dataset. Chen *et al.* [24] used a bi-directional LSTM with an attention mechanism to improve the results on the above-mentioned StanWiFi dataset. The proposed model achieved an average accuracy of 97.3%. Notably, they pointed out that the overall performance of the proposed models was better in environments with smaller interference from different subjects, which were moving in or out of the room.

Memmesheimer *et al.* [20] encoded the multi-variate signal sequences as images and then classified them using EfficientNet [21]. They have tested the performance of their model on ARIL [13] and observed accuracy of 94.91%. One of the important limitations of this work is the computational burden in transforming time-series data into image sequences. On top of that, using relatively heavy architectures like *Efficientnet* exacerbates the issue. In this proposed work we eliminate the need for this expressive encoding of time series sequences by directly extracting useful features from the time series data.

Ma *et al.* [19] performed sign language recognition using WiFi CSI data. The authors released a dataset of 8280 gesture instances with 276 different sign gestures involving hand, arm, head, and finger gestures. The data was collected in three different environments like in the lab, home, lab+home[2]. They proposed a nine-layer CNN and scored an average accuracy of 98.01%, 98.91%, and 94.81% in the three environments. Affirming [18], they have also presented the importance of the data collection environment in CSI-based applications. The average recognition accuracy dropped from 98.01% to 86.67% upon mixing 20% of the data collected in slightly different settings. This shows the non-robustness of the proposed model with respect to data variance.

Yan *et al.* [25] proposed WiAct, which used CSI signal data to extract the variance features for activity recognition. They designed a novel adaptive activity cutting algorithm (AACA) to segment the action part of the signal from the rest. They also extracted the activity-related Doppler shift correlation values from multiple WiFi antennas, which are then fed as input to the extreme learning machine (ELM) classifier for ten different activities. Their algorithm achieved overall recognition accuracy of 94.20%. However, the use of specially hand-crafted values of different hyperparameters and placing AP(access point) and RP(receiver point) close by may have problems when scaling up.

Chowdhury *et al.* [9] proposed a human activity recognition model named WiHACS. Here, a

[2]the third set of samples is a simple combination of the first two sets

single access point and a laptop as used to collect data to find correlation patterns over a range of subcarriers to detect activities in different environments. WiHACS enhanced the accuracy for activity recognition when there are multiple walls as obstructions between source and receiver. WiHACS correctly classified seven different human activities with an average accuracy of 97%, 92%, and 75% in the LOS scenario, across one wall, and two walls, respectively. Table 1 presents the summary of the literature study.

Deep learning techniques are used extensively and successfully in various research fields like computer vision [30, 31, 32] and natural language processing [33], which is one of the reasons inspiring us to use deep learning techniques instead of classical machine learning algorithms. The authors in [34] used an unsupervised approach to approximate the solution of a partially differentiable equation while we have used supervised learning methods in our work. Biologically inspired spiking neural network models can also be used as a classification model after the preprocessing of CSI signals. An additional advantage of these models is that they have less power consumption [35, 36]. Additionally, neuromorphic computing models [37, 38, 39, 40, 41] can also be used as classification models in the current study.

One of the major problems in WiFi CSI-based activity recognition is the lack of publicly available standard datasets. There are only a limited number of publicly available datasets available like the ARIL [13] dataset, StanWiFi [18] dataset, SignFi [19] dataset, and HuAc [23] dataset. Currently, most researchers in this domain evaluate their models using their own collected datasets. So, it is difficult to compare the performance of different methodologies for CSI-based activity recognition. As mentioned earlier, researchers built use-case-based architectures, which may not perform well across other activity recognition datasets. We address this problem by proposing a generic time series-based architecture. Encoding CSI data as images [20] for human activity recognition is computationally expensive as it involves two steps - encoding CSI data and classification, both of which are computationally heavy. Our proposed model directly uses the CSI data without any need for extra feature extraction steps.

3. Proposed Methodology

In this section, we present the proposed methodology of CSITime including the methods adopted for data pre-processing and a description of the few major components used in the neural network model. The proposed method is privacy preserving as the collection of the input to the model (CSI signals) does not capture any direct information of the visual posture or actions of a particular individual in terms of gray-scale or RGB images unlike in human activity recognition with computer vision. Jung *et al.* [42] discusses different privacy issues in using computer vision methods for human activity recognition, all of which can be eliminated with WiFi CSI data. The information generated by extracting personal and behavioral information contained in images and videos can cause privacy infringement issues.

3.1. Pre-processing the CSI signals

The raw CSI data has two components - phase and amplitude of the CSI signals. In this work, we only use the amplitude of CSI signals for activity recognition. For all three datasets *i.e.*, ARIL [13], StanWiFi [18], and SignFi [19], we standardize the training, validation, and test sets before feeding them to the proposed CSITime network. This standardization is applied at the batch level, which can be described as follows.

$$x_{std} = \frac{x - \mu}{\sigma} \tag{1}$$

where, x is the observed value, (μ) is the mean of the batch and (σ) is the standard deviation of the batch. For example, each data point of the ARIL has a shape of 52×192, where 52 is the number of subcarriers and 192 is the number of time steps. A batch size of such samples is taken and standardized across all time steps for each of the subcarriers. The preprocessed CSI signals for three classes of the ARIL dataset are shown in Figure 2.

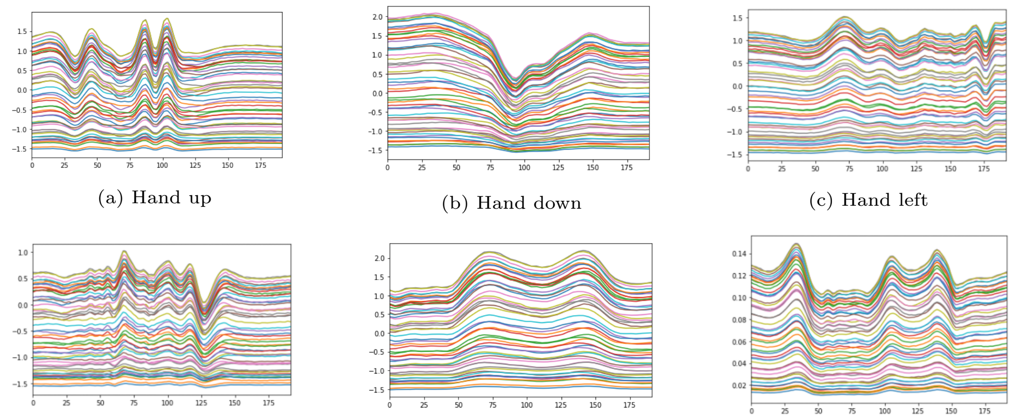
3.2. Components of CSITime

This section describes the adaptations and building blocks of the CSITime.

3.2.1. ***Mish Activation***

We utilized the Mish activation function because it produced better empirical results compared to Swish [43], ReLU [44], and Leaky ReLU [45] activation functions for similar tasks [46]. Mish [46] is a continuously differentiable function, unlike ReLU. The function has a lower bound, hence having strong regularization effects according to the authors [46]. Mish also avoids saturation while training for near-zero gradients. Mish Activation function $f(x)$ can be described as follows.

$$f(x) = xtanh(softplus(x)) = xtanh(ln(1 + e^x)) \tag{2}$$



(a) Hand up (b) Hand down (c) Hand left

(d) Hand right (e) Hand circle (f) Hand cross

Figure 2: ARIL [13] signal representations after preprocessing for, a) hand up, b) hand down, c) hand left - in the first row; and d) hand right, e) hand circle, f) hand cross - in the second row.

where, $tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$.

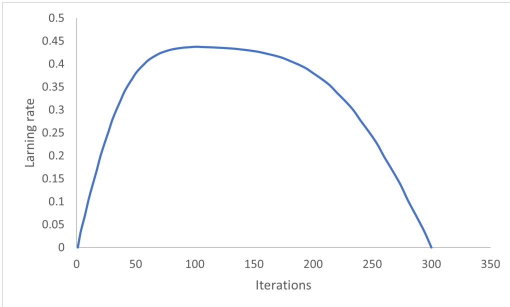
3.2.2. ***Learning Rate Schedules***

We have experimented with two different types of learning rate schedules - One cycle policy and Cosine Annealing.

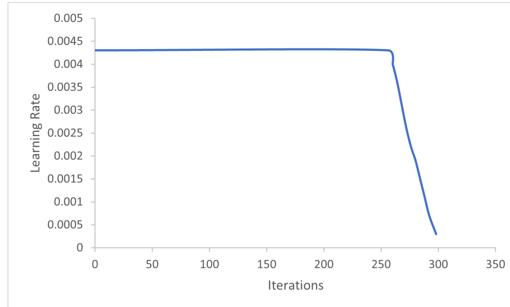
One cycle policy: One cycle policy has a bell-shaped curve for learning rates $w.r.t$ number of iterations as shown in Figure 3a. One cycle policy varies learning rate with the number of epochs to enhance the learning speed [47]. In the middle of training the neural network, higher learning rates of one cycle policy acts as regularisation techniques and helps to attain the global minima quickly. Figure 3a represents how the learning rate varies over iterations in one cycle policy strategy.

Cosine Annealing: Cosine Annealing follows the strategy of having a higher learning rate at the start of training and then decreasing it throughout training [48]. This schedule helps in initial aggressive exploitation to attain the global minima quickly, then reduces the learning rate slowly to avoid jumping out of the minima. Figure 3b represents how the learning rate varies over iterations in cosine annealing strategy. The equation for the learning rate at each epoch in cosine annealing can be described as follows.

$$a(t) = \frac{a_0}{2}\left(\cos\left(\frac{\pi \times mod\big(t-1, \lceil T/M \rceil\big)}{\lceil T/M \rceil}\right) + 1\right) \tag{3}$$



(a) One cycle



(b) Cosine Annealing

Figure 3: This figure represents how learning rate varies over iterations in (a) one cycle policy [47] strategy and (b) cosine annealing [48] strategy.

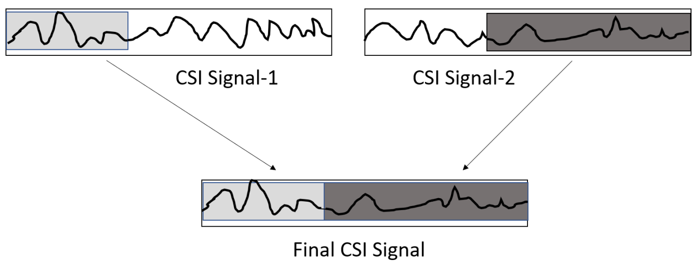


Figure 4: Cutmix [49] data augmentation. A part of the signal is concatenated with the latter part of another signal to get a new signal.

where, $a(t)$ is the learning rate at epoch t, a_0 is the maximum learning rate, T is the total number of epochs, M is the number of cycles, *mod* is the modulo operation, and the *square brackets* indicate the floor operation [48].

3.2.3. ***Optimizers***

The default optimizer (wherever not mentioned) used was ADAM [50] optimizer. But we also performed a comparative analysis with the Ranger [51] optimizer. Ranger is a synergistic optimizer using rectified ADAM (RAdam) and LookAhead [52]. LookAhead improves the learning stability and lowers the variance of its inner optimizer RAdam, with negligible computation and memory cost. The FastAI [51] community has been able to achieve higher accuracies using this optimizer even with lesser training.

3.2.4. ***Self-Attention***

The attention mechanism tries to find the relative importance of feature vectors in deciding the output of a particular layer of neural network [53]. In this proposed work, there is no apriori information to decide about the weights. So, the output data from the basic CSITime block is used to calculate weights for self-attention.

Let us assume that basic InceptionTime module gives m feature vectors as $(h_1, h_2, .., h_n)$. A score function (s_i) calculates the importance of each feature vector as follows.

$$s_i = \phi(W^i h_i + b) \tag{4}$$

where W^i is the weight vector and b is the bias vector.

After calculating the score for all of the feature vectors, the weights are normalized.

$$a_i = \frac{exp(s_i)}{\sum_{i=1}^{n} exp(s_i)} \tag{5}$$

Finally, the feature vector is multiplied by its relative weight to get the final feature vector.

$$\mathbf{O} = \sum_{i=1}^{n} a_i * \mathbf{h}_i \tag{6}$$

3.2.5. ***Data Augmentation***

We have experimented with two different types of data augmentation techniques *i.e.*, (a) Mixup [54] and (b) Cutmix [49]. Mixup is used to generate random datapoints from a pair of datapoints in the given data. We generate a synthetic training example $(\hat{x}, \hat{y})$ from (x_i, y_i) and (x_j, y_j), *i.e.*, a pair of ground truth datapoints and labels by using equations as follows.

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j \tag{7}$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j \tag{8}$$

Mixup regularizes the neural network to favor simple linear combinations of training examples. It also has no additional overhead. Figure 5 represents Mixup data augmentation strategy. In this, two different signals are mixed in a parametrized proportionate manner.

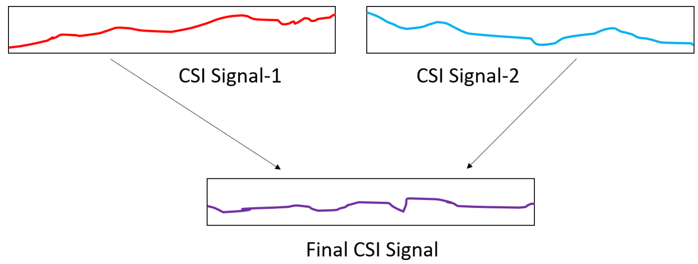


Figure 5: Mixup [54] data augmentation. Two different signals are mixed in a parametrized proportionate manner.

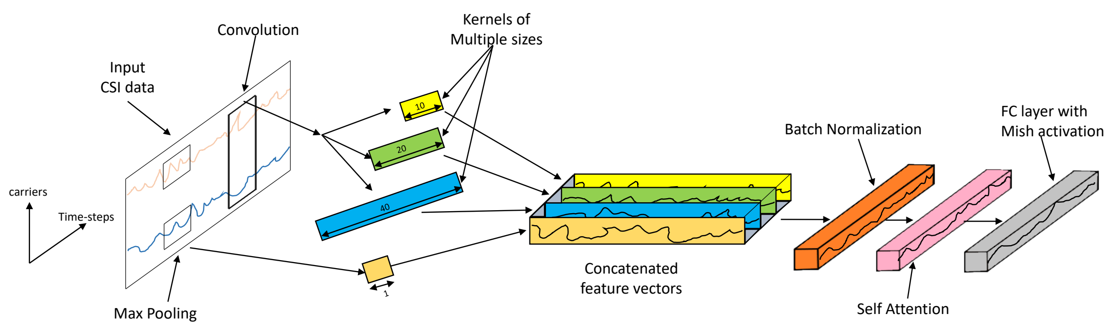


Figure 6: The basic block diagram of the proposed CSITime architecture. The CSI signals are passed through three kernels of different sizes. The output features are concatenated with features obtained from a max-pooling layer and passed through a batch-norm layer followed by a self-attention layer and a fully connected layer with the Mish activation function. One CSITime module has six such blocks. CSITime network has one CSITime module followed by a Global Average Pooling layer and a fully connected layer.

Cutmix technique cuts and pastes random patches of data from one sample to another. Cutmix increases the localization and generalization capability of the network, as it gives equal importance to the non-discriminative part of the sample. Figure 4 represents Cutmix data augmentation strategy. The ground truth labels are mixed in proportion to the area of patches in the images, similar to Mixup equations.

3.3. CSITime Architecture

In this sub-section, we provide details of the proposed architecture - CSITime. CSITime has three levels of abstractions in terms of its architecture. The lowest abstract layer consists of CSITime basic block. In a CSITime basic block, the CSI signals are passed through kernels of different sizes. In our work, we have considered kernels with three different sizes - 10, 20, and 40. The output features are concatenated with features obtained from a max-pooling layer and passed through a batch-norm layer followed by a self-attention layer and a fully connected layer with the Mish activation function. The next higher level abstraction - the CSITime Module, has six such basic blocks (where the depth is six). We have also used residual connections [29] after the second and fifth blocks. The final level abstraction - CSITime Network has one CSITime module followed by a global average pooling layer and a fully connected layer. Figure 6 presents the basic block diagram of the proposed CSITime architecture.

3.3.1. ***Reducing the Computational Burden***

The original InceptionTime [22] model had an ensemble of five inception modules. Each module was initialized with a different set of weights, derived from various random seed values. In our work, we only use one modified inception-time module, which reduces the computational burden to a larger extent. Although an ensemble of five models gives comparatively better results than a single model, we think that going with a single model is justified due to the need for less computational burden in WiFi CSI-based activity recognition. Different components described in Section 3.2 are used at different levels of the entire training process.

3.3.2. ***Data Augmentation***

We apply two label-mixing data augmentation strategies- Mixup and Cutmix on the pre-processed data. In [55], adversarial-based data augmentation is used, but is computationally expensive than Mixup and Cutmix. Unlike typical data augmentation strategies, these strategies do not increase the number of data samples. They simply do a parametrized mixing of different samples, altering the data samples. Also, unlike other data augmentation strategies, they also modify the labels.

3.3.3. ***Architectural Design***

InceptionTime [22] used the ReLu activation function as the main activation function in its basic unit. Recent studies demonstrated that the Mish activation function achieved superior performance compared to ReLu [46]. This inspired us to replace the ReLu with the Mish activation function in the proposed CSITime. One of the primary motivations of this paper is to see the WiFi CSI-based activity recognition as a time series classification problem. Hence, we take inspiration from successful time series techniques. Additionally, we use a self-attention mechanism to improve the performance of the CSTTime Network for activity recognition with WiFi CSI data. We use a self-attention layer in the third and fifth basic inception-time modules after batch norm and before the Mish. The intuition behind using the attention layer is as follows: the basic inception-time block has extracted useful features from a given data sample. But, the contribution of different learned features to the final activity recognition can be different and the attention layer learns these contributions in terms of the weights [24].

3.3.4. ***Training Phase***

Early stopping with a maximum number of epochs is a common practice in training neural networks. Recent works [47, 48] suggest that usage of learning rate schedulers like one-fit policy and cosine annealing gives better performance due to enhanced learning of the neural network model. As discussed in 3.2, the learning rate varies over the entire training phase in a particular pattern.

3.4. Training CSITime Network.

All of the experiments were performed using Colab GPU runtime[3]. For ARIL, we used the separate training and testing datasets as provided. The previous works [18, 24, 19] used k-fold validation for StanWiFi and SignFi datasets - ten folds for StanWiFi and five folds for SignFi. To maintain consistency and for a fair comparison, we followed the same pattern while testing our model.

Batch sizes for training and validation data were set as 64 and 128, respectively, *i.e.*, the neural network is trained on a batch of 64 samples and validated on a batch of 128 samples. The model was trained for 300 epochs for ARIL, and 20 epochs for StanWiFi and SignFi. As mentioned earlier, we used two different learning rate schedules - one cycle policy and cosine annealing. We used the cross-entropy loss ($CELoss$) function, which can be described as follows.

[3] `https://colab.research.google.com/`

$$CELoss = -\sum_{i=1}^{C^i} t_i log(f(s_i)) \tag{9}$$

$$f(s_i) = \frac{e^{s_i}}{\sum_{i=1}^{n} e^{s_i}} \tag{10}$$

where, s_i is the input feature vector, and $f(s_i)$ is the softmax activation function. C^i is the total number of classes and $t_i \in \{0, 1\}$.

4. Experimental Results

In this section, we present details about the experiments carried out with the proposed CSITime architecture on three different datasets. Besides the performance of the proposed model, we also present a comparative analysis of the results of the proposed model with respect to state-of-the-art models on all three datasets.

4.1. Dataset Details

We performed our experiments on three datasets as described below.

4.1.1. ***ARIL Dataset***

The ARIL dataset [13] comprises six hand activities *i.e.*, hand up, hand down, hand left, hand right, hand circle, and hand cross, as shown in Figure 7. For data collection, each volunteer repeats each of the six activities fifteen times at sixteen different locations and forms a dataset with 1440 activity samples, which are used for training and testing. The CSI data was collected using two universal software radio peripherals (USRPs). The USRP is used to broadcast or receive WiFi signals. We can use hand gestures for several applications where using cameras may have privacy concerns. We can set different meanings for each gesture and can be applied to our use case. For example, 'hand up' and 'hand down' can be used to turn up and down the volume in television, similarly, 'hand left' and 'hand right' may be used to switch channels.

4.1.2. ***StanWiFi Dataset***

The StanWiFi dataset [18] comprises six activities *i.e.*, lie down, fall, walk, run, sit down, and stand up, as shown in Figure 8. These six activities are performed by six individuals, with each

Figure 7: This figure shows activities in ARIL [13] dataset. The dataset consists of six actions-hand circles, hand up, hand cross, hand left, hand down, and hand right.

activity being performed 20 times. The data distribution among different classes is as follows: Lay down-657, Fall-1465, Walk-1209, Run-400, Sit down-304, and Stand Up-443 samples. The input feature vector is the raw CSI amplitude data, a 90-dimensional vector (3 antennas and 30 subcarriers). The data was collected using one WiFi router and a laptop fitted with Intel's network interface card (NIC 5300). The original dataset contains seven different activities, but we have only used the six mentioned above. A single training datapoint has a shape of: $number\ of\ samples \times number\ of\ features(500) \times number\ of\ timestamps(90)$.

4.1.3. ***SignFi Dataset***

The SignFi dataset [19] comprises 276 different sign gestures corresponding to different sign language words. WiFi CSI traces were collected in two different environments *i.e.*, lab and home environments. The dimension of the lab and home was $13m \times 12m$ and $4.11m \times 3.86m$, respectively. The lab has more surrounding objects, leading to a more complex multi-path environment than the home. Two example images from this dataset are shown in Figure 9. The dataset has a total of 8,280 instances of 276 sign gestures from the lab and home environments, with 5,520 instances from the lab and 2,760 instances from home. The data was collected using a WiFi

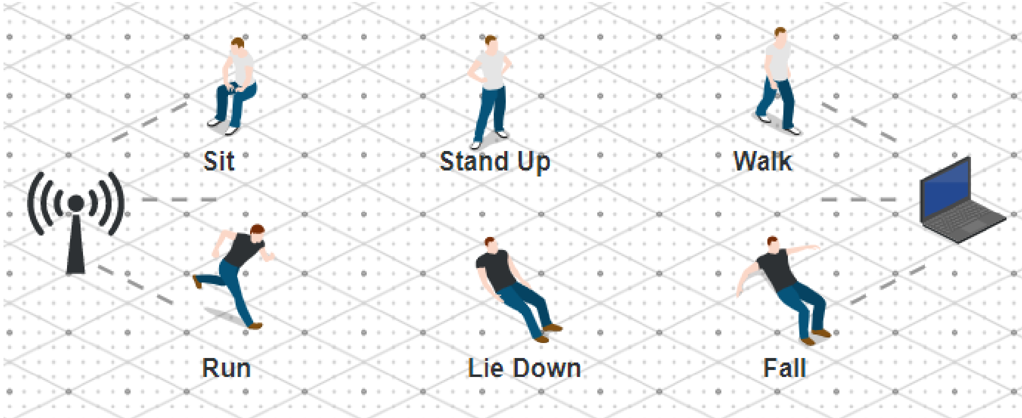


Figure 8: This figure shows activities from StanWiFi [18] dataset. StanWiFi CSI dataset consists of six basic activities performed by six individuals, each activity being performed 20 times.

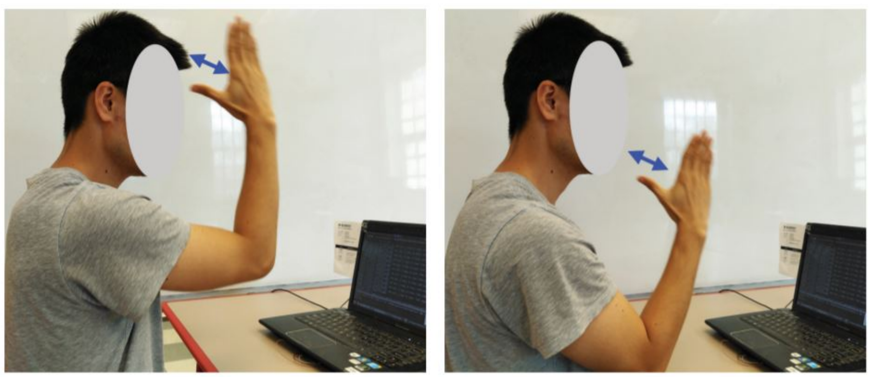


Figure 9: This figure shows activities in SignFi [19] dataset. The shown activities are symbols for 'Father' and 'Mother', respectively.

AP and a WiFi station. Only the data packets sent from WiFi Access Point to laptop (station) *i.e.*, downlink CSIs were used for experimentation. A single training datapoint has a shape of: $number\ of\ samples \times number\ of\ features\ (30) \times number\ of\ channels\ (3)$.

4.2. Performance Evaluation

This section presents the experimental results on the three datasets *i.e.*, ARIL, StanWiFi, and SignFi, using the proposed CSITime model. Table 2, 3, and 4 present the experimental results of the variants of the proposed CSITime architecture on the ARIL, StanWiFi, and SignFi datasets,

Table 2: Experimental results of the proposed network on ARIL [13].

Model Variant	Accuracy	Precision	Recall	F1 score
Mish	97.12%	97.15%	97.12%	97.12%
Mish + Ranger	97.48%	97.57%	97.48%	97.48%
Mish + Self Attention	97.12%	97.22%	97.12%	97.14%
Mish + Cos. Ann	96.04%	96.05%	96.04%	96.02%
Mish + Ranger + Cos. Ann	97.12%	98.23%	97.12%	97.11%
Mish + Ranger + MixUp	**98.20%**	**98.23%**	**98.20%**	**98.20%**
Mish + Ranger + CutMix	97.84%	97.86%	97.84%	97.83%

respectively. The order of increasing granularity of the three datasets is - ARIL (6 classes), StanWiFi (7 classes[4]), and SignFi (276 classes). We use four metrics - accuracy, precision, recall, and F1-score for the ARIL and StanWiFi datasets. For the SignFi dataset, we present the accuracies of the proposed model on different variants of the dataset provided by the authors. It can be seen from Table 2, 3, and 4 that different variants of the proposed model give similar results across all datasets with some exceptions. Also, the near-perfect results on all the datasets show the efficacy of our proposed CSITime architecture. Results (accuracy) vary very little on the ARIL dataset, with the minimum being 96.04% and the maximum being 98.20%. The mixup data augmentation strategy, along with the Ranger optimizer, achieves the highest accuracy of 98.20% on ARIL. Comparatively, ARIL has fewer samples of data, only 1440 samples for training and validation together. Hence, we infer that with the help of mixup data augmentation, additional synthetic samples help the neural network learn more features and generalize well on the data.

On StanWiFi, the CSITime model variant with only Mish activation function without other learning rate schedulers and data augmentation techniques yields the highest results and scored an accuracy of 98.88%, precision of 99.16%, recall of 98.87%, and F1-score of 99.01%.

Mish with cosine annealing learning rate scheduler outperforms other model variants for Lab, Home, Lab+Home versions of SignFi with accuracies of 99%, 98%, and 99.09%. For lab-5 users variant of SignFi, Ranger optimizer with Cutmix data augmentation strategy gives the highest result of 88.3%. The results obtained are pretty impressive, given that there are 150 classes in the dataset, which implies a random baseline of 0.06% accuracy. It can be inferred from Table 4 that the data augmentation severely worsens the performance in the home environment where the accuracies are 20-30% lower than the usual results. The superior performance of Cutmix data augmentation in the lab-5 users' scenario is relatively easy to interpret. Section 4.1.3 states that

[4]The original dataset contains seven classes, but we have used 6 classes for better comparison with other works, which also had done the same.

Table 3: Experimental results of the proposed network on StanWiFi [18].

Model Variant	Accuracy	Precision	Recall	F1 score
Mish	**98.88%**	**99.16%**	**98.87%**	**99.01%**
Mish + Ranger	98.21%	98.80%	97.53%	98.13%
Mish + Self Attention	98.43%	98.88%	98.25%	98.54%
Mish + Cos. Ann	98.21%	98.64%	97.58%	98.07%
Mish + Ranger + Cos. Ann	97.76%	98.45%	97.78%	98.06%
Mish + Ranger + MixUp	97.54%	98.21%	97.26%	97.64%
Mish + Ranger + CutMix	96.20%	97.21%	96.50%	96.85%

Table 4: Experimental results of the proposed network on SignFi [19].

Model Variant	Lab	Home	Lab + Home	Lab 5
Mish	98.64%	96.37%	98.12%	82.86%
Mish + Ranger	96.82%	83.15%	95.89%	81.60%
Mish + Self Attention	98.73%	95.83%	98.91%	81.73%
Mish + Cos. Ann	**99.00%**	**98.00%**	**99.09%**	81.46%
Mish + Ranger + Cos. Ann	98.91%	94.38%	98.42%	80.60%
Mish + Ranger + MixUp	95.74%	74.63%	92.33%	81.26%
Mish + Ranger + CutMix	93.93%	63.40%	89.00%	**83.60%**

Table 5: Comparison with the state-of-the-art results on the ARIL [13] dataset. **Blue** represents the previous state-of-the-art. **Red** denotes the best results.

Model	Accuracy
ARIL[13]	88.13%
Gimme' signals[20]	**94.91%**
Ours Variant-1(M+R+C)	97.12%
Ours Variant-2(M+R+M)	**98.20%**

Table 6: Comparison with the state-of-the-art results on the SignFi [19] dataset. **Blue** represents the previous state-of-the-art [19]. **Red** denotes the best results.

Model	Lab	Home	Lab+Home	Lab-5
SignFi[19]	**98.01%**	**98.91%**	**94.81%**	**86.66%**
Ours Variant(M+C)	**99.22%**	97.39%	**96.23%**	**88.83%**

the data collection environment for the first four users significantly differs from that of the last user. In fact, this distributional data difference is the main reason for less accuracy in lab-5 users scenario compared to other scenarios like the lab, home, *etc.* The data augmentation increases the number of samples of the fifth user and other users, giving the model more information about the fifth user while softening the data distribution. As Cutmix picks samples randomly from given training samples, there are high chances that it picks up samples from the fifth user to mix with the remaining four users' samples.

4.3. Discussion and Comparison

This section presents a comparative analysis of the methods and results of our proposed CSITime model on the three datasets *i.e.*, ARIL, StanWiFi, and SignFi. Our proposed model beats the state-

Table 7: The confusion matrix of StanWiFi [18], Gimme' signals [20], and our proposed CSITime model on the StanWiFi dataset. **Blue** represents the previous state-of-the-art [20]. **Red** denotes the best results.

Actual Classes \ Predicted Classes	Lie down	Fall	Walk	Run	Sit down	Stand up
Lie down	**0.95**	0.01	0.01	0.01	0.00	0.02
Fall	0.01	**0.94**	0.05	0.00	0.00	0.00
Walk	0.00	0.01	**0.93**	0.04	0.01	0.01
Run	0.00	0.00	0.02	**0.97**	0.01	0.00
Sit down	0.03	0.01	0.05	0.02	**0.81**	0.07
Stand up	0.01	0.00	0.03	0.05	0.07	**0.83**

Actual Classes \ Predicted Classes	Lie down	Fall	Walk	Run	Sit down	Stand up
Lie down	**0.96**	0.0	0.01	0.0	0.02	0.01
Fall	0.0	**0.99**	0.0	0.01	0.0	0.0
Walk	0.0	0.0	**0.98**	0.02	0.0	0.0
Run	0.0	0.0	0.02	**0.98**	0.0	0.0
Sit down	0.01	0.01	0.01	0.0	**0.95**	0.02
Stand up	0.01	0.0	0.0	0.0	0.01	**0.98**

Actual Classes \ Predicted Classes	Lie down	Fall	Walk	Run	Sit down	Stand up
Lie down	**0.98**	0.0	0.1	0.0	0.01	0.0
Fall	0.0	**1.0**	0.0	0.0	0.0	0.0
Walk	0.0	0.0	**0.99**	0.01	0.0	0.0
Run	0.0	0.0	0.0	**1.0**	0.0	0.0
Sit down	0.01	0.01	0.01	0.0	**0.96**	0.01
Stand up	0.01	0.01	0.01	0.0	0.02	**0.95**

of-the-art on all three datasets. With respect to the comparision of methods, Wang *et al.* [13] used 1D Resnet and Memmesheimer *et al.* [20] used encoding of time series data followed by classification with Efficientnet [21] on ARIL dataset. Ma *et al.* [19] used convolutional neural networks on SignFi dataset and Yousefi *et al.* [18] used LSTM on StanWiFi dataset. We have used the proposed

CSITime model on all of these three datasets. Table 5 presents the comparison with the state-of-the-art results on the ARIL [13] dataset. The original paper [13] (which provided the ARIL dataset) records 88.13% accuracy on ARIL, while the Gimme' signals paper [20] beats the then state-of-the-art by around 7%. The variant of the proposed network with Mish activation function, Ranger optimizer, and Mixup data augmentation technique gave the best result of 98.20%. It is also worth noting that all the variant of the proposed model gives higher results than the state-of-the-art.

Table 6 shows the comparative results for the SignFi dataset. Our proposed model performs better in all data collection environments except in the 'home' scenario. The reader should also note that the previous results were as high as 98%. So, it is harder to lift the state-of-the-art beyond that. In the lab-5 users' scenario, the accuracy was 2.17% higher than Ma *et al.* [19], which shows the comparative robustness of our model[5].

Previous works that worked on StanWiFi data used an average k-fold confusion matrix to present their final results. Hence, we also present the average k-fold confusion matrix (as shown in Table 7) of the proposed model on the StanWiFi dataset for comparison purposes. In Table 7, the confusion matrix represents the results of the LSTM-based model presented in Yousefi *et al.* [18]. The confusion matrix in Table 7 represents the results of the ABLSTM model (in Gimme' Signals [20]) presented in [24]. The confusion matrix obtained using our approach in Table 7 clearly beats the results on the ABLSTM model for the first five classes - Lie down, Fall, Walk, Run and Sit down and achieves perfect accuracy for Fall & Run classes. Although our model gives 3% fewer results for the 'Stand up' class, the average accuracy of the proposed model is higher than the state-of-the-art.

4.4. *Limitations of the proposed work*

We could not assess the performance of the proposed model in an environment with high interference due to the lack of availability of a standard dataset that uses the CSI data collected in such settings. We want to explore this in our future work. MiniRocket [56], a very recent work on time series classification, demonstrated that using random convolutions can provide better results within significantly lesser time. In comparison to this, our proposed model is relatively heavy in terms of the number of parameters[6].

[5] Note that this result is obtained with basic InceptionTime module without any additional incorporations.

[6] The approach in the MiniRocket [56] does not require any neural network parameters

5. Conclusion and Future work

In this paper, we presented CSITime, a generic neural network architecture for CSI-based human activity recognition. We have seen the WiFi CSI activity recognition problem as a multivariate time series classification problem and proposed CSITime taking inspiration from time-series methods. We have tested the proposed model on three different datasets - ARIL, StanWiFi (coarse-grained activity recognition), and SignFi (fine-grained activity recognition). Our model achieves state-of-the-art performance across all these datasets. We have presented a detailed experimental analysis to help replication of the results. We have also presented a comparative analysis of the results with respect to other works on the same datasets. Future work would include collection and experimentation with real-world datasets which may have been collected in environments with high interference. The publicly available datasets have only a limited number of preliminary activities. These datasets are far from realistic because an average person performs many other indoor activities in a day. The focus should also be directed towards developing lightweight and easily deployable solutions. Recognizing the activity of multiple people simultaneously using WiFi CSI data is more challenging and another good direction for future research.

Acknowledgment

The authors would like to thank anonymous reviewers and our parent organizations for extending their support for the betterment of the manuscript. We appreciate the assistance provided by CSIR, India.

Funding Information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] C. Han, K. Wu, Y. Wang, L. M. Ni, Wifall: Device-free fall detection by wireless networks, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 271–279. doi:10.1109/INFOCOM.2014.6847948.

[2] J. Wannenburg, R. Malekian, Physical activity recognition from smartphone accelerometer data for user context awareness sensing, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (12) (2016) 3142–3149.

[3] S. Kaplan, M. A. Guvensan, A. G. Yavuz, Y. Karalurt, Driver behavior analysis for safe driving: A survey, IEEE Transactions on Intelligent Transportation Systems 16 (6) (2015) 3017–3032.

[4] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, P. Havinga, Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey, in: 23th International conference on architecture of computing systems 2010, VDE, 2010, pp. 1–10.

[5] H. D. Mehr, H. Polat, A. Cetin, Resident activity recognition in smart homes by using artificial neural networks, in: 2016 4th international istanbul smart grid congress and fair (ICSG), IEEE, 2016, pp. 1–5.

[6] G. Wang, Y. Zou, Z. Zhou, K. Wu, L. M. Ni, We can hear you with wi-fi!, IEEE Transactions on Mobile Computing 15 (11) (2016) 2907–2920.

[7] S. K. Yadav, K. Tiwari, H. M. Pandey, S. A. Akbar, A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions, Knowledge-Based Systems (2021) 106970.

[8] Y. Ma, G. Zhou, S. Wang, Wifi sensing with channel state information: A survey, ACM Computing Surveys (CSUR) 52 (3) (2019) 1–36.

[9] T. Z. Chowdhury, C. Leung, C. Y. Miao, Wihacs: Leveraging wifi for human activity classification using ofdm subcarriers' correlation, in: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2017, pp. 338–342.

[10] W. Wang, A. X. Liu, M. Shahzad, K. Ling, S. Lu, Understanding and modeling of wifi signal based human activity recognition, in: Proceedings of the 21st annual international conference on mobile computing and networking, 2015, pp. 65–76.

[11] Z. Hao, Y. Duan, X. Dang, Y. Liu, D. Zhang, Wi-sl: Contactless fine-grained gesture recognition uses channel state information, Sensors 20 (14) (2020) 4025.

[12] D. Jiang, M. Li, C. Xu, Wigan: A wifi based gesture recognition system with gans, Sensors 20 (17) (2020) 4757.

[13] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, J. Han, Joint activity recognition and indoor localization with wifi fingerprints, IEEE Access 7 (2019) 80058–80068.

[14] S. Duan, T. Yu, J. He, Widriver: Driver activity recognition system based on wifi csi, International Journal of Wireless Information Networks 25 (2) (2018) 146–156.

[15] Z. Guo, F. Xiao, B. Sheng, H. Fei, S. Yu, Wireader: Adaptive air handwriting recognition based on commercial wifi signal, IEEE Internet of Things Journal 7 (10) (2020) 10483–10494.

[16] F. Wang, S. Panev, Z. Dai, J. Han, D. Huang, Can wifi estimate person pose?, arXiv preprint arXiv:1904.00277 (2019).

[17] H. Zou, J. Yang, H. Prasanna Das, H. Liu, Y. Zhou, C. J. Spanos, Wifi and vision multimodal learning for accurate and robust device-free human activity recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[18] S. Yousefi, H. Narui, S. Dayal, S. Ermon, S. Valaee, A survey on behavior recognition using wifi channel state information, IEEE Communications Magazine 55 (10) (2017) 98–104.

[19] Y. Ma, G. Zhou, S. Wang, H. Zhao, W. Jung, Signfi: Sign language recognition using wifi, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2 (1) (2018) 1–21.

[20] R. Memmesheimer, N. Theisen, D. Paulus, Gimme signals: Discriminative signal encoding for multimodal activity recognition, arXiv preprint arXiv:2003.06156 (2020).

[21] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[22] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, F. Petitjean, Inceptiontime: Finding alexnet for time series classification, Data Mining and Knowledge Discovery 34 (6) (2020) 1936–1962.

[23] L. Guo, L. Wang, J. Liu, W. Zhou, B. Lu, Huac: Human activity recognition using crowdsourced wifi signals and skeleton data, Wireless Communications and Mobile Computing 2018 (2018).

[24] Z. Chen, L. Zhang, C. Jiang, Z. Cao, W. Cui, Wifi csi based passive human activity recognition using attention based blstm, IEEE Transactions on Mobile Computing 18 (11) (2018) 2714–2724.

[25] H. Yan, Y. Zhang, Y. Wang, K. Xu, Wiact: A passive wifi-based human activity recognition system, IEEE Sensors Journal 20 (1) (2019) 296–305.

[26] N. Damodaran, J. Schäfer, Device free human activity recognition using wifi channel state information, in: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2019, pp. 1069–1074.

[27] C. Chen, R. Jafari, N. Kehtarnavaz, Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, in: 2015 IEEE International conference on image processing (ICIP), IEEE, 2015, pp. 168–172.

[28] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A. C. Kot, Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding, IEEE transactions on pattern analysis and machine intelligence 42 (10) (2019) 2684–2701.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[30] S. Zhang, F. He, Drcdn: learning deep residual convolutional dehazing networks, The Visual Computer 36 (9) (2020) 1797–1808.

[31] H. Li, F. He, Y. Chen, Y. Pan, Mlfs-ccde: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution, Memetic Computing 13 (1) (2021) 1–18.

[32] Q. Quan, F. He, H. Li, A multi-phase blending method with incremental intensity for training detection networks, The Visual Computer 37 (2) (2021) 245–259.

[33] Y. Pan, F. He, H. Yu, Learning social representations with deep autoencoder for recommender system, World Wide Web 23 (4) (2020) 2259–2279.

[34] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112790.

[35] S. Yang, J. Wang, B. Deng, M. R. Azghadi, B. Linares-Barranco, Neuromorphic context-dependent learning framework with fault-tolerant spike routing, IEEE Transactions on Neural Networks and Learning Systems (2021) 1–15doi:10.1109/TNNLS.2021.3084250.

[36] S. Yang, T. Gao, J. Wang, B. Deng, B. Lansdell, B. Linares-Barranco, Efficient spike-driven learning with dendritic event-based processing, Frontiers in Neuroscience 15 (2021) 97. doi:10.3389/fnins.2021.601109.
URL https://www.frontiersin.org/article/10.3389/fnins.2021.601109

[37] S. Yang, B. Deng, J. Wang, C. Liu, H. Li, Q. Lin, C. Fietkiewicz, K. A. Loparo, Design of hidden-property-based variable universe fuzzy control for movement disorders and its efficient reconfigurable implementation, IEEE Transactions on Fuzzy Systems 27 (2) (2018) 304–318.

[38] S. Yang, X. Wei, B. Deng, C. Liu, H. Li, J. Wang, Efficient digital implementation of a conductance-based globus pallidus neuron and the dynamics analysis, Physica A: Statistical Mechanics and its Applications 494 (2018) 484–502.

[39] S. Yang, B. Deng, J. Wang, H. Li, M. Lu, Y. Che, X. Wei, K. A. Loparo, Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons, IEEE transactions on neural networks and learning systems 31 (1) (2019) 148–162.

[40] S. Yang, J. Wang, X. Hao, H. Li, X. Wei, B. Deng, K. A. Loparo, Bicoss: toward large-scale cognition brain with multigranular neuromorphic architecture, IEEE Transactions on Neural Networks and Learning Systems (2021).

[41] S. Yang, J. Wang, N. Zhang, B. Deng, Y. Pang, M. R. Azghadi, Cerebellumorphic: large-scale neuromorphic model and architecture for supervised motor learning, IEEE Transactions on Neural Networks and Learning Systems (2021).

[42] I. Y. Jung, A review of privacy-preserving human and human activity recognition., International Journal on Smart Sensing & Intelligent Systems 13 (1) (2020).

[43] P. Ramachandran, B. Zoph, Q. Le, Swish: a self-gated activation function (10 2017).

[44] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.

[45] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853 (2015).

[46] D. Misra, Mish: A self regularized non-monotonic activation function, arXiv preprint arXiv:1908.08681 (2019).

[47] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay, arXiv preprint arXiv:1803.09820 (2018).

[48] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, K. Q. Weinberger, Snapshot ensembles: Train 1, get m for free, arXiv preprint arXiv:1704.00109 (2017).

[49] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.

[50] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[51] L. Wright, Meet Ranger - RAdam + Lookahead optimizer, `https://forums.fast.ai/t/meet-ranger-radam-lookahead-optimizer/52886`, [Online; August-2019] (2019).

[52] M. R. Zhang, J. Lucas, G. Hinton, J. Ba, Lookahead optimizer: k steps forward, 1 step back, arXiv preprint arXiv:1907.08610 (2019).

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv:1706.03762 (2017).

[54] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[55] T.-D. H. Nguyen, H.-N. H. Nguyen, Towards a robust wifi-based fall detection with adversarial data augmentation, in: 2020 54th Annual Conference on Information Sciences and Systems (CISS), IEEE, 2020, pp. 1–6.

[56] A. Dempster, D. F. Schmidt, G. I. Webb, Minirocket: A very fast (almost) deterministic transform for time series classification, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 248–257.